A MODEL FOR EVALUATING THE SOFTWARE RELIABILITY LEVEL

Gheorghe NOSCA[1]
PhD, Association for Development through Science and Education, Bucharest, Romania

E-mail: r_g-nosca@yahoo.com

Adriean PARLOG[2]
PhD, Defence National University, Bucharest, Romania

E-mail: adrian_parlog@hotmail.com

Abstract: *The COTS utilization in the software development is one of the nowadays software production characteristics. This paper proposes a generic model for evaluating a software reliability level. The model can be, also, used to evaluate any quality characteristics level.*

Key words: *software reliability; software quality; complex system theory; structure function; graph; simulation; Boolean operator*

1. Theoretical approach

The model is developed using the complex system theory. The software system is made up of some modules, and each module reliability level is known. The model is very useful in case of using COTS.

A complex software system was taken into consideration to build the model, and the following complex system structural properties have been taken into consideration:

P_1– the system is coherent if its functional structure is down up, and each element is important;

P_2 – an element *i*, $i \in \Phi$, is less important if $\Phi(1_i,x) = \Phi(0_i, x)$; $\forall(i,x)$

P_3– a system made up of m components, having the functional structure Φ has the following property:

$$x_1 \wedge x_2 \wedge \ldots \wedge x_m \le \Phi(x) \le x_1 \vee x_2 \vee \ldots \vee x_m \in \{0,1\}.$$

This means that the considered characteristic is bounded as follow:

- down, if all the structural components are optimum;
- upper, if at least one component is optimum;

P_4 - let us consider $K = \{1,2, \ldots, m\}$

$K_0(x)\,) = \{i, x_i = 0\}$
$K_1(x) = \{i, x_i = 1\}$.

A vector X with $\Phi(x) = 1$, having as correspondent $C_i(x)$ is called path.

P_5 – a path is minim if for each $y < x$, $y(i) < x(i)$, $i=1, 2, \ldots, m)$.

In other words, a minim path is a minim succession of elements that assure, for example, the system reliability.

It is taken into consideration the software system functional structure:

$$S = \Phi(x_1, x_2, \ldots, x_m)$$

and it is intended to establish a relationship $R = h(p_1, p_2, \ldots, p_m)$, among the levels of modules characteristics.

Let us consider a system having m components $x_1, x_2, \ldots, x_m$

A Boolean operator T is defined as follow:

- $T(x_i) = \bar{x}_i$, $i = 1, 2, \ldots, m$
- $T(x_1, x_2, \ldots, x_m) = T(x_1)\cdot x_1 \cdot T(x_2) \ldots x_1 \cdot x_2 \ldots x_{m-1} T(x_m) = \bar{x}_1 \vee x_1 \cdot \bar{x}_2 \vee \ldots \vee x_1 x_2 \ldots \bar{x}_m$
- $T(x_1 \vee x_2 \vee \ldots \vee x_m) = T(x_1)\cdot T(x_2) \ldots T(x_m) = \bar{x}_1 \cdot \bar{x}_2 \cdot \ldots \cdot \bar{x}_m$

The following algorithm is attached:

STEP 1: The system is presented as graph, and its structure function is established:

$\Phi = D_1 \vee D_2 \vee \ldots \vee D_m$, where:

$D_1, D_2, \ldots, D_m$ are minimum paths.

STEP 2: Calculate $F_i = D_1 \vee D_2 \vee \ldots \vee D$, $1 \le i \le m$, eliminating fram $D_1, D_2, \ldots, D_{i-1}$ the elements common for the peers (D_1, D_i), (D_2, D_i), …, (D_{i-1}, D_i).

STEP 3: Calculate $T(F_i)$ şi $D_i \cdot T(F_i)$, $1 \le i < m$

STEP 4: Calculate $R = \sum_{i=1}^{m} (D_i \cdot T(F_i))$ the quality characteristic indicator, where are attached $x_i \to p_i$, $\bar{x}_i \to q_i$

As example we consider a software structure shown in Figure 1.

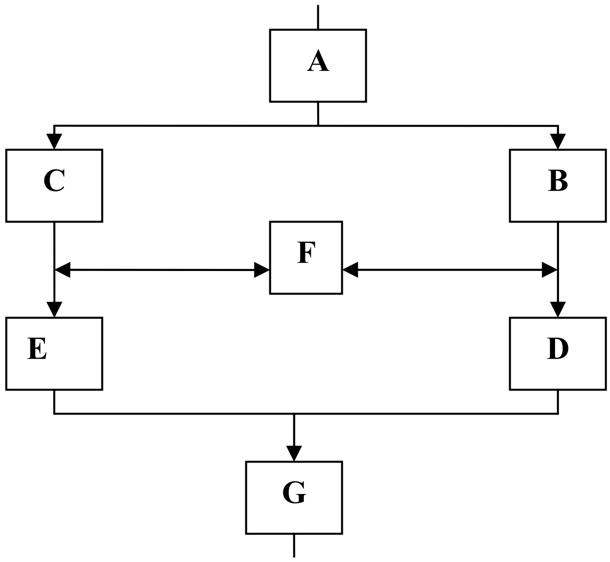

Figure 1. The software strucure

STEP 1: The structure function is shown below:

$\Phi = ABDG \vee ACEG \vee ABFEG \vee ACFDG$

STEP 2: $F_1 = 0$

$F_2 = D_1 =$ ABDG, and are eliminated the common elements from (ABDG, ACEG), that means A and G.

$F_2 = BD$

$F_3 = D_1 \vee D_2$; the common elements from (D_1, D_3); (D_2, D_3) (ABDG, ABFEG); (ACEG, ABFEG), are eliminated, and the result is

$F_3 = D \vee C$

$F_4 = B \vee E \vee BE$

STEP 3: Calculate $T(F_i)$:

$$T(F_1) = 0$$

$$T(F_2) = T \cdot (BD) = \overline{B} \vee B \cdot \overline{D}$$

$$T(F_3) = T(D \vee C) = \overline{D} \cdot \overline{C}$$

$$T(F_4) = T(B \vee E \vee BE) = T(B)T(E) \times T(BE) = \overline{B} \times \overline{E} \times \left(B \times B\overline{E}\right)$$

Calculate, further, $D_i \cdot T(F_i)$

$$D_1 \times T(F_1) = D_1 \times T(0) = D_1 = ABDG$$

$$D_2 \times T(F_2) = D_2 \times \left(\overline{B} \vee B\overline{D}\right) = ACEG\left(\overline{B} \vee B\overline{D}\right)$$

$$D_3 \times T(F_3) = D_3 \times (D \vee C) = ABFEG\left(\overline{D} \cdot \overline{C}\right)$$

$$D_4 \times T(F_4) = ACDFG\left[\overline{B} \cdot \overline{E}\left(\overline{B} \vee B \cdot \overline{E}\right)\right]$$

STEP 4: Calculate the indicator of the considered characteristic, let us say the reliability.

$$R = \sum_{i=1}^{m} D_i \cdot T(F_i) = ABDG + ACEG\left(\overline{B} \vee B\overline{D}\right) + ABFEG(\overline{D} \cdot \overline{C}) + ACDFG\left[\overline{B}\,\overline{E}\left(\overline{B} \vee B\overline{E}\right)\right] =$$
$$= p_A p_B p_D p_G + p_A p_C p_E p_G (q_B + q_B q_D) + p_A p_B p_E p_F p_G q_D q_c + p_a p_C p_D p_F p_G q_b q_E (q_B + q_B q_E)$$

This model is a theoretical base for a simulation model, in order to estimate the reliability of a software complex system. The indicator calculated at STEP 4 is an adimensional indicator of the system characteristic. It is an aggregated indicator obtained taken into consideration the characteristics of the component modules.

2. A simulation algorithm

A simulation algorithm for evaluating the system chosen characteristic level it is presented below.

STEP 1: Initialize the algorithm for generating the random numbers uniform distributed within the interval (0,1).

STEP 2: Initialize the algorithm for generating the coefficients of the modules characteristics. The general characteristic C_G^i can be evaluated.

STEP 3: Generate the level of the component modules characteristics (p_i) comparing C_G^i with $\alpha_{i.}$, where α_i *is* given.

STEP 4: Calculate the characteristic level of each module.

3. Algorithm for calculating the structure function

Let us consider the software system with *m* components. Its attached graph has *k* nodes.

STEP 1: Build the connections matrix, C*(k,k),* attached to the graph.

STEP 2: Add the unit matrix, *I (k,k)*. to the connections matrix.

STEP 3: Eliminate the first column and the last line from C. With the remaining lines and columns is built the determinant *D*, having the rank *k – 1*.

STEP 4: Developing the determinant, it is obtained the structure function.

Let us consider the example from Figure 1, having attached the graph shown in Figure 2.

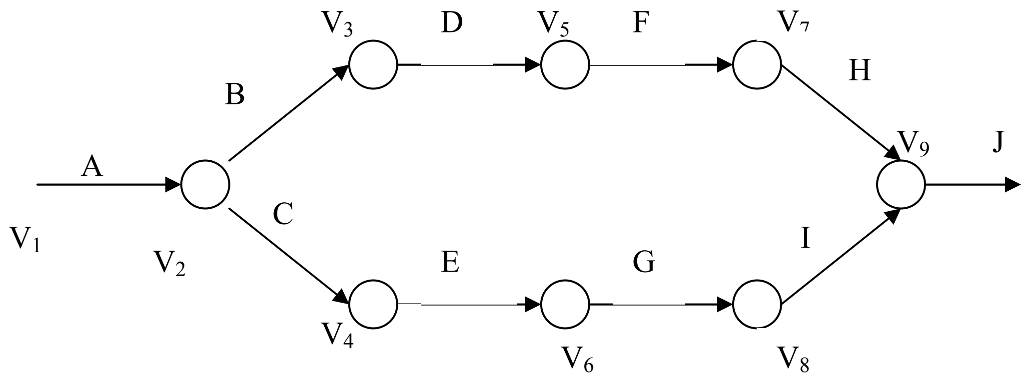

Figure 2. The system structure graph

The graph subcomponents are A, B, ..., J, and its nodes are $V_1, V_2, \ldots, V_9$.

$$C(9,9)=\begin{Vmatrix} 0 & A & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & B & C & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & D & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & E & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & F & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & G & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & H \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & I \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{Vmatrix}$$

Calculate C(9, 9) + I(9, 9)

The result is a matrix with 8 lines and 8 columns. The following determinant is attached to the above mentioned matrix.:

$$\begin{vmatrix} A & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & B & C & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & D & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & E & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & F & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & G & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & H \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & I \end{vmatrix}$$

Develop the determinant

$$A \times \begin{vmatrix} B & C & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & D & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & E & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & F & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & G & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & H \\ 0 & 0 & 0 & 0 & 0 & 1 & I \end{vmatrix} =$$

$$= A \times B \times \begin{vmatrix} 0 & D & 0 & 0 & 0 & 0 \\ 1 & 0 & E & 0 & 0 & 0 \\ 0 & 1 & 0 & F & 0 & 0 \\ 0 & 0 & 1 & 0 & G & 0 \\ 0 & 0 & 0 & 1 & 0 & H \\ 0 & 0 & 0 & 0 & 1 & I \end{vmatrix} + A \times C \times \begin{vmatrix} 1 & D & 0 & 0 & 0 & 0 \\ 0 & 0 & E & 0 & 0 & 0 \\ 0 & 1 & 0 & F & 0 & 0 \\ 0 & 0 & 1 & 0 & G & 0 \\ 0 & 0 & 0 & 1 & 0 & H \\ 0 & 0 & 0 & 0 & 1 & I \end{vmatrix} =$$

$$= A \times B \times D \times \begin{vmatrix} 1 & E & 0 & 0 & 0 \\ 0 & 0 & F & 0 & 0 \\ 0 & 1 & 0 & G & 0 \\ 0 & 0 & 1 & 0 & H \\ 0 & 0 & 0 & 1 & I \end{vmatrix} + A \times C \times \begin{vmatrix} 0 & E & 0 & 0 & 0 \\ 1 & 0 & F & 0 & 0 \\ 0 & 1 & 0 & G & 0 \\ 0 & 0 & 1 & 0 & H \\ 0 & 0 & 0 & 1 & I \end{vmatrix} =$$

$$= A \times B \times D \times \begin{vmatrix} 0 & F & 0 & 0 \\ 1 & 0 & G & 0 \\ 0 & 1 & 0 & H \\ 0 & 0 & 1 & 1 \end{vmatrix} + A \times C \times E \times \begin{vmatrix} 1 & F & 0 & 0 \\ 0 & 0 & G & 0 \\ 0 & 1 & 0 & H \\ 0 & 0 & 1 & I \end{vmatrix} =$$

$$= A \times B \times D \times F \times \begin{vmatrix} 1 & G & 0 \\ 0 & 0 & H \\ 0 & 1 & I \end{vmatrix} + A \times C \times E \times \begin{vmatrix} 0 & G & 0 \\ 1 & 0 & H \\ 0 & 1 & I \end{vmatrix} =$$

$$= A \times B \times D \times F \times \begin{vmatrix} 0 & H \\ 1 & I \end{vmatrix} + A \times C \times E \times \begin{vmatrix} G & 0 \\ 1 & I \end{vmatrix} =$$

$$= A \times B \times D \times F \times H + A \times C \times E \times G \times I$$

Further we propose an algorithm to evaluate the global quality, taken into consideration the characteristics quality.

4. Algorithm for evaluating the global quality

STEP 1: Calculate $C_G^j, \quad j = 1, 2, ..., J_{\max}$, where:

C_G^j - the global quality of the module *j*

J_{max} – the maximum number of iterations for calculating the C_G for a module, taking into consideration the simulated values for the coefficients that appear in the module.

STEP 2: Calculate $C_g^i = \frac{\sum_{j=1}^{J_{\max}} C_G^j}{J_{\max}}$, where *i = 1, 2, ..., m*.

Assuming that the characteristic of the component modules is independent from stochastic point of view, these modules are: operational or non operational.

The following algorithm is used to estimate the general indicator of the system:

STEP 1: Define structure graph attached to the system.

STEP 2: Define the function structure.

STEP 3: $i = 1$

STEP 4: $l = 1$

STEP 5: $k = 1$

STEP 6: Calculate C_G^k

STEP 7: $k = k + 1$; if $k < n$, continue with STEP 6, else STEP 8.

STEP 8: Calculate $C_G^l = \frac{\sum_{k=1}^{n} C_G^k}{n}$

STEP 9: If $C_G^l < \alpha_l$, then $c = \frac{\alpha_l}{C_G^l}$, and generate $u \in (0, ½)$,

else $c = \frac{C_G^l}{\alpha_l}$, and generate $u \in (1/2, 1)$.

STEP 10: If $u < c$, then $x_l = 0$, else $x_l = 1$.

STEP 11: $l = l + 1$; if $l < m$, then continue with STEP 5, else STEP 12.

STEP 12: Calculate the status $S_i = \Phi(x)$.

STEP 13: i = i + 1; if i < n, then continue with STEP 4, else STEP 14.

STEP 14: $R = \frac{\sum_{i=1}^{n} S_i}{n}$

The structured logical schema is presented in Figure 3. The variables have the significations presented in text.

The each module quality characteristic is supposed known. In the case the software is built with reusable components, already tested in use, the characteristic level is known. In other cases there are used methods as experts' judgments, simulation etc.

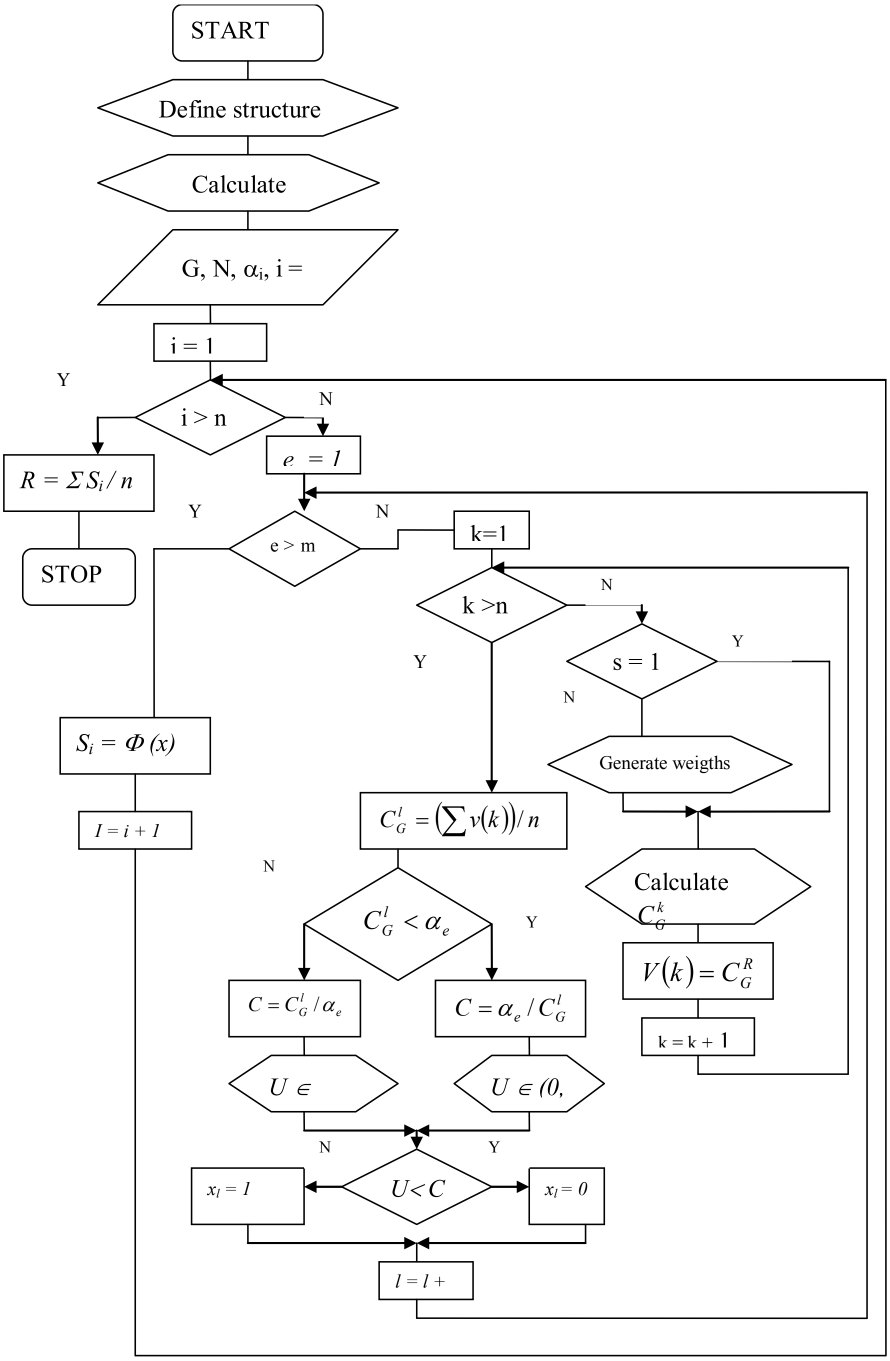

Figure 3. A Simulation Model

In **conclusion** the proposed model is a generic one, that can be utilized to assess the reliability of a complex system made up of modules, and the modules reliabilities are known.

Bibliography

1. Dudewicz, E.J. Mishra, and S.N. **Modern Mathematical Statistics,** John Wiley & Sons, Inc., New York, 1988
2. Elsayed, E. **Reliability Engineering,** Addison Wesley, Reading, MA, 1996
3. Kececioglu, D. **Reliability Engineering Handbook,** Volume 1, Prentice Hall, Inc., New Jersey, 1991
4. Kececioglu, D. **Maintainability, Availability, & Operational Readiness Engineering,** Volume 1, Prentice Hall PTR, New Jersey, 1995
5. Knuth, D.E. **The Art of Computer Programming: Volume 2 - Seminumerical Algorithms, Third Edition,** Addison-Wesley, 1998
6. L'Ecuyer, P. **Communications of the ACM,** Vol. 31, pp.724-774, 1988
7. L'Ecuyer, P. **Proceedings of the 2001 Winter Simulation Conference,** pp.95-105, 2001
8. Mann, N.R., Schafer R.E. and Singpurwalla N.D. **Methods for Statistical Analysis of Reliability and Life Data,** John Wiley & Sons, Inc., New York, 1974
9. Meeker, W.Q. and Escobar L.A. **Statistical Methods for Reliability Data,** John Wiley & Sons, Inc., New York, 1998
10. Mettas, A. **Reliability Allocation and Optimization for Complex Systems,** Proceedings of the Annual Reliability & Maintainability Symposium, 2000
11. Nelson, W. **Applied Life Data Analysis,** John Wiley & Sons, Inc., New York, 1982
12. Nosca, Gh. **Metode si tehnici de optimizare a costului calitatii produselor program,** PhD Dissertation, Academy of Economic Studies, Bucharest, 2003
13. Parlog, A.. **Sistem de modele de simulere,** PhD Dissertation, Higher Military Studies Academy, Bucharest, 1994
14. Peters, E.E. **Fractal Market Analysis: Applying Chaos Theory to Investment & Economics,** John Wiley & Sons, 1994
15. Press, W.H., S.A. Teukolsky, W.T. Vetterling and B.R. Flannery, **Numerical Recipes in C: The Art of Scientific Computing, Second Edition,** Cambridge University Press, 1988
16. ReliaSoft Corporation, **Life Data Analysis Reference,** ReliaSoft Publishing, Tucson, Arizona, 1997
17. Tillman, F.A., Hwang, C.L. and Kuo, W., **Optimization of Systems Reliability,** Marcel Dekker, Inc., 1980

---

[1] Gheorghe Nosca graduated Mechanical Faculty at Military Technical Academy in 1981, and Cybernetics, Statistics and Informatics Economics Faculty at Academy of Economics Studies in 1992.
He obtained his PhD degree in Economics, Cybernetics and Statistics Economics specialty in 2003.
He is currently researcher at Association for Development through Science and Education.
He has published (in co-operation) 3 books, 16 articles in informatics journals.
He has taken part in about 20 national and international conferences and symposiums.
His research interests include data quality, data quality management, software quality cost, informatics audit, and competitive intelligence.

[2] Adriean Parlog graduated Economic Cybernetics and Statistics Faculty, specialized in economic cybernetics at the Academy of Economics Studies, in 1979. He obtained his Ph.D. degree in Military Science in 1994. He has also graduated post academics courses in Romania and abroad, specialized in information quality management, conflict prevention and diplomacy. He is currently associate professor at Defense National University and guest professor at Lucian Blaga University and at the National School of Political and Administrative Studies. He has published two books, more than 30 articles in scientific journals. He has taken part in about 30 national and international conferences and symposiums. He has taken part in designing and developing very complex Information Systems. His research interests include information system development, data quality, data quality management, modeling, simulation and optimization.