REVISTA SAN GREGORIO

http://dx.doi.org/10.36097/rsan.v1i41.1474
Artículo Original

Optimization of the level of components' functionality of developing software systems

Optimización del nivel de funcionalidad de los componentes del desarrollo de sistemas de software.

Authors
Yakov Evseyevich Lvovich[1], Nikita Aleksandrovich Ryndin[2], Alexander Alekseevich Ryndin[3*], Yurii Serafimovich Sakharov[4]

[1]Department of CAD and Information Systems, SBGEI HE Voronezh State Technical University, yakov.lvovich@bk.ru
https://orcid.org/0000-0002-7051-3763

[2]Department of CAD and Information Systems, SBGEI HE Voronezh State Technical University, nikita_ryndin@bk.ru
https://orcid.org/0000-0002-0774-2352

[3*]Department of CAD and Information Systems, SBGEI HE Voronezh State Technical University, alexander.ryndin2017@yandex.ru
https://orcid.org/0000-0002-3470-3228

[4]SBGEI HO Moscow Region "University "Dubna" (State University "Dubna"), yurii.sakharov@inbox.ru
https://orcid.org/0000-0001-5148-1287

Fecha de recibido: 2020-10-14
Fecha de aceptado para publicación: 2020-11-18
Fecha de publicación: 2020-11-20

Abstract
The article discusses the design issues of complex multi-component software systems that function throughout the entire life cycle, taking into account the optimal level of functionality of their components. The authors provide a review of existing design techniques and life cycle models of such systems. The authors conclude that it is necessary to optimize the level of functionality of components with resource constraints for development and maintaining their compliance with functional requirements throughout the entire life of software systems. The concept of the function of matching the functionality of a component with current system requirements is introduced. Moreover, the modification of a system component at certain points in time requires corresponding costs for the modernization and completion of the system component. This is especially true for the class of developing software systems (DSS), which are constantly being upgraded over time. The process of changing the DSS is random. Therefore, probabilistic estimates of the effectiveness of measures and, accordingly, costs are introduced, which allow for maintaining the level of the component's functionality. The random process is approximated by a function with random parameters. As such a function, given the nature of the change in functionality at the stage of the life cycle, an exponential function is selected. The calculation of the effectiveness assessment allows ensuring the distribution of the integral resource between the periods of the system modernization in such a way as to maximize the efficiency of its use. An optimization problem is formulated and an algorithm for its solution is proposed based on a multi-step process for making optimal decisions. As a result, the authors obtained the values of the necessary resources for the modernization of the system at certain intervals of time, ensuring the maximum efficiency of their use to maintain the required functionality of the system.

Keywords: developing software systems, life cycle, resource optimization for system modernization, stochastic models.

Resumen
El artículo analiza los problemas de diseño de los complejos sistemas de software multicomponente que funcionan durante todo el ciclo de vida, teniendo en cuenta el nivel óptimo de funcionalidad de sus componentes. Los autores proporcionan una revisión de las técnicas de diseño existentes y los modelos de ciclo de vida de dichos sistemas. Los autores concluyen que es necesario optimizar el nivel de funcionalidad de los componentes con limitaciones de recursos para el desarrollo y mantener el cumplimiento de los requisitos funcionales durante toda la vida útil de los sistemas de software. Se introduce el concepto de la función de hacer coincidir la funcionalidad de un componente con los requisitos actuales del sistema. Además, la modificación de un componente del sistema en determinados momentos requiere los correspondientes costes de modernización y finalización del componente del sistema. Esto es especialmente cierto para la clase de sistemas de software en desarrollo (DSS), que se actualizan constantemente con el tiempo. El proceso de cambiar el DSS es aleatorio. Por lo tanto, se introducen estimaciones probabilísticas de la efectividad de las medidas y, en consecuencia, los costos, que permiten mantener el nivel de funcionalidad del componente. El proceso aleatorio se aproxima mediante una función con parámetros aleatorios. Como tal función, dada la naturaleza del cambio en la funcionalidad en la etapa del ciclo de vida, se selecciona una función exponencial. El cálculo de la evaluación de la efectividad permite asegurar la distribución del recurso integral entre los períodos de modernización del sistema de manera que se maximice la eficiencia de su uso. Se formula un problema de optimización y se propone un algoritmo para su solución basado en un proceso de múltiples pasos para la toma de decisiones óptimas. Como resultado, los autores obtuvieron los valores de los recursos necesarios para la modernización del sistema en determinados intervalos de tiempo, asegurando la máxima eficiencia de su uso para mantener la funcionalidad requerida del sistema.

Palabras clave: desarrollo de sistemas de software, ciclo de vida, optimización de recursos para la modernización de sistemas, modelos estocásticos.

Introduction

When developing modern software systems (SS), a rather large number of software projects can be distinguished, at most stages of which there is considerable uncertainty in the sphere of estimated information needs. In this case, the SS development project is characterized by a long duration and a serious degree of variability of requirements. Given that the successful implementation of a project of this kind requires that a part of the resources be used for internal reorganization of tasks, subject area, architecture, and development tools, it is possible to classify the software systems developed within them as a subclass of *developing systems* (DS) (Glushkov et al., 1983). This subclass includes dynamic systems with a discrete phase space and an initial nonzero amount of development resources. To study this class of systems, in the general case, algorithmic methods based on the theory of automata and the use of the cybernetic approach are used.

The integrated description of developing systems is convenient enough for their close study, but its main drawback is the practical impossibility of using real software systems to describe it because of their complexity. In the case of applying the SS model to finding system solutions in the field of software design for computing systems, the objective complexity of modern software, as well as the trajectory of its development, makes it difficult to construct rigorous mathematical models and solve problems within their framework.
In the case of modeling the simplest SS, there are two functions:

1. Internal, ensuring its existence.
2. External, which is the result of interaction with the external environment.

The internal function depends on changes in the SS operating conditions, changes in the composition of hardware, new requirements from the emerging versions of the operating system, new device drivers working with the system.

The external function is primarily determined by the emergence of new functional requirements for the system throughout the entire period of software operation and depends on the inherent redundancy or minimization of these requirements during the initial SS development.

In this regard, the choice of optimal compliance of functional requirements with the entire period of SS operation is an urgent task and allows saving material and time resources throughout the life cycle of the system.
The concept of the software life cycle is one of the main concepts in software engineering and is de-

fined as the period that begins from the moment a decision is made about the need to create software and ends when it is completely decommissioned (IEEE Std 610.12, 1990). However, there are many software life cycle models developed by various authors at different times. Typically, a software LC model includes stages, the results of work at each stage, and key events are the points of completion and decision-making. The most famous life-cycle models are cascading and iterative models.

In 1970, W. Royce proposed the so-called cascade model or “waterfall model”, which took into account the severe time limits imposed on software developers by state contracts. A characteristic feature of the model is a strict sequence of stages of development, operation, and maintenance of software, fixing the requirements for the system before its delivery to the customer, the transition to the next stage of the project only after the completion of work on the previous one. This approach has many disadvantages, associated primarily with the fact that the real process of creating software never fits into such a rigid scheme. The real process is usually *iterative* in nature. The results of the next stage often cause changes in design decisions developed in the previous stages. As a result, the so-called iterative model was born, the development of which is the “spiral model” proposed by Barry Boehm in the mid-1980s. The principal features of the model were the rejection of fixing requirements and the assignment of priorities to user requirements, the identification and analysis of risk at each iteration, the separate design of system components and, as a result, the parallelization of system development, the use of prototypes to integrate the developed system components into a single product.

In this regard, the success of the project most often depends on the optimal allocation of resources for the creation and operation of the system throughout the entire life cycle of its existence, ensuring the maximum efficiency of its use to maintain the required functionality of the system.

Methodology

A feature of ensuring the optimal nature of the development of software systems is the need to take into account the life cycle of their components (Vendrov, 2006; Reussner et al., 2019). In the case of using as an evaluation function the development of a component over a certain period $t = \overline{0,T}$ of the adherence function of the component's ty $F(t)$ with current requirements $S(t)$:

$$F(t) \in [0,1] \quad (1)$$

The change graph (1) has a characteristic form (Fig. 1):

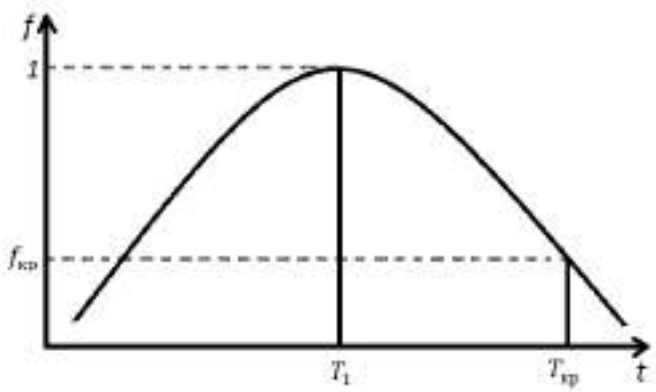

Fig. 1. The change graph in the adherence function

During the life cycle of a component of a developing software system (DSS):
T_1 – Time to reach the value $f(T_1) = 1$;
$T_{кр}$ – Time to reach the critical threshold $f_{кр}(T_{кр})$.
To maintain the level of functionality of a component at a temporary stage $t = \overline{T_1, T_{кр}}$ with the adherence function

$$f(t) \geq f_{кр} \quad (2)$$

At time points $t_k > T_1, k = \overline{1,K}$, measures are taken that require certain costs $z_k = z(t_k)$ so that resource constraints on the development of software system Z are met:

$$\sum_{k=1}^{K} z_k \leq Z \quad (3)$$

The change process (1) at the stage $t = \overline{T_1, T_{кр}}$, based on design practice $P \cap C[2]$, is random:

$$\tilde{f}_k = \tilde{f}(t_k), k = \overline{1,K} \quad (4)$$

Where $\tilde{f}$ is the designation of a random value.
Then, to assess the effectiveness of measures and, accordingly, the costs z_k that allow maintaining the level of component functionality, it is proposed to use a probabilistic assessment of the fulfillment of conditions (2) (Reussner et al., 2019):

$$E(z_k, t_k) = P\left(1 \geq f(z_k, t_k) \geq f_{кр}\right), k = \overline{1,K} \quad (5)$$

Where P is the probability of a random event.
To calculate (5), it is necessary to approximate the random process by some function with random parameters (Reussner et al., 2019). As such a function, taking into account the nature of the change $f(t)$ at the stage of the life cycle $t = \overline{T_1, T_{кр}}$, the authors choose the exponential function

$$\tilde{f}_k = 1 - \tilde{Q}_1 e^{\tilde{Q}_2 k}, k = \overline{1,K}, t_1 > T_1,$$

(6)

Where $\tilde{Q}_1, \tilde{Q}_2$ are random parameters' values.
In practice, random changes (4) of the function $f(t)$ are such that, when approximating (6):
Firstly, the relative standard deviation for $\tilde{Q}_1$ is much smaller than the corresponding value for Q_2:

$$\frac{D^{\frac{1}{2}}(\tilde{Q}_1)}{m(\tilde{Q}_1)} \ll \frac{D^{\frac{1}{2}}(\tilde{Q}_2)}{m(\tilde{Q}_2)},$$

(7)

Where m, D is the designation of the values of mathematical expectation and variance of a random variable,
Therefore, it should be considered (IEEE Std 610.12, 1990) that the parameter $Q_1 = m(\tilde{Q}_1)$;
Secondly, the random variable $\tilde{Q}_2$ obeys the normal distribution law with density

$$\omega\left(\tilde{Q}_2\right) = \frac{1}{\sqrt{2\pi} D^{\frac{1}{2}}(\tilde{Q}_2)} exp\left(-\frac{(\tilde{Q}_2 - m(\tilde{Q}_2))^2}{2D(\tilde{Q}_2)}\right).$$

(8)

Given (6) - (8), random variables $\tilde{f}_k, k = \overline{1,K}$ also obey the normal distribution law (IEEE Std 610.12, 1990) with numerical characteristics calculated for fixed values z_k, $m\left(\tilde{f}_k, z_k\right), D\left(\tilde{f}_k, z_k\right)$:

$$\omega\left(\tilde{f}_k, z_k\right) = \frac{1}{\sqrt{2\pi} D^{\frac{1}{2}}\left(\tilde{f}_k, z_k\right)} exp\left(-\frac{(\tilde{f}_k - m\left(\tilde{f}_k, z_k\right))^2}{2D\left(\tilde{f}_k, z_k\right)}\right), k = \overline{1,K}.$$

(9)

To assess the fulfillment of condition (5), it is necessary to determine $m\left(\tilde{f}_k, z_k\right), D\left(\tilde{f}_k, z_k\right) k = \overline{1,K}$.
Let us use the expansion of function (6) in a Taylor series in a neighborhood of each point $t_k, k = \overline{1,K}$.
When determining the mathematical expectation using its properties, let us restrict ourselves to the second-order expansion term and the variance to the first-order expansion (Nathan et al., 2003). Then the following is obtained:

$$m\left(\tilde{f}_k, z_k\right) = m\left(\tilde{Q}_1\right) e^{m\left(\tilde{Q}_2\right)k}\left[1 + m\left(\tilde{Q}_2\right)k + \frac{1}{2}D(\tilde{Q}_2)k^2\right],$$

(10)

$$D\left(\tilde{f}_k\right) = [m\left(\tilde{Q}_1\right) k e^{m\left(\tilde{Q}_2\right)k}]^2 D\left(\tilde{Q}_2\right).$$

(11)

Characteristics (9) - (11) make it possible to determine the cost-effectiveness assessment of $z_k, k = \overline{1,K}$ to maintain the level of $P \cap C$ component functionality using the tabulated normalized Laplace function (Nathan et al., 2003):

$$E(z_k, t_k) = \Phi\left(\frac{1 - m\left(\tilde{f}_k, z_k\right)}{D^{\frac{1}{2}}\left(\tilde{f}_k, z_k\right)}\right) - \Phi\left(\frac{f_{кр} - m\left(\tilde{f}_k, z_k\right)}{D^{\frac{1}{2}}\left(\tilde{f}_k, z_k\right)}\right),$$

(12)

Where Φ is the notation of the normalized Laplace function.
The ability to calculate the efficiency estimate (12) allows for the distribution of the integral resource Z between periods $t_k, k = \overline{1,K}$ in such a way as to maximize the efficiency of their use to maintain functionality when condition (3) is fulfilled. A formalized record of such an issue looks as follows:

$$\sum_{k=1}^{k} E(z_k, t_k) \to \max_{z_k, k=\overline{1,K} \in G};$$

$$G = \left\{ \begin{matrix} z_k \left| \sum_{k=1}^{K} z_k \le Z \right. \\ z_k \ge 0, k = \overline{1,K} \end{matrix} \right\}$$

(13)

An effective way to solve the optimization problem (13) is to organize a K-step decision-making process (a multi-step process of making optimal decisions - MSPMOD) (Smirnov & Dunin-Barkovsky, 1965), in which the vector of optimized variables:

$$z = (z_1, z_2, \dots, z_k, z_{k+1}, \dots, z_K)$$

(14)

It is divided into k interconnected vectors q_k of smaller dimension:

$$q_k = (z_{k1}, \dots, z_{km}) \in G, k = \overline{1,K}$$

(15)

Vectors (15) are combined into a vector of controlled parameters:

$$q = (q_1, q_2, \dots, q_k, q_{k+1}, \dots, q_K)$$

(16)

The vector's components (16) meet the conditions:

$$\cup_{k=1}^{K} q_k = Z;$$

$$q_k \cap q_l = \emptyset, k, l = \overline{1,K}, k \ne l.$$

(17)

As in (Smirnov & Dunin-Barkovsky, 1965), MSPMOD, let us consider in the reverse correction of the Kth step with index "1" to the first step with index "K" and introduce the state vector before and after selection of q_k, respectively:

$$p_{k+1} = (p_{k+1,1}; p_{k+1,2}; \dots; p_{k+1,m}),$$

(18)

$$p_k = (p_{k1}; p_{k2}; \dots; p_{km}).$$

(19)

In this case, let us set the condition for converting vector (18) to vector (19) after selecting the vector of controlled parameters q_k:

$$p_k = \varphi_k(p_{k+1}, q_k),$$

(20)

Where the vector q_k satisfies the constraint system $G_{qk} \in G$:

$$q_k \in G_{qk}, k = \overline{1,K} \quad ,$$

(21)

And the performance indicator (5) depends on state vectors and controlled parameters:

$$E_k = \psi_k(p_{k+1}, q_k), k = \overline{1,K}$$

(22)

Taking into account (20), (21), based on dependence (22), a comparison is made between valid vectors and controlled parameter sq_k at a fixed state p_{k+1}.

Generalizing the transformation (22) to the whole k-step process of making optimal decisions, and taking into account the additivity of the criterion of the optimization problem (13), we obtain an efficiency estimate for the entire period of $T_1, T_{кр}$:

$$\psi(q_1, \dots, q_k, \dots, q_K, p_{k+1}) = \sum_{k=1}^{K} \psi_k(p_{k+1}, q_k) \quad .$$

(23)

Based on (23), let us write the optimization problem (13) in the following form:

$$\sum_{k=1}^{K} \psi_k(p_{k+1}, q_k) \to \max_{(q_1,\dots,q_k,\dots,q_K) \in G_{qk}} .$$

(24)

Following the conclusions of (Smirnov & Dunin-Barkovsky, 1965), let us proceed from the solution (24) to the sequence of solving one-dimensional optimization problems:

$$\psi_1(p_2) = \max_{0 \le \hat{z}_1 \le p_2} \psi_1\left(\hat{z}_1\right);$$

$$\psi_2(p_3) = \max_{0 \le \hat{z}_2 \le p_3} \{\psi_2\left(\hat{z}_2\right) + \psi_2(p_3 - \hat{z}_2)\};$$

$$\psi_k(p_{k+1}) = \max_{0 \le z_k \le p_{k+1}} \left\{\psi_k\left(\hat{z}_k\right) + \psi_{k-1}\left(p_{k-1} - \hat{z}_k\right)\right\}, \quad (25)$$

Where $\hat{z}_k = \frac{z_k}{z}$ is the normalized cost values.

The system of optimization problems allows implementing the computational scheme of the MSPMOD algorithm.

However, at each step of this scheme, in some cases, there is ambiguity in choosing the optimal value z_k^*. Using the expert assessment procedure (Lvovich et al., 2016), the optimal solution to the problem (13) is obtained:

$$z_1^*, \dots, z_k^*, \dots, z_K^* \quad .$$

(26)

Solution (26) determines the optimal distribution of the resource z between periods $t_k, k = \overline{1,K}$ to maintain the level of component functionality at the stage of the life cycle $T_1, T_{кр}$.

Results

The presented optimization models and algorithms for solving the problem of the optimal allocation of resources for the creation and operation of DSS throughout the entire life cycle of its existence allow, within the framework of various LC models, to get an additional opportunity to complete a project to create a complex multi-component system. Given the ever-changing requirements for the DSS functionality, this approach allows us to ensure the economic efficiency of development, planning, and balancing resources at various stages of development, implementation, and operation of the system.

The introduction of the function of correspondence of the DSS functionality component to user requirements over time allows us to formalize the task of evaluating the effectiveness of measures and, accordingly, the costs z_k that allow maintaining the level of functionality of the component. At the same time, the proposed probabilistic assessment of the fulfillment of the conditions for the compliance of the component functionality with the current requirements is calculated based on the assumption of the randomness of time intervals at which time it becomes necessary to upgrade the system and related resources.

The proposed method for calculating the effectiveness estimate (12) allows for the distribution of the integral resource Z between periods $t_k, k = \overline{1,K}$ in such a way as to maximize the efficiency of their use to maintain functionality when the condition for the function to achieve compliance with the critical value is fulfilled.

Discussions

Analytical studies of recent years show that only 15-20% of projects for the development of software systems are completed on time, do not exceed the planned budget and implement all the required functions and capabilities, the rest are either completed with a delay and excess of the budget or canceled at all (Klotins et al., 2019). In this regard, the development of methods for improving the planning of resources for the development and operation of software systems, especially the DSS category, which allows optimizing the allocation of allocated resources throughout the life cycle of the DSS, subject to all requirements for the functionality of the components and the system as a whole, are relevant and practically necessary for developers, managers and project managers, as well as operating units for such systems.

The issues of formalizing approaches to designing DSS are being discussed for a rather long time. However, apart from development standards, design methodology, and methods for implementing complex projects that generalize the empirical knowledge and experience of developers, there is currently no mathematical apparatus for solving the problem of choosing the architecture, composition of components and development tools that provide the optimal combination of allocated resources and given system functionality.

Attempts to formalize this process are now being undertaken by many researchers and scientific schools (Dell'Anna et al., 2019; Ryndin, 2018). At the same time, significant difficulties arise associated with the uncertainty of requirements, heterogeneity of data, and multicriteria of these tasks. Nevertheless, in any case, to improve the quality of projects, reduce costs, and obtain optimal characteristics of the negotiating system, mathematical modeling and optimization of future design decisions are necessary, as well as accounting for the costs of their modernization throughout the entire life cycle.

Conclusion

Summing up the work done, the authors note the main points of the study:

1. Based on an analysis of the design and development methods of complex software systems and their life cycle, it is concluded that there is a need for a rational distribution of resources for the development, implementation, and operation of the system throughout its entire existence.
2. A technique and a formalized formulation of the problem of the distribution of system resources in time-based on multicriteria optimization models and stochastic algorithms for its solution are proposed.
3. The obtained cost values for each stage of the DSS existence allow for optimizing the total costs while fulfilling the requirements for the functionality of the system components.

References

Batishchev, D.I., Lvovich, Ya.E., Frolov, V.N. (1997). *Optimization in CAD.* Voronezh: publishing house of the Voronezh State University.

Dell'Anna, D., Dalpiaz, F., & Dastani, M. (2019). Requirements-driven evolution of sociotechnical systems via probabilistic reasoning and hill climbing. Automated Software Engineering, 26(3), 513-557.

Glushkov V.M., Ivanov V.V., Yanenko V.M. (1983). *Modeling of developing systems.* Moscow: Nauka, FIZMATLIT, 351 p.

IEEE Std 610.12 – (1990). *IEEE Standard Glossary of Software Engineering Terminology*.

Klotins, E., Unterkalmsteiner, M., & Gorschek, T. (2019). Software engineering in start-up companies: An analysis of 88 experience reports. *Empirical Software Engineering*, *24*(1), 68-102.

Lvovich, I.Ya., Lvovich, Ya.E., Frolov, V.N. (2016). *Information Technologies for Modeling and Optimization: A Brief Theory and Applications*. Voronezh: CPI "Scientific Book".

Lvovich, Ya.E., Lvovich, I.Ya. (2010). *Decision making in an expert-virtual environment*. Voronezh: Scientific Production Center.

Nathan, A.A. Gorbachev, O.G., Goose, S.A. (2003). *Fundamentals of the theory of random processes*. - Moscow: MIPT Publishing House.

Reussner, R., Goedicke, M., W. Hasselbring, W. (2019). *Managed Software Evolution: Springer*, 439 c.

Ryndin, AA. (2018). *Multiple integrations: theory and applications in CAD: monograph*. Voronezh: Voronezh State Technical University.

Smirnov, N.V., & Dunin-Barkovsky, I.V. (1965). *A course in probability theory and mathematical statistics for technical applications*. Moscow: Nauka,.

Vendrov, A.M. (2006). *Software Engineering for Economic Information Systems: A Textbook*. - 2nd ed., Revised. Moscow: Finance and Statistics, - 544s.