

---

# SOFTWARE RELIABILITY. NON-PROBABILISTIC APPROACH

Dmitry A. Maevsky, Helen D. Maevskaya, Alexander A. Leonov

•  
Odessa National Polytechnic University, Odessa, Ukraine  
e-mail: [toe-onpu@ukr.net](mailto:toe-onpu@ukr.net)

## ABSTRACT

The article describes the main provisions of the new theory of software reliability, which is not based on probability theory and the theory of non-equilibrium processes. Emerging from the operation of software systems, defects are considered as the result of the forward and reverse defect flows. Relations are developed to predict the number of identified and introduced to system defects and they are opening the possibility of modeling the reliability of software systems, taking into account the secondary defects. It is shown that the majority of existing software reliability models can be derived from the provisions of the dynamics of software systems.

## 1 INTRODUCTION

Software reliability is the most confusing and intriguing area of general reliability theory. On the early stages of its development this theory was based on the probabilistic reliability concepts. The main features of software reliability are: stochastic nature of failures, time dependence of failures and independence of failures from other ones. However, various attempts to create a single universal model that describes defects exposure law on this conception have failed. Now there are more than twenty different models that are trying to describe the same physical process – software defects exposure. Naturally, such diversity shows that this theory requires a thorough revision.

One of the most influencing reliability experts Igor Ushakov wrote (Ushakov 2006): “Errors caused by software have no stochastic nature: they will repeat as soon as some conditions will be repeated. Errors of software, in a sense, are not “objective” – they depend on type of operations, type of inputs and, at last, on type of users”. And later: “... attempts to put “hardware reliability shoes” on “software legs” are absolutely wrong and, moreover, will lead only to a logical dead end”.

Other opinion through software reliability is: “It should be stressed that so far the theory of software reliability can’t be regarded as an established science. ... one can ascertain the presence of a substantial gap between theory (mathematical models and methods) and practice” (Kharchenko et al. 2004).

Six years ago, in 2012, Igor Ushakov wrote (Ushakov 2012): “One thing is clear: software reliability specialists should distinguish their reliability from hardware reliability, develop their own non-probabilistic and non-time dependent mathematical tools”.

This article is devoted to the new non-probabilistic approach to the software reliability problem.

## 2 DYNAMIC THEORY OF THE SOFTWARE SYSTEMS: FUNDAMENTALS

Theory of the Software System Dynamics (SSD) considers a software system (SS) as an open non-equilibrium system that interacts with the environment. Subject area of the SS is considered as the environment. Non-equilibrium system is a system which has gradients of certain properties of

the system, such as concentration, temperature, etc. In SSD the number of defects in the systems at any given time is considered as such property. In the general theory of non-equilibrium processes of physical nature of the subject matter of these properties, which are called "thermodynamic potentials" (Onsager 1931), does not matter. The only important thing is that their gradients that play a role of forces exist in the system. Under the influence of these forces there are flows that are designed to bring the system to equilibrium with its environment. The dynamics of such a system are determined by the spatio-temporal distribution of these flows, with their values at each physical point.

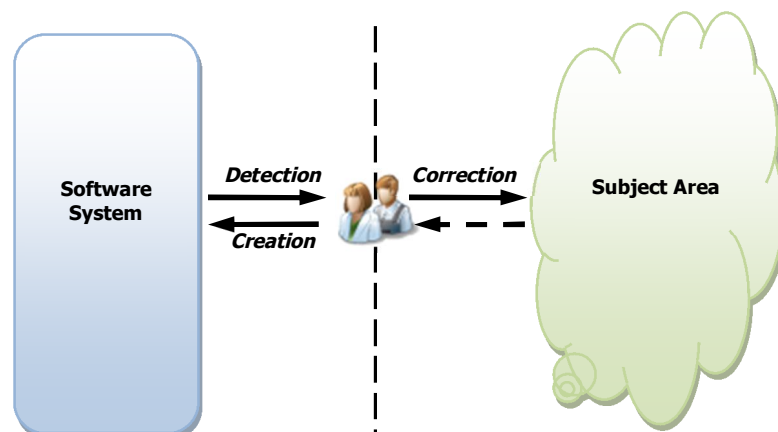
For the SS the concept of "space area" is possible only in the sense of "within" or "outside" the system and the notion of a physical point generally cannot be used. Therefore, with respect to the SS one can only talk about the patterns of distribution of flows over time. The openness of the SS is determined by the nature and extent of its relationship with the environment, which serves as the subject area of the system, and the level of equilibrium is determined by the number of defects contained in the system. In this case subject area itself is accepted as the etalon, that is, by definition it does not contain defects.

Let's denote the number of defects contained in the SS at the specific time  $t$  as  $f$  or  $f(t)$ .

SSD is based on the following hypothesis:

1. SS is an open non-equilibrium system that interacts with its subject area according to the laws of the non-equilibrium processes.
2. The state of the SS is characterized by a special state function – the number of the defects contained in it.
3. Disappearance and appearance of defects in the SS is the result of the joint action of the direct (output) and backward (incoming) defect flows.
4. The intensity of each flow is proportional to the number of defects, forming the flow.
5. All defects are equal and participate in the formation of the flow in the same way, regardless of the causes, location, and type of defect and the possible consequences of its manifestation (the principle of equality).
6. Function  $f(t)$  is differentiable on the whole domain (the principle of continuity).

The basic concept SSD is the concept of software defect flows. Each defect is seen as an integral part of the total flow, which obeys not the laws of the theory of probability but the laws of emergence and evolution of the flows in non-equilibrium systems. Emergence of the defect flows in the SS is shown at Figure 1.



**Figure 1.** Defect emergence in the SS.

During operation of the SS defects lead to the fact that result, which produces its software, does not meet the outcome that is expected by subject area. This discrepancy is detected by the user which is in contact on the one hand with the SS, on the other – with its subject area. Thus, the user acts as the first, error detector, and secondly – a sort of "contact surface" between the SS and its

subject area. We assume that the user is ideal, that is, detects and records each defect at the time its manifestation.

In the process of fixing the defect disappears from the SS due to changes made in its code. This loss can be considered as a result of the removal of defects from the SS. Considering this process in time, we have the flow of defects from the SS out through the "contact surface" - the user. These streams are shown in Figure 1 are shown by arrows "Detection" and "Correction".

In the process of fixing defects in the SS is possible to introduce additional "secondary" defects. The process of introduction of the secondary defect may be regarded as the second, counter-flow of defects, which operates in the direction from the subject area to the SS.

The flow of defects will be numerically characterized by the speed (intensity) of the flow, which can be determined by hypothesis 6 (principle of continuity). Taking into account only the output stream, SS is characterized by a number of defects, which are contained in the system – coordinate  $f(t)$ . It can be considered as having only one degree of freedom and is described by the differential equation of first order. In the case of taking into account of the second process - introduction secondary defects, its coordinate is their current number –  $f_2(t)$ . In general, taking into account both processes we obtain two coordinates, which characterize the effect of defects in a software system –  $f_1(t)$  and  $f_2(t)$ . SS in this case should be considered as a system with two degrees of freedom and described by differential equations of second order. On this basis, we introduce the concept of the order of the SS model.

**Definition.** The order of the SS model is the order of a differential equation, which in this model describes the variation of the number of defects over time.

In the non-equilibrium dynamics by the flux vector  $\mathbf{j}$  of some value  $f$  we consider a vector, whose modulus is equal to the value  $f$  transferred during a unit of time through the unit area  $dS$  perpendicular to the direction of the transfer  $x$ :

$$j = \frac{df}{dt \cdot dS},$$

and the direction – the same as the direction of transport [6]. The very flow of value  $f$  in this case corresponds to the integral

$$J = \int_S \mathbf{j} \cdot d\mathbf{S} = \frac{df}{dt}.$$

This expression makes it possible to avoid the use of the concept of area, indeterminable for the SS.

In reliability theory, the value of  $j$  has a corresponding failure rate  $\lambda$ :

$$\lambda = \frac{df}{dt}.$$

### 3 DEFECT FLOW IN THE FIRST-ORDER SS MODEL

The one-dimensional system – is the simplest case SSD. It is assumed by almost all currently available software reliability models, based on the traditional theory of reliability. Therefore, walkthrough of the dynamics of the SS will start with such one-dimensional case.

We assume that the software system has only the direct flow of defects, ie, the flow is directed from the SS. The statement of the uniqueness of the flow is equivalent to the following two assumptions of most well-known software reliability models (Lyu 1996):

- when an error occurs it is corrected before the discovery of the next;
- new defects are not introduces during the fixing of existing ones.

Indeed, the first assumption is actually equivalent to the presence of the defects flow (they are fixed, that is, cease to exist, are removed from the PC), and the second assumption says that there is no flow of secondary defects.

The flow of any scalar value in a non-equilibrium system only occurs by the action of the driving forces behind this flow (Prigogine 1991). As a driving force in continuous systems there is a gradient of the potential of the corresponding value, and in discontinuous – the difference of potentials at the contact boundary. SS, as shown in Figure 1, should be seen as a discontinuous - there are defects within the system, and in the environment they are absent. From this it follows that the potential at the contact boundary changes abruptly. Given the lack of defects in the external environment, we can take the potential of defects of this medium to be zero. Then, according to [9], the flow of primary defects, being the value of  $f_I$ , can be represented as:

$$\frac{df_I}{dt} = -G_I \cdot \varphi_I, \quad (1)$$

where  $G_I$  – aspect ratio, and  $\varphi_I$  – the potential of the defects in the SS. "Minus" sign in the formula (1) says that the flow is directed toward decreasing the potential, that is, from the SS to the external environment.

Between potential  $\varphi_I$  and its corresponding value of  $f_I$  there is a relation

$$f_I = C_I \cdot \varphi_I, \quad (2)$$

where ratio  $C_I$  will be called as a defect capacity of the system regarding to the value  $f_I$ . Therefore, considering (2), defined

$$A_I = \frac{G_I}{C_I},$$

the equation (1) can be represented as

$$\frac{df_I}{dt} = -A_I \cdot f_I. \quad (3)$$

Let us explain the physical meaning of the coefficients  $G_I$  and  $C_I$  for the SS. In the theory of non-equilibrium processes the coefficient  $G_I$  is called the conductivity of the system with respect to value  $f_I$ . From equation (1) it follows that for a constant value  $\varphi_I$  the rate of detection of the primary defects is directly proportional to size  $G_I$ . In the real SS rate of detection of defects is directly proportional to the frequency of calls to the system. Therefore, the conductivity  $G_I$  in (1) can be interpreted as the frequency of user calls to the SS. In this case we mean an "ideal user", each time specifying a different, in general case random set of input data. In fact, approaching the ideal can be considered as staff members, each member of which works with its narrow set of data sets. Thus, as the conductivity  $G_I$  in the SS we take the rate of the access, and conductivity itself has a dimension of  $s^{-1}$ . Potential  $\varphi_I$ , because of this, must be dimensionless.

Defect capacity  $C_I$  shows how a number of the defects in the SS should increase to that their potential  $\varphi_I$  grows by one. Keeping in mind that  $\varphi_I$  is dimensionless; defect ration is dimensionless as well. Defect ration of the SS can be understood as the maximum possible defect number which can be contained in the analyzed system.

Equation (3) is a homogeneous linear differential equation. Its solution can be obtained in the form

$$f_I(t) = F_0 \cdot e^{-A_I \cdot t}, \quad (4)$$

where  $F_0$  – initial number of defects in SS at the start of research.

According to the formula (4) the number of defects that remain to SS at the time  $t$  shall be calculated. As shown (Lyu 1996), the most convenient for experimental determination of the dependence of the total number defects identified in the system at the same time (cumulative number of defects  $\mu$ ). To calculate  $\mu$  we can use

$$\mu(t) = \int_0^t \lambda(\tau) d\tau, \quad (5)$$

therefore:

$$\mu(t) = F_0 - F_0 \cdot e^{-A_1 t}. \quad (6)$$

Note that the formula (4) and (6) fully comply with similar expressions for the most famous models of software reliability. This suggests that these models are consistent with the theory of first-order SSD.

#### 4 DEFECT FLOWS IN THE SECOND-ORDER SS MODEL

In the case taking into account the secondary flow of defects, SS has two degrees of freedom and is characterized by two coordinates – the number of defects  $f_1$ , which will be removed from the system and the number of the secondary defects  $f_2$ . The connection between the flows of primary and secondary defects is represented by the system of equations:

$$\begin{cases} \frac{df_1}{dt} = -G_{11} \cdot \varphi_1 - G_{12} \cdot \varphi_2 \\ \frac{df_2}{dt} = -G_{21} \cdot \varphi_1 - G_{22} \cdot \varphi_2 \end{cases}. \quad (7)$$

In this system,  $\varphi_1$  – the potential of removed defects, and  $\varphi_2$  – potential for insertion of the secondary. Ratios  $G_{11}$  and  $G_{22}$  characterize the influence of potentials  $\varphi_1$  and  $\varphi_2$  on the flows related to them. By analogy with previous statements, these factors play a role of conductivity and characterize the frequency of accesses to the system. The frequency of entering the secondary defects into software system tends to be lower than the frequency of detection of the primary. On this basis, it can be said that  $G_{11} > G_{22}$ . We call these ratios the intrinsic conductivities of the SS.

Ratios  $G_{12}$  and  $G_{21}$  characterize the influence of potentials  $\varphi_1$  and  $\varphi_2$  on the flows related to them. According to the Onsager symmetry principle (Onsager 1931), these cross-effects are the same, which leads to the equality  $G_{12} = G_{21}$ . Ratios  $G_{12}$  and  $G_{21}$  will be called mutual conductivities.

Potentials  $\varphi_1$  and  $\varphi_2$  are associated with the corresponding values of  $f_1$  and  $f_2$  by the relations of the form (2):

$$f_1 = C_1 \cdot \varphi_1; f_2 = C_2 \cdot \varphi_2,$$

where  $C_1$  – defect capacity of the SS related to the primary defects, and  $C_2$  – defect capacity of the same system related to the secondary ones. Obviously, if we are talking about the same system, then these two should be equal:  $C_1 = C_2$ .

Using the relation between the number of defects and the corresponding potential, taking into account the equality  $G_{12} = G_{21}$  and defining

$$A_1 = \frac{G_{11}}{C_1} = \frac{G_{22}}{C_2}, A_2 = \frac{G_{12}}{C_1} = \frac{G_{21}}{C_2},$$

system (7) can be re-written as:

$$\begin{cases} \frac{df_1}{dt} = -A_1 \cdot f_1 - A_2 \cdot f_2 \\ \frac{df_2}{dt} = -A_2 \cdot f_1 - A_1 \cdot f_2 \end{cases}. \quad (8)$$

The system (8) is an autonomous system of differential equations whose solution is fully determined only by the initial conditions and to determine the time variation of the existing substation primary and secondary defects.

It should be noted that the flow described by the first equation of the system is the flow of defects **carried out** from this SS, rather than the primary flow. In fact, trapped in a system of secondary defects are indistinguishable but the primary and along with them are removed from SS (Lyu 1996). In this sense we can say that the division of defects existing in the SS to primary and secondary is purely arbitrary. They differ only in the moment of introduction, but impact on the state of SS in the same way.

The solution of (8) for the outgoing stream of defects is an expression

$$f_1 = F_0 \cdot e^{-A_1 t} \cdot \cosh(A_2 t). \quad (9)$$

Comparing (9) with (4) obtained for the output stream of defects without a countering input flow, we can see that it differs by the presence of the factor  $\cosh(A_2 t)$ , whose role is to adjust the output stream of defects by the countering flow of the secondary ones.

To interpret and analyze the results, Figure 2 shows plots of the number of defects that remain in the system from time to time for different ratios  $k=A_2/A_1$ . These curves are plotted for a hypothetical software system with the following parameters: initial number of defects  $F_0=100$ , value of the ratio  $A_1=100 \text{ days}^{-1}$ , ratio  $k$  varies from 0 to 1,1. Here  $k=0$  corresponds to the complete absence of the secondary flow of defects, and the value  $k=1$  – case, where the correction of one of the primary defect is accompanied by the introduction of a second. For values of  $k>1$ , the number of secondary defects exceeds the number of fixed ones.

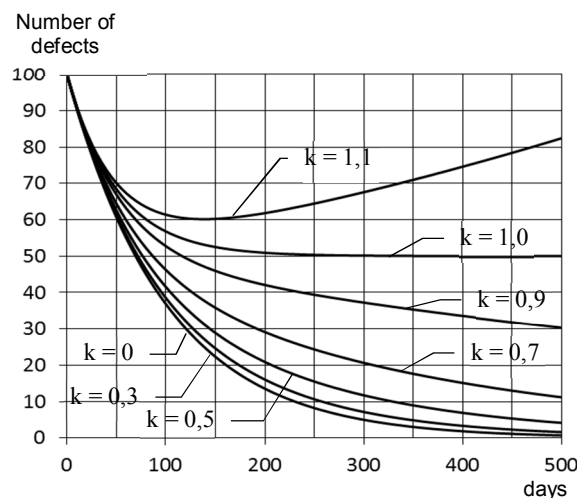


Figure 2. The relation of the number of defects in the SS through time

Analyzing the relations following conclusions can be made:

- In the absence of secondary defects ( $k=0$ ), formula (9) coincides with formula (4), obtained without regard to their influence. This coincidence with reality, which may indicate the correctness of the basic statements of the SSD.

- The influence of secondary defects reduces to increasing the decay time of the output flow. Thus, the SSD theory confirms intuitive assertion that in case of introduction of the secondary defects to SS, the total time of their identification increases.

- With  $k=1$  the number carried out from SS defects stabilizes and tends to the value of  $F_0/2$ . Non-evident interpretation of this fact we will give later.

The solution of (7) for output flow of defects is an expression



$$f_2 = -F_0 \cdot e^{-A_1 t} \cdot \sinh(A_2 t). \quad (10)$$

The sign of "minus" in (10) can be explained on the basis of differences in directions of output and input flows. However, given the fact that the number of defects cannot be negative, in the future, when determining the number of secondary defects "minus" sign will be omitted.

Figure 3 shows plots of relation  $f_2(t)$ , built for the same hypothetical SS for different values of the coefficient  $k$ . Analyzing the relations presented in Figure 3, the following conclusions can be made:

- With  $k=0$  there is no secondary defects flow.
- With  $0 < k < 1$  the number of secondary defects being introduced in the SS has a maximum, which is expressed the more the bigger value of  $k$  is.
- The rate of increase of the number of secondary defects is most important at the initial stage, before reaching the maximum. After that, the number of secondary errors tends to zero, but with a much slower rate.
- With  $k=1$  number of secondary of defects decreases with time, which corresponds to processes in the real SS, and can serve as a confirmation of the SSD.
- With  $k=1$  the number of defects introduced into the SS is stabilizing and tends to the value  $F_0/2$ .

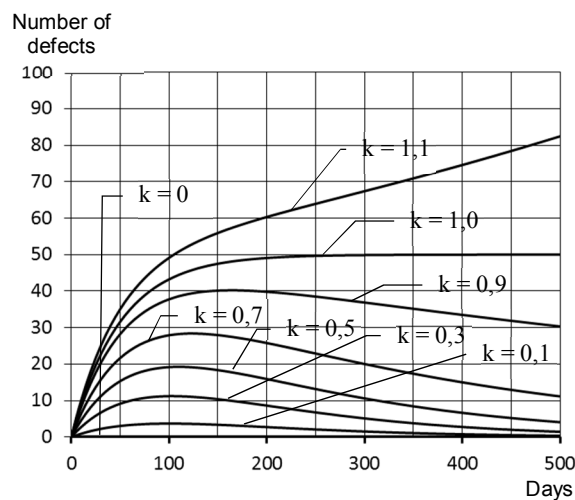


Figure 3. The relation of the number of secondary defects in the SS through time

At any arbitrary point in time the number present in SS of defects can be calculated as the sum of the number of defects that will be removed from it ( $f_1$ ) and of the number of the already introduced secondary defects ( $f_2$ ). For a plot of this relation we have to simply sum the corresponding curves from the plots in Figure 2 and 3. The result of this adding is shown in Figure 4. As can be seen from Figure 4, provided  $k = 1$ , ie, when the number of introduced secondary defects equals the number of corrected, the residual amount of defects in the SS remains unchanged. Now it is clear why, when  $k = 1$ , the values of  $f_1(t)$  and  $f_2(t)$  tend to the value  $F_0/2$ . Indeed, in this case their sum is at any given time is equal to the initial number of defects –  $F_0$ , which fully corresponds to the physical representations of the processes that must occur in SS at a given condition.

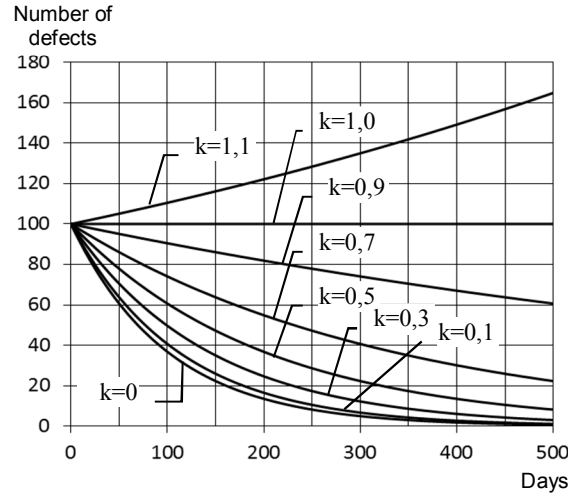


Figure 4. The relation of the total number of defects in the SS through time

## 5 RELIABILITY MODEL BASED ON THE SSD

For the Software Reliability Model (SRM) creation we have to define a set of input data, write the mathematical relationships that define the reliability and bring the method of determining the coefficients in these dependences.

As input data of the model developed on the basis of SSD time series giving the cumulative number of detected defects are adopted. Despite the fact that the defects form flows with well-defined laws, the process of their identification has a significant uncertainty (Kharchenko at al. 2004). Therefore, consideration for modeling of each individual defect complicates the analysis of the results by having a considerable "noise". Due to this, SRMs, input data of which are points of identification of every defect, cannot ensure the accuracy of the simulation, due to the fact that their input data are already inaccurate. Time series, forming a cumulative number of defects is more accurate, because a random registration or non-registration of each specific defect cannot affect the overall trends in this series. In fact, the time series formed by the cumulative number of defects is relieved of the random component and is a trend.

On the basis of foregoing, for the construction of the model mathematical relationships derived from the theory of SSD should be converted to operate with cumulative defects trends.

On the basis of the formula (5), for a cumulative trend of output flow we obtain the expression:

$$\lambda_1(t) = \int_0^t \frac{F_0}{2} [(A_1 + A_2) \cdot e^{(A_2 - A_1)t}] + \int_0^t \frac{F_0}{2} [(A_1 - A_2) \cdot e^{-(A_2 + A_1)t}] dt. \quad (11)$$

It is not difficult to see that having  $A_1 = A_2$  primitive of  $\lambda_1(t)$  does not exist, since the difference between the  $A_1$  and  $A_2$  is zero. Therefore, finding the cumulative of the number of defects of the original flow we consider separately for the two cases.

Case 1.  $A_1 \neq A_2$ .

In this case, the primitive for  $\lambda_1(t)$  always exists, therefore after integration we obtain:

$$\mu_1(t) = \frac{F_0}{2} \left[ \left( \frac{A_2 + A_1}{A_2 - A_1} e^{A_2 t} + \frac{A_2 - A_1}{A_2 + A_1} e^{-A_2 t} \right) e^{-A_1 t} \right] - F_0 \cdot \frac{A_1^2 + A_2^2}{A_2^2 - A_1^2}. \quad (12)$$



Case 2.  $A_1 = A_2$ .

In this case, before finding a primitive we transform the expression (11), given that  $A_1=A_2$ . Therefore:

$$\lambda_1(t) = F_0 \cdot A_1,$$

and:

$$\mu_1(t) = F_0 \cdot A_1 \cdot \int_0^t dt = F_0 \cdot A_1 \cdot t. \quad (13)$$

Expression (13) having  $A_1=A_2$  correlates well with the expected result. Indeed, with each defect, which is removed from SS, there is one secondary defect that is introduced into it. Therefore, the total number of defects in the SS remains unchanged and the frequency of making defects, because of this, too, remains unchanged. Thus, when  $A_1=A_2$  linear relationship of the cumulative number of defects removed through time is expected, and is derived from the SSD.

For the cumulative trend of input flow (secondary defects), we obtain:

$$\mu_2(t) = \frac{F_0}{2} \left( \frac{A_2 + A_1}{A_2 - A_1} e^{A_2 t} - \frac{A_2 - A_1}{A_2 + A_1} e^{-A_2 t} \right) e^{-A_1 t} - 2F_0 \cdot \frac{A_1 \cdot A_2}{A_2^2 - A_1^2} \quad (14)$$

having  $A_1 \neq A_2$  and

$$\mu_2(t) = F_0 \cdot A_1 \cdot t \quad (15)$$

having  $A_1=A_2$ .

Comparing (13) and (15) can be seen that for  $A_1=A_2$  cumulative trends in output and input flows are the same. This result also corresponds to the physical representations. If  $A_1=A_2$  then the number of defects, which are introduced into the system equals the number of removed ones. From this it follows that their cumulative trends, too, must be the same.

Thus, the mathematical model of software reliability are the expressions (12), (13) and (14), (15) for the outgoing and input flow, respectively. For practical application of reliability models it is necessary to develop a methodology for calculating the parameters of the model based on experimental data.

In the experimental determination of parameters of the model cumulative trend of defects identified at a certain time interval acts as the experimental data. Subject to determination of model parameters are the influence coefficients  $A_1$  and  $A_2$ , as well as the initial number of defects in the system  $F_0$ .

Determination of the parameters will be carried out in two stages. The first step is a preliminary assessment of the parameters, while the second - is their clarification.

For a preliminary assessment, we assume that the input stream of defects is absent, so the coefficient  $A_2=0$ . In this case, the cumulative relationship of these defects will be exponential

$$\mu_1(t) = F_0 - F_0 \cdot e^{-A_1 t}. \quad (16)$$

In of this relation the inverse of the coefficient  $A_1$ , is called time constant of the process

$$\tau = \frac{1}{|A_1|},$$

and is the length of sub-tangent exponentially. In turn, the value of sub-tangent can be defined as shown in Figure 5.

From Figure 5 it follows that magnitude of sub-tangent  $\tau$  can be defined as:

$$\tau = \frac{F_0 - f_1}{tg\alpha},$$

where the slope of the  $tg\alpha$  can be determined from the formula

$$tg\alpha \approx \frac{f_1 - f_2}{t_1 - t_2}.$$

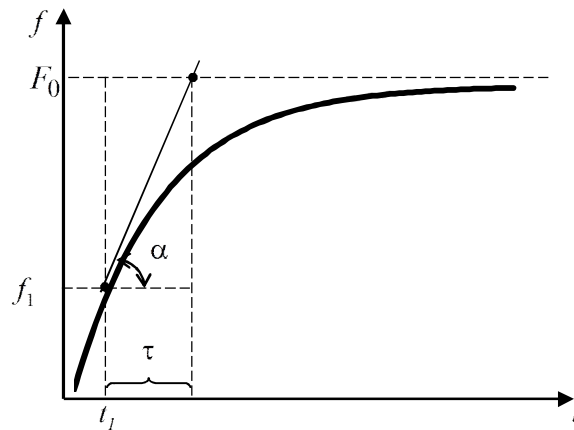


Figure 5. For the  $A_1$  definition

To improve the accuracy of the calculations one should determine the slope of the  $tg\alpha$  for every two consecutive points of the experimental cumulative curve. As an unknown quantity  $F_0$  it is possible to use the last point of the cumulative curve  $f_n$ . Values of  $\tau$  defined in this manner for each successive pair of points must be averaged. From the average of  $\tau$  we find an approximation for the coefficient of influence  $A_1$ :

$$A_1 = \frac{1}{\tau}. \quad (17)$$

Approximation for the  $F_0$  can be derived from the expression:

$$F_0 = \frac{f_i}{1 - e^{-A_1 t_i}}. \quad (18)$$

To improve the accuracy values found in this way  $F_0$  are averaged over all points of the experimental cumulative curve.

Please note that setting the coefficient  $A_2=0$ , that is, excluding the impact of input flow, we have inflated estimates for the  $F_0$  and  $A_1$ .

After defining the approximations for  $F_0$  and  $A_1$  we must iteratively validate them. It should be kept in mind that there is a dependence of  $\mu_1(t)$  on the coefficients  $A_1$  and  $A_2$ , which is shown in Figure 6.

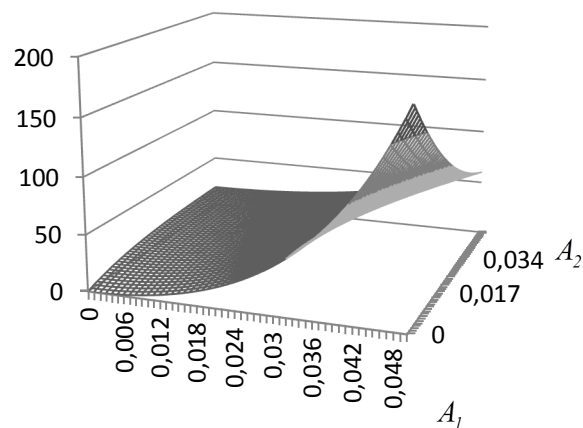


Figure 6. Relation  $\mu_1(t)$  on the influence coefficients

From Figure 6 it follows that with the change of the coefficient  $A_2$  values of  $\mu_1(t)$  change only slightly. Therefore, clarification the model parameters should start with the coefficient  $A_2$ : defects admitted here have negligible impact on the accuracy of the process. To assess the accuracy of determining the parameters you should use the criterion of standard deviation (SD), calculated as:

$$SD = \frac{\sum_{i=1}^n (f_{io} - f_{ic})^2}{n}, \quad (19)$$

where  $n$  – number of points in the experimental cumulative curve;  $f_{io}$  – observed value of  $i$ -th point of the curve;  $f_{ic}$  – value calculated with given parameters.

To clarify the parameters of the model the following algorithm is proposed:

**Step 1.** Definite initial value of  $CKO_b$  for approximations of  $A_1$  and  $F_{10}$  obtained through (12) and (13). Assume  $A_2 = 0$ .

**Step 2.** Changing  $A_2$  by step

$$\Delta A_2 = \frac{1.5 \cdot A_1}{10},$$

while current value of  $CKO_x$  is less, than  $CKO_b$  obtain clarified value for  $A_2$ . Assume  $CKO_b = CKO_x$ .

**Step 3.** Changing  $F_{10}$  in range from 0,5 to 1,5 from approximation obtained in step 1, while current value of the  $CKO_x$  is less, than  $CKO_b$ , find clarified value for  $F_{10}$ . Assume  $CKO_b = CKO_x$ .

**Step 4.** Changing  $A_1$  in range from 0,5 to 1,5 from approximation obtained in step 1, while current value of the  $CKO_x$  is less, than  $CKO_b$ , find clarified value of  $A_1$ . Assume  $CKO_b = CKO_x$ .

**Step 5.** Repeat step 2 – step 4 for the next stage of refinement. Refinement process is considered complete if at the next stage the value  $CKO_b \leq \varepsilon$  is achieved, where  $\varepsilon$  – required calculation precision.

## 6 EVALUATION OF THE ACCURACY OF SIMULATION IN THE SSD

To assess the accuracy of modeling the reliability of the model (SSD model) and its comparison with existing models we used data on the identified defects in twenty different software systems (Android, Lyu 1996) To increase the accuracy of modeling, each series of observations was divided into intervals during which the time variation of the cumulative curves of these defects remained unchanged. 123 total numbers of observations processed. To compare the accuracy of modeling, in addition to the described SRMs there have been taken well-known reliability models,

covering all existing classes of models. They are based on different concepts, which make it possible to objectively evaluate and compare the accuracy of models between each other. The following models were taken into the research: Jelinsky-Moranda's (Moranda & Jelinski 1972), nonhomogeneous Poisson process (Goel & Okumoto 1979), Schneidewind's (Schneidewind 1993), Musa's (Musa 1979), Weibull's (Quadri & Ahmad 2010), S-shaped model (Yamada et al. 1983), Duan's model (Duan 1964), Moranda's geometric model (Moranda 1979) and logarithmic model of Musa-Okumoto (Musa & Okumoto 1984). 1230 total estimations of the reliability modeling were made. Modeling results are shown at Figure 7.

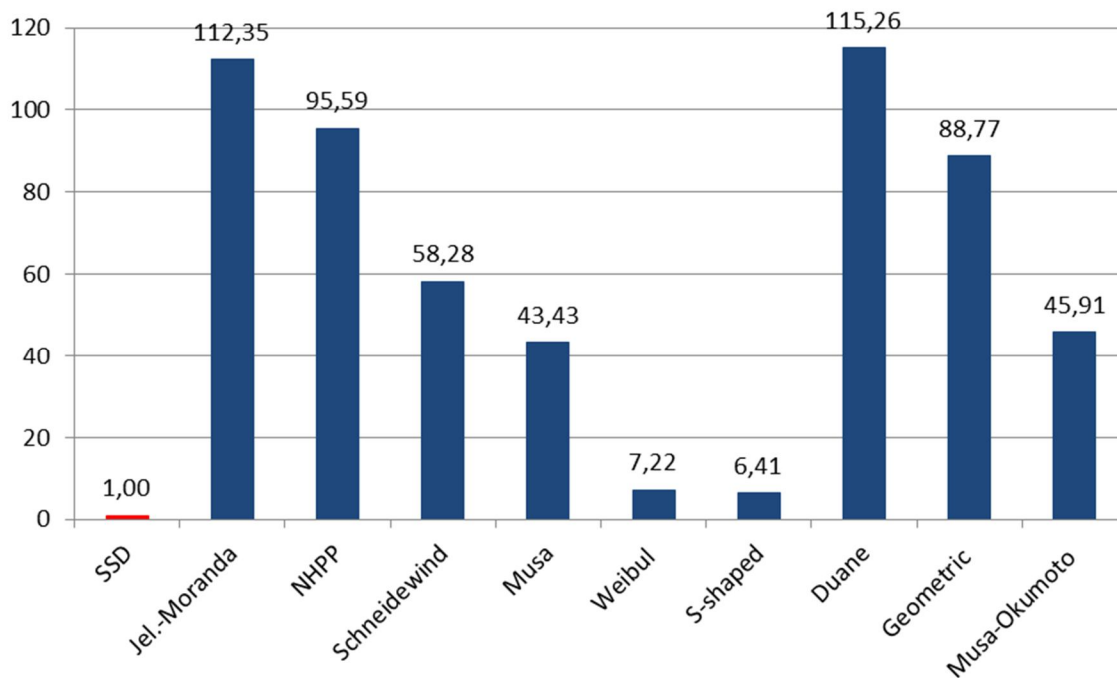


Figure 7. Reliability models' comparison results

In the diagram shown in Figure 7 there are SD values shown which were obtained from the comparison. For convenience, standard deviation of the proposed model of SSD is assumed to be equal to one. The diagram shows that for all classes of the test software model of the SSD showed a result with accuracy of more than six times superior to the result of the best known reliability models – S-shaped model.

## 7 CONCLUSION

The comparative analysis of different SRMs showed that the model based on the SSD proposed in this article consistently shows the best results in comparison with other known models. The stability of the simulation results is essential to avoiding phase selection model of reliability for each software product.

Therefore, the results are a practical confirmation of the SSD, and reliability model developed based on them can be used for modeling and predicting a wide class of reliability indices of software systems.

The new theory of software reliability proposed in the article opens a wide scope for research. In particular, it is interesting to find out the physical nature of the defect capacity index and suggest ways to define it, and to explore other effects, whose existence is a consequence of the new theory.

## 8 REFERENCES

- Android – An Open Handset Alliance Project. <http://code.google.com/p/android/issues/list>
- Duan J.T. (1964). Lurning Curve Approach to Reliability Monitoring. *IEEE Trans. on Aerospace*. Vol. 2: 563-566.
- Goel, A.L., K. Okumoto. (1979). Time-Dependent Error-Detection Rate Model for Software and Other Performance Measures. *IEEE Transactions on Reliability*, v. R-28, № 5: 206-211
- Kharchenko V.S., Sklar V.V., Tarasyuk O.M. (2004). *Methods for modelling and evaluating the quality and reliability of the programming security*. Kharkov: Nat. Aerospace Univ. "HAI"
- Lyu, M.R. (1996). *Handbook of Software Reliability Engineering*. London: McGraw-Hill.
- Moranda, P.B. (1979). *Event-Altered Rate Models for General Reliability Analysis* IEEE Transactions on Reliability. Vol.R-28. No. 5: 376-381
- Moranda, PL., Jelinski, Z. (1972). *Final Report on Software Reliability Study*, McDonnell Douglas Astronautics Company, MADC Report Number 63921.
- Musa J.D. (1979). *Validity of Execution time theory of software reliability*. IEEE Trans. on reliability. No 3: 199-205.
- Musa J.D. Okumoto K. A. (1984). *Logarithmic Poisson Time Model for Software Reliability Measurement*. Proc. Sevent International Conference on Software Engineering. – Orlando, Florida: 230-238.
- Onsager, L. (1931) *Reciprocal relations in irreversible processes*, «Physical Review», 1931, v. 38, № 12: 405-4026
- Prigogine, I. (1991) *The Behavior of Matter Under Nonequilibrium Conditions: Fundamental Aspects and Applications*. Progress Report for Period April 15,1990 – April 14, 1991. Center for Studies in Statistical Mechanics and Complex Systems at the University of Texas-Austin, United States Department of Energy-Office of Energy Research.
- Schneidewind, N.F. (1993). *Software Reliability Model with Optimal Selection of Failure Data*. IEEE Transactions on Software Engineering, Vol. 19, No. 11: 1095-1104.
- Quadri, S. M. K. Ahmad N. (2010). *Software Reliability Growth Modeling with New Modified Weibull Testing-effort and Optimal Release Policy*. International Journal of Computer Applications Vol. 6. № 12: 1-10
- Ushakov, I. (2006). *Reliability: past, present, future*. Reliability: Theory & Applications, No.1: 10 – 16.
- Ushakov, I. (2012). *Reliability theory: history & current state in bibliographies*. RT&A # 01 (24) Vol.1: 8 – 35
- Yamada, S. Ohba M., Osaki S. (1983). *S-Shaped Reliability Growth Modeling for Software Error Detection*. IEEE Transactions on Reliability, vol. R-32, no. 5: 475-478.