A MODEL FOR EVALUATING THE SOFTWARE RELIABILITY LEVEL

Gheorghe NOSCA
PhD, Association for Development through Science and Education, Bucharest, Romania
E-mail: r_g-nosca@yahoo.com

Adriean PARLOG
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 $\Phi(1, x) = \Phi(0, x) \quad \forall (i, x)$
- $P_3$ – a system made up of $m$ components, having the functional structure $\Phi$, has the following property:

$$x_1 \land x_2 \land \ldots \land x_m \leq \Phi(x) \leq x_1 \lor x_2 \lor \ldots \lor x_m \in \{0,1\}.$$ 

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;
P₄ - 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.

P₅ – 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 \( R = h(p_1, p_2, \ldots, p_n) \), 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 follows:

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

The following algorithm is attached:

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

\[
\phi = D_1 \lor D_2 \lor \ldots \lor D_m, \text{ where:}
\]

\( D_1, D_2, \ldots, D_m \) are minimum paths.

**STEP 2:** Calculate \( F_i = D_1 \lor D_2 \lor \ldots \lor D_i \), \( 1 \leq i \leq m \), eliminating from \( D_1, D_2, \ldots, D_i \) the elements common for the peers \( (D_1, D_i), (D_2, D_i), \ldots, (D_{i-1}, D_i) \).

**STEP 3:** Calculate \( T(F_i) \), \( 1 < i < m \)

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

As example we consider a software structure shown in Figure 1.
STEP 1: The structure function is shown below:

\[ \Phi = ABDG \lor ACEG \lor ABFEG \lor ACFDG \]

STEP 2:  
\[ F_1 = 0 \]
\[ F_2 = D_1 = ABDG, \text{ and are eliminated the common elements from } (ABDG, ACEG), \text{ that means } A \text{ and } G. \]
\[ F_2 = BD \]
\[ F_3 = D_1 \lor D_2; \text{ the common elements from } (D_1, D_2); (D_2, D_3) \]
\[ (ABDG, ABFEG); (ACEG, ABFEG), \text{ are eliminated, and the result is } \]
\[ F_3 = D \lor C \]
\[ F_4 = B \lor E \lor BE \]

STEP 3: Calculate \( T(F_i) \):
\[
T(F_1) = 0
\]
\[
T(F_2) = T \cdot (BD) = \overline{B} \lor B \cdot \overline{D}
\]
\[
T(F_3) = T(D \lor C) = \overline{D} \cdot \overline{C}
\]
\[
T(F_4) = T(B \lor E \lor BE) = T(B)T(E) \times T(BE) = \overline{B} \times \overline{E} \times (B \times B \overline{E})
\]

Calculate, further, \( D_i \times T(F_i) \)
\[
D_1 \times T(F_1) = D_1 \times T(0) = D_1 = ABDG
\]
STEP 4: Calculate the indicator of the considered characteristic, let us say the reliability.

\[
R = \sum_{i=1}^{m} D_i \times T(F_i) = ABDG + AEG(B \lor B\overline{D}) + ABFEG(D \lor \overline{C}) + ACDFG(B \cdot E(B \lor B\overline{E}))
\]

\[
\]

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 \( \alpha_i \) 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.
The graph subcomponents are A, B, ..., J, and its nodes are V₁, V₂, ..., V₉.

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

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:

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

Develop the determinant

Figure 2. The system structure graph
\[
\begin{align*}
\begin{bmatrix}
B & C & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & D & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & E & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & F & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & G & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & H \\
0 & 0 & 0 & 0 & 0 & 1 & I
\end{bmatrix}
&= A \times
\begin{bmatrix}
D & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & E & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & F & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & G & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & H & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & I
\end{bmatrix} \\
&= A \times B \times
\begin{bmatrix}
1 & E & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & F & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & G & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & H & 0 & 0 \\
0 & 0 & 0 & 1 & I & 0 & 0
\end{bmatrix} + A \times C \times
\begin{bmatrix}
1 & E & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & F & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & G & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & H & 0 & 0 \\
0 & 0 & 0 & 1 & I & 0 & 0
\end{bmatrix} \\
&= A \times B \times D \times
\begin{bmatrix}
0 & F & 0 & 0 & 0 \\
1 & 0 & G & 0 & 0 \\
0 & 1 & 0 & H & 0 \\
0 & 0 & 1 & 1 & 0
\end{bmatrix} + A \times C \times E \times
\begin{bmatrix}
1 & F & 0 & 0 \\
0 & 0 & G & 0 \\
0 & 1 & 0 & H \\
0 & 0 & 1 & I
\end{bmatrix} =
\end{align*}
\]
Further we propose an algorithm to evaluate the global quality, taken into
collection the characteristics quality.

4. Algorithm for evaluating the global quality

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

- $C_G^j$ – the global quality of the module $j$
- $J_{\text{max}}$ – 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 $C_G^i = \frac{\sum_{j=1}^{J_{\text{max}}} C_G^j}{J_{\text{max}}}$, 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 $C_G^i = \frac{\sum_{k=1}^{n} C_G^k}{n}$

**STEP 9**: If $C_G^i < \alpha_i$, then $c = \frac{\alpha_i}{C_G^i}$, and generate $u \in (0, \frac{1}{2})$,
else $c = \frac{C_G^i}{\alpha_i}$, 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.

\[ R = \sum \frac{S_i}{n} \]

**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

13. Parlog, A. Sistem de modele de simulare, PhD Dissertation, Higher Military Studies Academy, Bucharest, 1994

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 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 written 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.