Quality Metrics System for Very Large Collections

Ion IVAN, Sorin-Lucian PAVEL
Academy of Economic Studies, Bucharest, Romania
ionivan@ase.ro, pavelsorin@gmail.com

The paper identifies the current need for large data collections (LDC) and software oriented on LDC and defines the concepts. In order to measure the quality characteristics, a method for size estimation is proposed and implemented. Five quality characteristics for LDC – accuracy, completeness, homogeneity, reliability and maintainability – are described, and quality metrics for each characteristic are expressed. Three different proprieties are taken into account for each metric: sensitivity, non-compensatory and non-catastrophic character. A case study is designed measuring the quality metrics for multiple dataset collections. The index of general quality is defined and refined. A system of quality indexes is formed and a method for analyzing its stability is proposed. The method should state whether a system of indexes is stabile or not.

Keywords: *Datasets, Metrics, Quality, Stability, Method*

1 Large datasets

Computerization of contemporary society, the spread of citizen-oriented software, and promulgation of new laws in the IT field in recent years have led to the emergence of applications that work with large and very large datasets ($10^7 \div 10^{10}$ sets). The goal of each set of data (DS) is to capture reality in an objective and accurate manner and to record it as stored information that is used later in different processes. To achieve the intended purpose, the datasets must take into account the nature of reality that is recorded, in order to contain specific data.

The informational reality is characterized by:

- *complexity* due to numerous details, connections, influences and manifestations of processes; each aspect must be captured and recorded in a dataset, as its informational power and value is given by the completeness and accuracy of data submitted; the complexity of reality depends on the area of observation, on the impact and importance of component issues, and on the degree of interaction with other domains of reality;
- *variation* because the behavior of data, indicators or actual processes does not necessarily follow strict mathematical laws; so the values recorded are part of the set of possible values; the degree of uncertainty is high, as extreme values are possible at any time and data sets must be able to include such values;
- *granularity* due to large number of constituent elements organized in types, classes, subclasses, and so on; each element of a class is different from any other item in a different class by characteristics; due to the complexity, the organization by classes and elements is not accurate in many cases; a form of organization is presented in Figure 1.

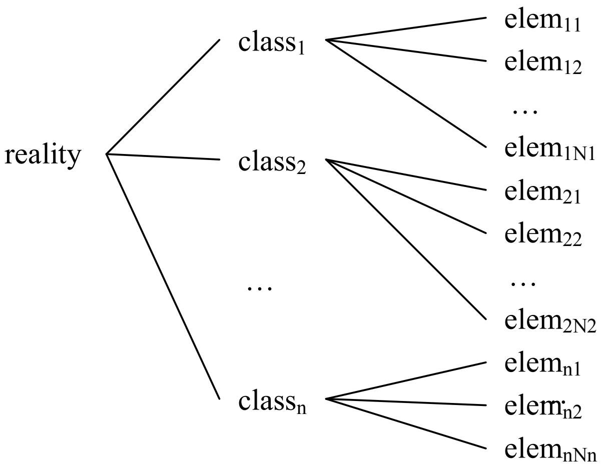

Fig. 1. Granularity of informational reality

Where:
N – number of reality classes;
N_i– number of elements from class *i*.

Let C_i be a class containing elements $\{e_{i1}; e_{i2}; ...; e_{iNi}\}$. For the components of this class, M_i descriptive characteristics are noted $\{k_1;k_2;...;k_{Mi}\}$, available for every element.

The values of these attributes are determined by:

- *measuring*, in case there are both units of measure for that attribute as well as tools for determining the characteristic value; measure-determined fields describe: height, length, weight, area, temperature, pressure etc.;
- *counting*, if the field describes the frequency or cardinality of a countable set such as: cases, components, events, objects etc.;
- *generation*, where unique keys, identification names, codes, passwords etc. are required for security; the generation is modeled by algorithms that assure the usage of values in the intended purpose;
- *purchase*, if the value is given from outside by placing or taken from other sources as already existing values: name, birth date, address etc.;
- *qualification*, where values are chosen from a predefined set of options to ensure the integrity of formal data: color, sex, marital status, occupation etc.

A table is thus built which, for each element of the class C_i will register the characteristics values k_i, obtaining the data set.

Table 1. Components of C_i class and the descriptive characteristics

	k_1	k_2	...	k_l	...	k_{Mi}
e_{i1}						
e_{i2}						
...						
e_{ij}				v_{ijl}		
...						
e_{iNi}						

Where:
v_{ijl} – the value of k_l characteristic measured for the element e_{ij} from class $C_{i.}$

If N_i, the number of elements of a class is very large, that implies the problem of creating large datasets which should be:

- *complete* in terms of number of elements and number of descriptive features; in quantitative terms, the dataset must include all components and to capture all of the descriptive characteristics, so there is no blank or null elements;
- *accurate* in value; in order for data to be used for their processing results, sets need to record content in accordance with reality; correctness testing involves both the data acquisition methods and the cross-validation of the recorded values;
- *homogeneous* both in structural terms – of the dataset format, and in terms of content – the dataset's values; homogeneity is important for determining other quality characteristics; in addition, the LDC processing is also dependent on a level of homogeneity that is accepted as high enough for calculations;
- *comparable* so that they are available for mutual analysis and processing; comparing sets there is only acceptable in terms of homogeneity, because in certain situations a number of factors affect the evolution of characteristic values, making them incomparable.

Since the quality of data sets is an issue as important as it is sensitive, LDC creation should follow a standard plan – like the one presented in [1], whose steps include:

- defining the datasets by specifying their objectives, the data need and the data sources to be used;
- choosing the descriptive characteristics included in the dataset and building up the list of fields with the format in which they are stored;
- setting the structure of a record or file, by specifying for each describing characteristic the data type and memory length;
- measuring and determination of values for each descriptive characteristic contained in the dataset and validation of values and integration within the limits of the definition of the descriptive characteristics;
- effective introduction of data or acquisition of pre-validated data, horizontally (for a single element is inserted all the features) or vertically (for a single feature to include all elements);
- obtaining the physical form by grouping the describing characteristics and "packaging" them as a set of data (record, file, structure).

The whole process takes into account the software and hardware implications to large data sets and is performed incrementally.

2 Estimating the LDC size

The way of approaching LDC from start to finish should take into account the collection's cardinality along with its physical size. Some of the quality characteristics, as well as the whole general quality index are influenced by the estimated size of the DS collection. This is one of the reasons the research effort is focused on methods of size estimation. Other reasons refer to:

- determining the required space for data storage and estimating the growth rate of data sets;
- justifying the implementation of certain scheme of data distribution, when data sets are stored optimal decentralized;
- adjusting the search engines' dataset processing;
- determining the size of a single set of data and resource requirements for its acquisition, processing, transfer etc.

Some of the estimation methods are based on Capture-Recapture (CR) introduced in [10], using Laplace's approach from 1802 to estimate the population of France. The concept uses the relationship between known and unknown data. CR method extracts a set of random objects from population, marks them and places them back in the collection. Then makes another extraction and counts the objects that were repeated in the two drawings. Based on this number the approximate size of the whole collection of data is determined.

To estimate the relative size of collection of data sets, many methods are based on the following probabilistic model:

- let it be sets A and B, and their intersection $A \cap B$;
- *P(A)* is the probability of an element to belong to set *A;*

- $P(A \cap B \mid A)$ is the probability that the element belongs to $A \cap B$ and in the same time to A;
- then

$$P(A \cap B \mid A) = \frac{|A \cap B|}{|A|}$$

or

$$P(A \cap B \mid B) = \frac{|A \cap B|}{|B|}$$

which follows:

$$\frac{|A|}{|B|} = \frac{P(A \cap B \mid B)}{P(A \cap B \mid A)}$$

For estimation of a collection with N datasets, a sub-set *A'*, with K_i documents is extracted. Then another subset with the same dimensions is extracted, *B'*. If the withdrawals are random, the probability that any document from *B'* was already extracted in *A'* is $\frac{K_i}{N}$.

The probability of having *i* duplicate documents between any two subsets of dimension K_i is:

$$m(i) = \binom{K_i}{i}\left(\frac{K_i}{N}\right)^{i}\left(1 - \frac{K_i}{N}\right)^{K_i - i}$$

The possible values for number of duplicate documents are *0,1,2,...* K_i. So:

$$E(X) = \sum_{i=0}^{K_i} i * m(i) = \sum_{i=0}^{K_i} i * \binom{K_i}{i}\left(\frac{K_i}{N}\right)^{i}\left(1 - \frac{K_i}{N}\right)^{K_i - i} = \frac{K_i^{\,2}}{N}$$

Extending to T subsets, the MCR (Multiple Capture-Recapture) method is implemented. The number of pairs of duplicate documents is:

$$D = \binom{T}{2} E(X) = \frac{T(T-1)}{2} E(X) = \frac{T(T-1)K_i^{\,2}}{2N}$$

And N is estimated as:

$$\hat{N} = \frac{T(T-1)K_i^{\,2}}{2D}$$

The steps of the proposed estimation algorithm are:

- the dimension of the subset is defined – K_i, the number of datasets within a sample; $K_i$must be resonable chosen in order to represent the DS collection;
- the subset *i* is extracted; the K_i datasets are analized and tha number of already marked documents is determined, R_i;
- the unmarked datasets are marked, and the total number of marked documents from collection is retained, M_i;
- the medium phisycal size of unmarked documents is calculated, S_i;
- the previous steps are repeated for a reasonable number of times T (determined in the following section of research);
- the collection's cardinality is estimated (number of datasets):

$$\hat{N} = \frac{\sum_{i=1}^{T} K_i M_i^2}{\sum_{i=1}^{T} R_i M_i}$$

- the phisycal dimension is estimated (storage space measured in B, KB, MB etc.):

$$\hat{S} = \hat{N}\frac{\sum_{i=1}^{T} S_i}{T}$$

The distributed software of cardinality and physical space estimation is built. The algorithm is implemented and verified for K_i=100, T=40, N=100.000, S=4,65GB. The estimation results are displayed in Figure 2.

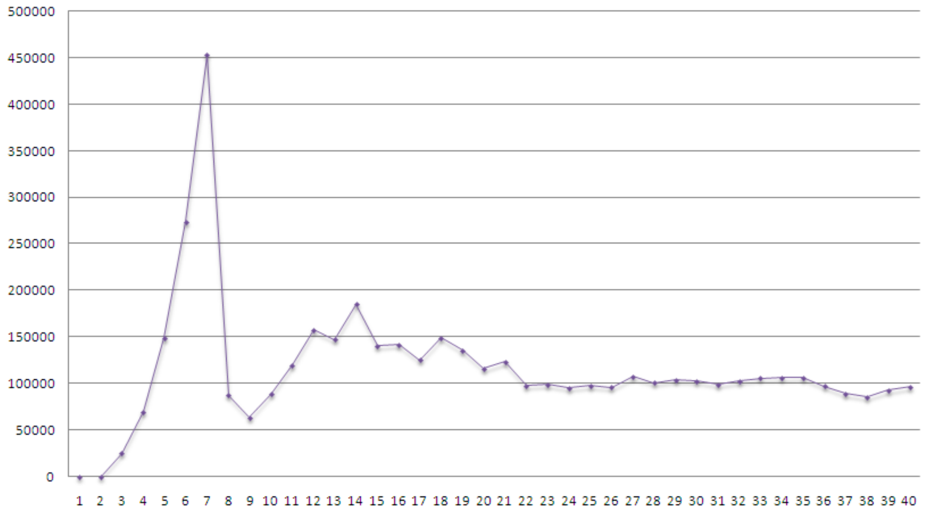

Fig. 2. The evolution of LDC cardinality estimation

The estimations are close after ~20 iterations for K_i=100. It is necessary to know the minimum number of iterations that have to be executed in order to obtain a close estimation. Because the variable depends on both the collection size and the sample size, the evolution of estimation is observed with the software application. The number of iterations is set to 120 while $k_i \in \{120;100;80;60\}$ and $N \in \{100.000;150.000;200.000;250.000;300.000\}$.

For each value, the Table 2 records:

- iteration i_1 from which the estimation doesn't exceed error er_1=±*10%* meaning the value belongs to $[N-10\%;N+10\%]$;
- iteration i_2 from which the estimation doesn't exceed error er_2=±5%, meaning the value belongs to $[N-5\%;N+5\%]$.

Table 2. Estimation evolution for *T* determination

K	*N*	i_1	i_2
120	100.000	18	38
100	100.000	31	51
80	100.000	42	53
60	100.000	64	77
120	150.000	34	50
100	150.000	49	77
80	150.000	53	83
60	150.000	71	90
120	200.000	54	81
100	200.000	67	93
80	200.000	83	101
60	200.000	87	113
120	250.000	64	87
100	250.000	74	101
80	250.000	93	107
60	250.000	96	117
120	300.000	69	94
100	300.000	81	106
80	300.000	99	115
60	300.000	103	118

The bigger collections need more iterations for correct estimation while the dispersion rate is growing proportionally. A mathematical law is determined for defining the minimum number of iterations needed for close estimations. The determination parameters are:

- the chosen size of sample, *K*;
- the collection size *N*;
- the error or the grade of closeness desired – *ERR* giving the collection size (*%N*).

The 40 observations above are included in a multiple regression determined by *Least Squares* method in E-Views. The results are:

```
Variable        Coefficient  Std. Error   t-Statistic  Prob.
C               107.2000     6.014550     17.82344     0.0000
K               -0.570000    0.045994     -12.39281    0.0000
N               0.263500     0.014545     18.11655     0.0000
ERR             -420.0000    41.13865     -10.20938    0.0000
```

The equation follows:

$$I = 107{,}2 - 0{,}57K + \frac{0{,}2635N}{10^3} - 420ERR$$

Where:

I – iteration from which the estimation belongs to the interval;

K – number of datasets within a sample;

N – collection size;

ERR – the interval of estimation as percent of collection size; if the desired interval is $[N - 5\%; N + 5\%]$ then $ERR = 0{,}05$.

Applying the above mentioned formula, in order to estimate a 500.000 datasets collection with an error of 0,01 by sampling 150 datasets at a time, the estimation is correct after a number of iterations:

$$I = 107{,}2 - 0{,}57 * 150 + \frac{0{,}2635 * 5 * 10^5}{10^3} - 420 * 0{,}01 \approx 150$$

The estimation is very closed after extracting at most 22.500 datasets, representing 4,5% of collection size. Observing the anterior 40 records, the estimation is closed after extracting at most 4% from the entire collection, proving the algorithm's efficiency.

3 The quality characteristics system

Giving the estimated collection size, the quality characteristics are evaluated. In [2] the software quality is defined and data quality characteristics are presented. To clarify the quality-related concepts, the following terms must be clearly defined and delimitated:

- *data quality* refers to the extent that existing data are available or suitable for processing, decision taking or resource planning; data quality is defined by the measure in which the reality is captured while data meets the specified form and content requirements;
- *software quality* refers to the extent that a computer application is conform to the design requirements and meets user needs; software quality also characterizes the use of resources and the user interaction through metrics: reliability, versatility, maintainability, security, consistency etc.;
- *management quality* in software development aims at the process of designing and implementing the application, together with all side-activities; quality of software development measures the degree of effectiveness for the activities associated with designing and building computer applications;
- *operation quality* of applications characterizes the way users manipulate the program to achieve the desired results; the operation quality is influenced both by the software quality and the degree of knowledge of options and processes included in application.

So the concept of quality is applied to different aspects involved in working with LDC. Figure 3 shows the relationships between the terms defined above.

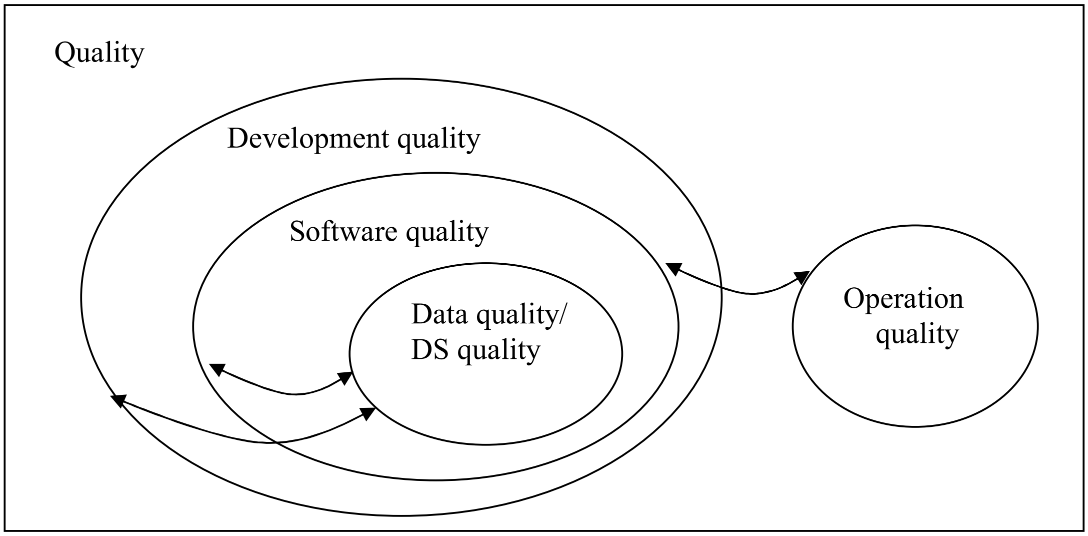

Fig. 3. Quality concepts and the relationships between

The LDC-oriented applications must pursue the following quality characteristics:

- *correctness* or *accuracy* of the dataset that characterizes the proximity of the value/values to the value/values considered to be real or true; the accuracy is achieved when data sets collected by a computer system reflects the real world it intends to represent;
  - o *effects*: in case of poor accuracy, the processing results are incorrect, unrepresentative and therefore unusable;
  - o *influencing factors*: the accuracy of the data set is influenced by the quality of measuring instruments, the dataset format, the state of input devices, communication channels and storage space, and the human factors;
  - o *planning*: the maximum level of accuracy is difficult to obtain because of so many influencing factors; for the level to be acceptable, a schema for planning, realization and control of correctness must be built, by considering the important fields of the data sets, checking measuring instruments, testing and validation of the recorded values and protecting the data after introduction;
- *completeness* refers to the degree to which values are present in the DS collection; in terms of data existence inside a dataset, only two situations are possible: a value is assigned to the characteristic, or the characteristic doesn't take values; the completeness is achieved when all descriptive characteristics of an item are recorded;
  - o *effects*: if the data sets are not complete, they are not available for processing, for planning or decision-making; such data sets are therefore unusable;
  - o *influencing factors:* completeness of data sets is influenced by the existence of a default value for that feature, by algorithms and tests that report fields not entered, by the quality of the input pattern, by the existence of immeasurable fields, by the structure of the dataset and the human factor;
  - o *planning*: designing, managing and achieving the level of completeness includes the construction of signaling mechanisms for incomplete sets, mechanisms for automatic filling of blank fields with default values and structure evaluation for locating immeasurable dataset fields;
- *homogeneity* of DS is a quality characteristic that expresses the degree to which the datasets resemble to one another within the collection; homogeneity is considered in both the structural and content terms (the dataset format and its values);
  - o *effects*: the importance of homogeneity is high because it influences the determination of other quality characteris-

tics; working on LDC is not accepted outside of specified levels of homogeneity;

- *influencing factors*: homogeneity is influenced by the reality recorded by the datasets, the structure of the dataset, the types of data structures used in the description and the stability of input pattern;
- *planning*: designing, managing and achieving optimal level of homogeneity should consider type validations, size limitation of media files and application processes, and standardized data acquisition.

- *reliability* of DS collections requires that data should not contain errors of morphological or syntactic nature which cause system failure;
  - *effects*: if the reliability is not present at the general level of the entire population, data sets will generate errors that will prevent the operation or decision-making processes;
  - *influencing factors*: reliability is directly influenced by the structure of datasets, the method of distributed storage, the communication channels, the volume of data and data consistency;
  - *planning*: designing, managing and achieving optimal level of reliability for LDC needs to consider the distributed storage system (formation of virtual collections of datasets), universal datasets processing, validation and verification of each value for operations participation;
- *maintainability* of LDC characterizes the probability that an incorrect dataset is restored to specified conditions within a timeframe in which maintenance is performed according to procedures; maintainability measures the ability to isolate and fix an error in a dataset in a given time;
  - *effects*: if maintainability is low, data sets are irretrievable, and if their accuracy is poor, the collection must be completely eliminated;
  - *influencing factors*: maintainability of LDC is directly influenced by their structure, degree of value transparency, accessibility and component flexibility;
  - *planning*: designing, managing and achieving optimal level of LDC maintainability should consider the use of flexible data structures, ensuring continuous access to data and identifier storage for each data set separately.

The quality characteristics mentioned above are influencing each other and the procedures for implementing LDC quality have to balance these influences. The Table 3 shows the direction in which quality characteristics are influencing each other.

Table 3. The mutual influence of the quality characteristics

	Accuracy	Completeness	Homogeneity	Reliability	Maintainability
Accuracy	+	0	-	+	-
Completeness	0	+	-	0	0
Homogeneity	-	-	+	0	0
Reliability	+	0	0	+	-
Maintainability	-	0	0	-	+

Where:

0	– no mutual influence;
-	– negative influence (if one rises the other one decreases);
+	– positive influence (if one rises the other also rises).

To mathematically quantify the quality characteristics, indicators and metrics are built. Their value expression allows the generation of models and correlations, and incorporation into a metrics system.

4 Quality metrics for LDC

Software quality is a multidimensional concept. Its professional approach differs greatly from those of the typical user. Quality metrics are abstractions of quality characteristics used for the quantitative expression of a software application status. Building quality metrics aims to:

- measure the quality of existing LDC by discrete expression of the state;
- estimate the quality if the application is in design stage (produce values for cost of quality calculation).

Each index that defines a quality metric is analyzed in relation to three properties: sensitivity, non-compensatory and non-catastrophic character.

Sensitivity is a property that captures the relationship between parameters and results. It points out that any variation of the independent variables cause variations in the values of the dependent variables.

Let *M* be the index whose value is a function of independent variables $x_1, x_2, ..., x_n$.

$$M = f(x_1, x_2, ..., x_n)$$

Variations $\Delta_1, \Delta_2, ..., \Delta_n$ are noticed, with $\Delta_i \neq 0, i = \overline{1,n}$, for each independent variables, and the new index *M'* has the following format:

$$M' = f(x_1 + \Delta_1, x_2 + \Delta_2, ..., x_n + \Delta_n)$$

Index *M* is sensitive if the relation is true:

$$M - M' \neq 0$$

In case:

$$M = \frac{x_1}{x_2}$$

And the two variables are modified with the same amount $k \neq 0$, then the index value

$$M' = \frac{kx_1}{kx_2}$$

will have the same value with the first one, in which case the index is characterized as non-compensatory.

The sensitivity property belongs to software metrics describing indicators that are functional dependent by a number of factors. Among these are:

- DS complexity depending on the number of fields;
- DS completeness depending on the number of fields are missing;
- DS accuracy depending on the number of errors.

The *non-catastrophic character* of a given index is given by the extent to which there are particular values in its components that make impossible to obtain a result [3]. One index is catastrophic if there are situations where the defining mathematical expression is meaningless. Using these indexes should be preceded by a clear definition and analysis of these situations. Taking into account the rules of numeracy, the non-catastrophic character is generated by the situations where:

- denominator of a ratio is 0;
- argument of a logarithmic function is negative or 0;
- value under the radical is negative.

Therefore, the index with format:

$$M = \frac{A}{B}$$

$$M = \log_x y$$

$$M = \sqrt{z}$$

Or any combination of those forms or any expression that includes one of them should be accompanied by restrictions and rules so that they can be calculated:

$$B \neq 0$$

$$y > 0$$

$$z > 0$$

The *non-compensatory* nature of an indicator ensures that variations in the levels of independent variables cause different levels of the result variables. This property is the fundamental assumption of unique statements included in the study. To ensure the representativeness and significance of the results, situations should be avoided where the same results are obtained for different levels of input variables.

Let *M* be the index of a quality metric, with

$$M = x + y$$

Where *x* and *y* are independent variables.

Given the variations Δ_x respectively Δ_y the following index is obtained:

$$M' = (x + \Delta_x) + (y + \Delta_y)$$

For $\Delta_x = -\Delta_y$ the index is

$$M' = x + y + \Delta_x + \Delta_y = x + y + \Delta_x - \Delta_x = x + y = M$$

Which reveals a compensatory character of index *M* because, given the independent variable variations, the same level of index is obtained.

For index:

$$M = \frac{\max(x_1, x_2, ..., x_n)}{\min(y_1, y_2, ..., y_n)} = \frac{x_i}{y_j}$$

The compensatory character is available in case each variable is proportionally modified with the same value *k :*

$$M' = \frac{\max(kx_1, kx_2, ..., kx_n)}{\min(ky_1, ky_2, ..., ky_n)} = \frac{kx_i}{ky_j} = \frac{x_i}{y_j} = M$$

The non-compensatory character of indicators is analyzed for datasets that ensures the property in order to verify the correlation between variation of the independent variables and change in the index values.

For the five quality characteristics – accuracy, completeness, homogeneity, reliability and maintenance – the measurement metrics are defined as follows.

$$I_1 = M_{accuracy} = \frac{Ncv}{Ntv}$$

where:

Ncv – number of correct values;

Ntv – total number of values;

Ncv – is obtained as $Ncv = Ntv - Niv$

Where *Niv* is the number of incorrect values which validates the following relation for at least one field of the dataset

$$\Delta = |v - v'| > 0$$

where:

v – the real value of the field;

v' – the recorded value of the field.

$$I_2 = M_{completness} = \left(\sum_{i=1}^{m} \alpha_i l_i + \sum_{j=1}^{n} \beta_j c_j \right) / \left(n \cdot \sum_{i=1}^{n} \alpha_i + m \cdot \sum_{j=1}^{m} \beta_j \right)$$

$$I_2 = \frac{\sum_{i=1}^{m} \alpha_i l_i + \sum_{j=1}^{n} \beta_j c_j}{n \sum_{i=1}^{m} \alpha_i + m \sum_{j=1}^{n} \beta_j}$$

where:

It is presumed that the dataset is organized as a matrix with *m* lines and *n* columns; to every line and column is assigned an importance coefficient α_i respectively β_j;

l_i – number of elements that are missing from line *i*;

c_j – number of elements that are missing from column *j*;

m – total number of lines;

n –total number of columns;

α_i – importance of line *i*;

β_j – importance of column *j*, with

$$\sum_{i=1}^{m} \alpha_i + \sum_{j=1}^{n} \beta_j = 1$$

$$I_3 = M_{\hom ogeneity} = \frac{Nt}{Nf},$$

where:

Nt – number of definition types (types of data structures) present in the dataset;

Nf – number of fields from the dataset.

$$I_4 = M_{reliability} = R(t) = P(T > t), \quad t \geq 0$$

where:

P – the probability that datasets generate correct results in the time interval between time 0 and time *t*;

T – random variable that defines the failure time.

If the variable *T* has the density function *f(t)* then

$$R(t) = \int_t^{\infty} f(s)ds \text{ or } f(t) = -\frac{d}{dt}[R(t)]$$

$$I_5 = M_{mantainability} = V(t) = P(T \leq t) = \int_0^t g(s)ds$$

where:

T – the variable that describes the repair time or the total idle time (of non-functionality);

G(t) – the density function of variable *T;*

V(t) – maintainability – the probability that the failed system is restored before time *t*.

For the five index mentioned above, each propriety is tested, the results being displayed in the Table 4.

Table 4. The proprieties of the quality metrics index

Characteristic	Index	Sensitive	Non-catastrophic	Non-compensatory
Accuracy	I_1	*		
Completeness	I_2	*	*	*
Homogeneity	I_3	*		
Reliability	I_4	*	*	
Maintainability	I_5	*	*	

Quality indicators in their mathematical form are applied and measured for seven different DS collections with the same number of datasets. Measurement results are presented in Table 5.

$Card(C_i) = 50, i = \overline{1,5}$

Table 5. The measured values for the quality indexes

DS Collection	I_1	I_2	I_3	I_4	I_5
C_1	0,92	0,71	0,7	0,91	0,8
C_2	0,86	0,69	0,6	0,91	0,88
C_3	0,96	0,84	0,8	0,93	0,85
C_4	0,88	0,77	0,7	0,90	0,89
C_5	0,8	0,87	0,5	0,89	0,87
C_6	0,94	0,9	0,5	0,93	0,89
C_7	0,9	0,85	0,6	0,94	0,9

For quantitative expression of a general quality index for DS collections, the five indicators are to be aggregated into a single expression:

$$IGQ = aI_1 + bI_2 + cI_3 + dI_4$$

where $a,b,c,d \in [0,1]$ and $a+b+c+d=1$ so that $IGQ \in [0;1]$.

For the first time, equal weights are assigned to each index:

$$a = b = c = d = 0{,}2$$

So that *IGQ* has the following values for the same DS:

Table 6. The index of general quality

DS Collection	IGQ_1
C_1	0,808
C_2	0,788
C_3	0,876
C_4	0,828
C_5	0,786
C_6	0,832
C_7	0,838

But *IGQ* calculated by the expression used in [4] has different values, leading to differences $\Delta_1 = |IGC_1 - IGC_2|$

Table 7. Differences between the ways IGQ calculation

DS Collection	IGQ_1	IGQ_2	Δ_1
C_1	0,808	0,845	0,037
C_2	0,788	0,82	0,032
C_3	0,876	0,9	0,024
C_4	0,828	0,85	0,022
C_5	0,786	0,805	0,019
C_6	0,832	0,89	0,058
C_7	0,838	0,86	0,022

After refining estimations and statistic calculus, the following estimations are obtained: *a*=0,4; *b*=0,2; *c*=0,1; *d*=0,2; *e*=0,1.
This leads to obtaining IGQ_3

Table 8. IGQ Recalculation

Collection	IGQ_3
C_1	0,842
C_2	0,812
C_3	0,903
C_4	0,845
C_5	0,809
C_6	0,881
C_7	0,868

The second approximation is better, because $\Delta_2 = |IGC_2 - IGC_3|$ is smaller:

Table 9. Differences between the two IGQ estimations

SD Collection	Δ_1	Δ_2
C_1	0,037	0,003
C_2	0,032	0,008
C_3	0,024	0,003
C_4	0,022	0,005
C_5	0,019	0,004
C_6	0,058	0,009
C_7	0,022	0,008

Therefore, the general formula of *IGQ* is:

$$IGC = 0{,}4I_1 + 0{,}2I_2 + 0{,}1I_3 + 0{,}2I_4 + 0{,}1I_5$$

For different DS collections C_{11}-C_{17} the quality indexes are planned and the values are presented in Table 10:

Table 10. The planned values of the quality indexes

DS Collection	I_{1p}	I_{2p}	I_{3p}	I_{4p}	I_{5p}	*IGQ*
C_{11}	0,9	0,7	0,6	0,9	0,9	0,83
C_{12}	0,8	0,6	0,6	0,7	0,8	0,72
C_{13}	0,7	0,8	0,8	0,9	0,8	0,78
C_{14}	0,8	0,8	0,7	0,9	0,9	0,82
C_{15}	0,8	0,8	0,6	0,8	0,8	0,78
C_{16}	0,9	0,9	0,7	0,9	0,8	0,87
C_{17}	0,9	0,8	0,8	0,9	0,9	0,87

For the concrete measurements, the formula is used, with a time of *t=60* days for setting the reliability and maintainability levels. The measured, the estimated and the differences values are cumulated in the table.

Table 11. The planned values, the measured ones and the differences between

	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}	C_{16}	C_{17}
I_{1m}	0,87	0,85	0,66	0,87	0,9	0,88	0,94
I_{1p}	0,9	0,8	0,7	0,8	0,8	0,9	0,9
Δ_1	0,03	0,05	0,04	0,07	0,1	0,02	0,04
I_{2m}	0,77	0,56	0,73	0,85	0,82	0,91	0,84
I_{2p}	0,7	0,6	0,8	0,8	0,8	0,9	0,8
Δ_2	0,07	0,04	0,07	0,05	0,02	0,01	0,04
I_{3m}	0,51	0,64	0,65	0,62	0,67	0,7	0,73
I_{3p}	0,6	0,6	0,8	0,7	0,6	0,7	0,8
Δ_3	0,09	0,04	0,15	0,08	0,07	0	0,07

I_{4m}	0,93	0,84	0,96	0,91	0,89	0,93	0,96
I_{4p}	0,9	0,7	0,9	0,9	0,8	0,9	0,9
Δ_4	0,03	0,14	0,06	0,01	0,09	0,03	0,06
I_{5m}	0,98	0,94	0,92	0,92	0,87	0,91	0,97
I_{5p}	0,9	0,8	0,8	0,9	0,8	0,8	0,9
Δ_5	0,08	0,14	0,12	0,02	0,07	0,11	0,07
IGC_m	0,837	0,778	0,759	0,854	0,856	0,881	0,906
IGC_p	0,83	0,72	0,78	0,82	0,78	0,87	0,87
Δ_{IGC}	0,007	0,058	0,021	0,034	0,076	0,011	0,036

All values corresponding to Δ_{IGQ} are smaller than 0,1 which proves that the estimations of the quality characteristics are true.

5 The stability of quality indexes

In linear systems theory [6], [7], [8], [9] stability is described as a system property to remain in a stationary path as long as it is not affected by any exterior force, and when the action occurs, the system changes its state of stable equilibrium, tending to return in a finite time to a new equilibrium state. If this is not done, meaning that the size of the output has an amplitude variation with increasingly higher value over time, the system is declared unstable.

In systems theory the issues discussed are:

- internal stability, which does not depend on external signals and refers to the free evolution of the analyzed system;
- external stability characterizing the evolution of dynamical systems output when the input is affected by pulse signals.

From the mathematical point of view, a system with only one entrance and exit has the canonical form:

$$\Sigma = (A, b, c^T),$$

where:

A – n*n matrix;

b – n*1 matrix.

A system with above mentioned canonical form:

- is internal stabile if $\exists M > 0,$ so that $\left\| e^{At} \right\| \leq M, \forall t \geq 0$;
- is external stable if $\exists M > 0,$ so that $|h(t)| \leq M, \forall t \geq 0;$

In the system of quality indexes for LDC, stability is defined as property of indexes to vary proportionally with quality factors: small variations of factors lead to reduced variations of index, while large variations of factors lead to significant index variations.

May I be an index depending on factors $x_1, x_2, ..., x_n$

$$I = f(x_1, x_2, ..., x_n)$$

The index is stabile if

$$\frac{\Delta_I}{I} \cong \frac{\overline{\Delta_x}}{\sum_{i=1}^{n} x_i}$$

where :

$\Delta_I = |I - I'|,$ with

$I' = f(x_1 + \Delta_1, x_2 + \Delta_2, ..., x_n + \Delta_n)$ is the index modification;

$\overline{\Delta_x} = \frac{\sum_{i=1}^{n} |\Delta_i|}{n}$ - is the mean modification of factors.

Let S be a system with m indexes. S is:

- totally stabile if $R=1$;
- predominantly stable if $0{,}7 \leq R < 1$;
- partially stabile if $0{,}4 \leq R < 0{,}7$;
- instable $R<0{,}4$.

where

$$R = \frac{nri_s}{m}$$

nri_s – number of stabile indexes.

So, for measuring the stability of the indicators system, it has to be demonstrated that, in conditions of factors linearity, indexes do not differ significantly from one dataset to another.

To test the stability of quality indexes for LDC:

i. data collection C_{11} is considered, from which are extracted:

- samples from the estimation metrics: $A^{11}, A^{12}, A^{13}, A^{14}, A^{15}$;
- samples from the measurement metrics: $A^{21}, A^{22}, A^{23}, A^{24}, A^{25}$;

ii. the values of each index for each sample are estimated, $I_i^{1i}, i=\overline{1,5}$ and the values tabled.

Table 12. The estimated values of sample quality indexes

	I_1^e	I_2^e	I_3^e	I_4^e	I_5^e
A^{11}	0,89	0,82	0,73	0,93	0,95
A^{12}	0,82	0,85	0,69	0,95	0,93
A^{13}	0,91	0,79	0,71	0,91	0,96
A^{14}	0,87	0,81	0,73	0,94	0,94
A^{15}	0,89	0,83	0,71	0,93	0,93

iii. The definition of stability criterion is applied for the estimated indexes:

Table 13. Stability of estimated indexes

	Stable	Unstable
I_1^e	*	
I_2^e	*	
I_3^e		*
I_4^e	*	
I_5^e	*	

iv. The values of each index for each sample are measured, $I_i^{2i}, i=\overline{1,5}$ and the results are tabled:

Table 14. The measured values of sample quality indexes

	I_1^m	I_2^m	I_3^m	I_4^m	I_5^m
A^{21}	0,91	0,83	0,68	0,91	0,95
A^{22}	0,94	0,85	0,71	0,93	0,93
A^{23}	0,87	0,81	0,73	0,95	0,92
A^{24}	0,89	0,87	0,69	0,91	0,91
A^{25}	0,91	0,91	0,73	0,93	0,95

v. The definition of stability criterion is applied for the measured indexes:

Table 15. Stability of measured indexes

	Stabile	Unstable
I_1^m	*	
I_2^m	*	
I_3^m	*	
I_4^m	*	
I_5^m	*	

vi. The value of *R* is calculated and the system stability is defined:

$R_{estimated} = \frac{4}{5} = 0{,}8 \Rightarrow$ the estimation metrics system is predominantly stabile;

$R_{measured} = \frac{5}{5} = 1 \Rightarrow$ the measurement metrics system is totally stabile.

The stability of quality indexes systems is therefore determined by a series of standard steps that are applicable in all cases. Stability of index systems points to a great extent the representativeness of indexes along with the degree of trust in their own results.

6 Conclusions

To obtain the results used in decision-making processes, the LDC must meet a number of quality characteristics. Each of them is measured by metrics and capture different aspects of the sets, expressing in numbers the level of quality. Indicators have to possess certain properties – sensitivity, non-compensatory and non-catastrophic – but stability as well.

Determination of stability shows the index dependence of the factors and characterizes the entire system of indexes. The case study reveals the importance of stability and the differences between the estimation and measurement systems.

References

[1] S. Pavel, "Very-large-datasets Oriented Software Architecture", in *Proc. Annual Conference of Economic Science PhD Students*, Academy of Economic Studies, Bucharest, Romania, May 2009.

[2] I. Ivan, Gh. Nosca, O.Parlogand S. Tcaciuc, *Data Quality*, Bucharest: Inforec Publishing House, 1999.

[3] I. Ivan and C. Boja, *Statistical Methods in Software Analysis*, Bucharest: ASE Publishing House, 2004.

[4] S. H. Kan, "Metrics and models in Software quality engineering", 2nd edition, Addison-Wesley, 2004, pg. 534.

[5] J. Tian, *Software Quality Engineering: Testing, Quality Assurance, and Quantifiable Improvement*, Wiley-IEEE Computer Society Press, 2005, 440 pg.

[6] I. Ivan, C.Boja and C. Ciurea, *Collaborative System Metrics*, Bucharest: ASE Publishing House, 2007.

[7] E. Fridmana and M.Gil, "Stability of linear systems with time-varying delays: A direct frequency domain approach", *Journal of Computational and Applied Mathematics*, Vol. 200, No. 1, March 2007, pp. 61-66.

[8] L. V. Hiena and V. N. Phatb, "Exponential stability and stabilization of a class of uncertain linear time-delay systems", *Journal of the Franklin Institute*, Vol. 346, No. 6, August 2009, pp. 611-625.

[9] A. Arapostathisa and M. E. Broucke, "Stability and controllability of planar, conewise linear systems", *Systems &Control Letters*, Vol. 56, No. 2, February 2007, pp. 150-158.

[10] S. C. Amstrup, T. L. Mcdonald, B. F. J. Manly, *Handbook of Capture-Recapture Analysis*, Princeton University Press, 2005, 296pg.

Ion IVAN has graduated the Faculty of Economic Computation and Economic Cybernetics in 1970. He holds a PhD diploma in Economics from 1978 and he had gone through all didactic positions since 1970 when he joined the staff of the Bucharest Academy of Economic Studies, teaching assistant in 1970, senior lecturer in 1978, assistant professor in 1991 and full professor in 1993. Currently he is full Professor of Economic Informatics within the Department of Computer Science in Economics at Faculty of Cybernetics, Statistics and Economic Informatics from the Academy of Economic Studies. He is the author of more than 25 books and over 75 journal articles in the field of software quality management, software metrics and informatics audit. His work focuses on the analysis of quality of software applications.

Sorin PAVEL has graduated the Faculty of Economic Cybernetics, Statistics and Informatics (2008) and the Masters of Software Project Management (2010) from the Bucharest Academy of Economic Studies. He is currently following the Doctoral School in Economic Informatics, on very large datasets oriented software development and software quality.