T. O. Golubieva; V. M. Dubovoy, Dc. Sc. (Eng.), Prof.

MODEL OF SOFTWARE DEVELOPMENT PROCESSES AS A DYNAMIC MANAGEMENT SYSTEM

The problem of planning software development processes is considered in terms of parameters set uncertainty and in the conditions of a human factor influence. As a result, model of the system for the management of software development production processes has been elaborated.

Keywords: *software development production process, planning, dynamic model.*

Introduction

Software development has changed lately from the art of individual personalities to the production process where millions of specialists from all over the world are engaged. But software development production processes (SDPP) are distinguished by the necessity of planning mental and, to a certain extent, creative activities. Besides, these processes are now fully performed by people and the probability of their automation is still rather low. Nevertheless, SDPP planning and prediction are required for proper management of process resources and this **problem** is of great **current importance**.

SDPP planning and forecasting are standardized [1 – 3]. A number of SDPP systems were implemented [4, 5]. However, modern SDPP planning systems do not take into account parameter uncertainty and the character of separate components of the processes taking place during SDPP, in particular, the influence of resources management during SDPP that turns SDPP into a closed dynamic system. As a result, the task evolves to elaborate the model of a system of software development production processes management as a dynamic system functioning in uncertainty conditions.

Problem solution

Assume that there exists a set of resources $H = \{H_1, H_2, ..., H_n\}$ and a set of tasks $Z = \{Z_1, Z_2, ...Z_m\}$. Resources can be combined into subsets H in accordance with the solutions. In order to fulfill one task, only the resources that can realize a corresponding task can be used.

In the literature that has been analyzed [1 – 5] no references were found as to the tasks being different in their properties. However, in SDPP ever more attention is paid to the fact that processes, taking place in the course of performing tasks, differ from the planned ones due to a number of reasons. As a rule, deviations from the planned course of tasks fulfillment are caused by the existence of risky events. If such risky event occurs, it is necessary to perform the work on its elimination. As a result, to the planned set of tasks a set of works on the elimination of the risky events consequences (EREC) is added $\boldsymbol{R} = \{R_1, R_2, ..., R_k\}$. It should be noted that in this case the number of resources, as a rule, is not increased which leads to the necessity of re-planning the resources already at the stage of performing the work. It should be also noted that EREC works differ from the tasks: the task should necessarily be completed while EREC work is characterized by a certain probability that a corresponding risky event may occur.

For simplification SDPP can be characterized by the following statements:

– The process consists of the planning phase and the phase of performing the work.

– The process has planned dates of beginning the works (DBW) D_{DBW} inishing the works (DFW) D_{DFW}.

– By the planning phase all the problems are considered to be solved because they are not included into the planning process (availability of problems and the necessity of their solutions during the process complicates the model considerably).

– The tasks are not interconnected.

– The amount of resources and the number of tasks are fixed values.

– One task is performed by one executive. (If a task must be performed by several executives, at the planning stage this task can be divided into subtasks, each of them being performed by one executive);

– EREC work is such that cannot stop production process for a long time.

In accordance with the adopted designations, SDPP management system will have the form shown in fig. 1. This diagram is constructed taking into account the above-mentioned statements. In the diagram it is shown that supervisor of the works (WS) distributes the resources between the subsets according to the tasks. Then he estimates the time required for the completion of each task and the number of available man-hours required for the solution of each task. After that the resources are distributed between the tasks and EREC works. Then optimization of the developed plan is carried out using automated systems of enterprise and process management. At this stage the planning phase is completed and WS receives the process time-table. After that the process passes to the process fulfillment phase. At the process fulfillment phase the hours spent on each task and EREC works are entered into the developed time-table. After each of such entries WS receives the updated time-table and evaluates the corresponding process state. If the process is not completed or stopped, a repeated time-table optimization is necessary. Some resources are redistributed between the tasks. After that hours spent on the fulfillment of each task and on EREC works are entered into the developed time-table again. The updated time-table is being optimized until the process is completed due to a certain reason (process completion does not necessarily mean the complete fulfillment of the tasks).

In fig. 1 the adaptive loop of the resources distribution system for software development is shown by a dashed line. In fact this loop is the work fulfillment phase described above.

The distinguishing feature of software development processes is the fact that all development work is performed by people. Arbitrary operation, where human labour is used, is characterized by the time for its completion being approximately set by the works supervisor and, quite often, with a considerable reserve. This leads to the inefficient use of human resources. Evidently, it would be more expedient to determine the time interval required for the completion of the work. Such representation will enable the program to specify the time that would be optimal in terms of modeling. This would make it possible for the manager to review the duration of certain works in order to optimize the resources allocation. Now this work is done by a manager himself on the basis of experience and knowledge of individual executive. As the time for each task fulfillment is determined by the process supervisor, fuzzy logic application to describe the time required for task fulfillment is considered to be expedient [6].

But representation using mathematical apparatus of fuzzy logic is not complete. Such estimates can, in some cases, be refined using the experience of previous processes and that of the completed portion of the work. This may be helpful in case of the tasks repeated in the course of the process. The majority of software testing tasks belong to them. The distinguishing feature of the testing process planning is the necessity of repeated fulfillment of identical tasks. In such case the same employee will spend different time on the fulfillment of these tasks. Therefore, it is expedient to use previous information for subsequent planning. In this case we obtain the model where certain time parameters are set by the manager and others are computed by the system stochastically (fig. 2). Here it is possible to estimate both the approximate time required for the task fulfillment as well as the time necessary for a certain employee to complete the corresponding task.

Fig. 1 shows that in the course process planning and fulfillment data are entered into the knowledge base while already accumulated knowledge is used for planning.

The tasks can be performed in parallel, i. e. if there are no restrictions as to the task completion in a definite sequence, several executives can perform a number of tasks simultaneously with one manager performing one task at a time.

Fig. 1 shows the diagram of the system of software development process management.

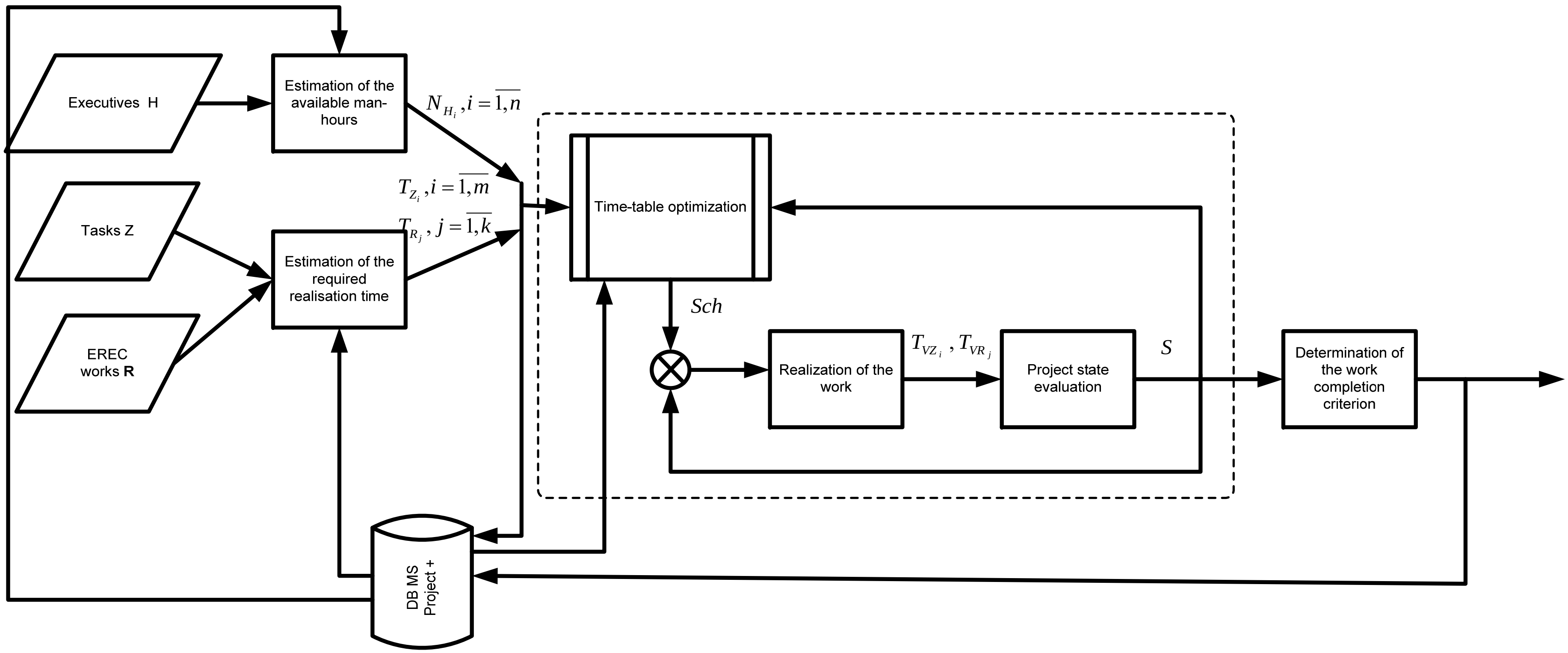

Fig.1. Diagram of the management system for software development production process

Prediction of the process completion time is determined by the dynamics of the closed loop shown in fig. 1 by a dotted line. Now we shall consider this part of the system in more detail. Detailed diagram is presented in fig. 2.

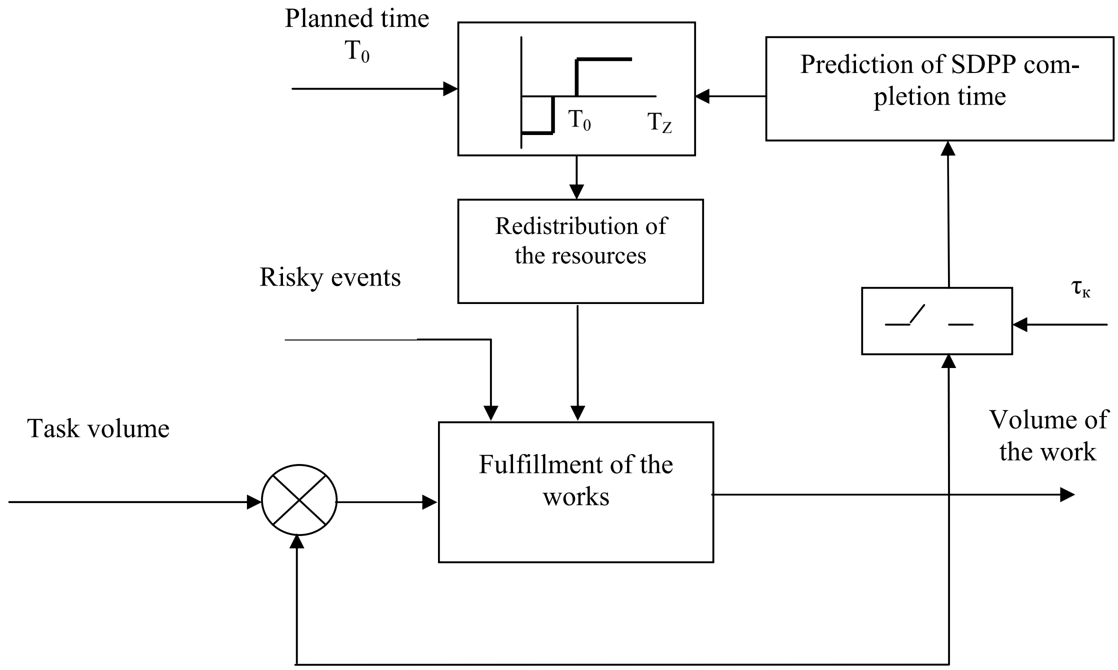

Fig.2. SDPP as a closed dynamic system

Let's consider the models of each system link dynamics.

Model of the "Task fulfillment" link dynamics can be represented graphically by a phase portrait depicted in fig. 3.

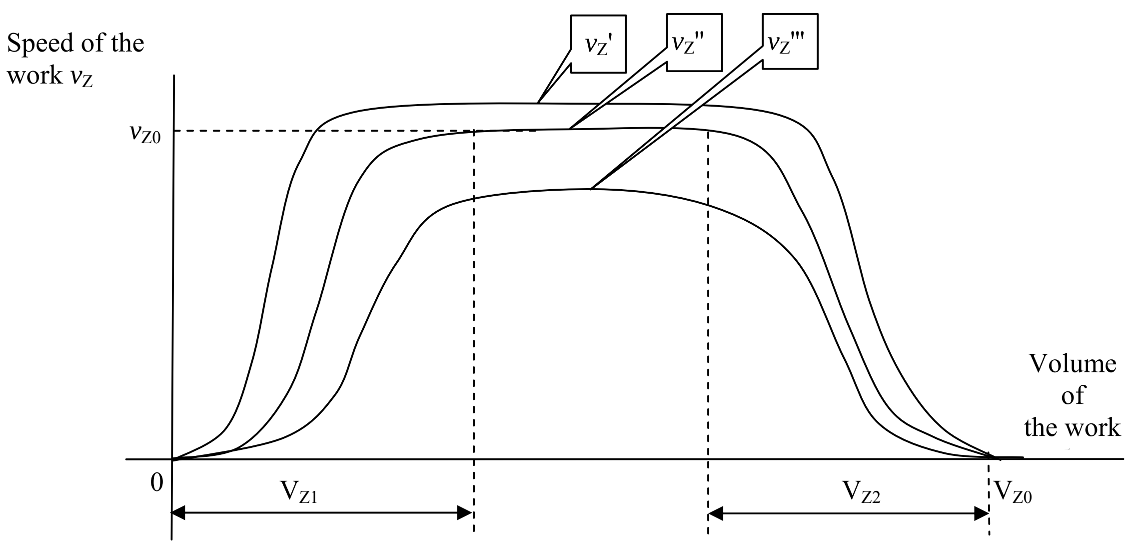

Fig. 3. Phase portrait of the works fulfillment link

Task fulfillment speed can be defined as the function of a set of factors

$$v_{Z_i} = f(V_Z, V_{Z0}, Sk_{Z_i}, Tp_{Z_i}, \{K_{E_j}, Mt_{E_j}, Rs_{E_j}, HV_{E_j}, SO_{E_j}, Hl_{E_j}, PS_{E_j}\}, Cm_{Z_i}), \quad (1)$$

where V_Z is the completed volume of the task; V_{Z0} – general volume of the task; K_E – qualification of the executive; Sk_{Z_i} – complexity of the task, Tp_{Z_i} – type of the task, Mt_{Z_i} – motivation of the executive, Rs_{Z_i} – responsibility of the executive, Cm_{Z_i} – cooperation in the team, HV_{Z_i} – working equipment, SO_{Z_i} – self-organization of the executive, Hl_{Z_i} – general state of the executive, PS_{Z_i} – learning ability. Phase portraits constructed in $v_Z(V_Z)$. The rest of the factors determine the form of the phase trajectory that is characterized by three parameters: the length of the initial V_{Z1} portion and corresponding time of the executive's preparation to the task, maximal work fulfillment speed v_{Z0}, the length of final potion V_{Z2} and corresponding work completion time that, in its turn, depends on the number of errors made while performing the task.

Total work fulfillment time can be estimated on the basis of the reverse phase trajectory integration:

$$T_Z = \int_0^{V_{Z0}} \frac{dV_Z}{v_Z}. \quad (2)$$

Each ERCE work is characterized by a certain probability p_{R_j} of its necessity to be performed. Even if a risky event occurs, EREC work may also have the fulfillment priority $w_{R_j} \in [0;1]$.

As a result, weighted estimates of the time spent on the fulfillment of the task and EREC works will be obtained:

$$T\_zv_{Z_i} = w_{Z_i} T_{Z_i}; \quad (3)$$

$$T\_zv_{R_j} = w_{R_j} p_{R_j} T_{R_j}. \quad (4)$$

The second component of the dynamic model is the model of work fulfillment state control that is characterized by periodicity $\tau_к$ and error ΔV_Z. As the control is performed by an expert supervisor, there will be fuzzy error estimation. Periodicity of the control depends on the priority of the work taking into account the restrictions: maximum once in a day and at least once in a week.

The third element of the dynamic model is the prediction of work completion time T. The prediction is based on the relationship (1) and formula (2). As (1) is a fuzzy relationship, integral (2) is also fuzzy [8].

The fourth component of the model is the element of comparison of the predicted work completion time with the planned time $T_0 = \{T_{0min}, T_{0max}\}$ and decision taking as to the necessity of resources redistribution.

Predictions of the work completion time and making decisions on the necessity of the resources redistribution are formal procedures and the time spent on them could be ignored.

Resources redistribution is the problem of search and optimization with restrictions. It is performed using the basic model (1), (2). Restrictions set on the resources redistribution are concerned with both the resources themselves and redistribution time. These restrictions depend on the priority of the task – the higher the priority is the less stringent restrictions will be. As SDPP resources are a discrete value, resources redistribution is performed on the basis of multi-step strategy. Its example is shown in fig. 4.

Thus, general dynamic model is given by the set of equations

$$v_{Z_i} = f(V_Z, V_{Z0}, Sk_{Z_i}, Tp_{Z_i}, \{K_{E_j}, Mt_{E_j}, Rs_{E_j}, HV_{E_j}, SO_{E_j}, Hl_{E_j}, PS_{E_j}\}, Cm_{Z_i});$$

$$T_Z = \sum_{k=1}^{m} \int_{V_{Z_{k\,start}}}^{V_{Z_{k\,end}}} \frac{dV_Z}{v_{Z_k}};$$

$$m < (T_Z - t)/\tau_k \,; \tag{5}$$

$$u = \begin{vmatrix} +1, & T_Z > T_{0\max}; \\ 0, & T_{0\min} \le T_Z \le T_{0\max}; \\ -1, & T_Z < T_{0\min}; \end{vmatrix}$$

$$\left(m, \left[V_{Z_{k\,start}}, V_{Z_{k\,end}}\right], \{E_k\}\right): \min\left|T_Z - T_0\right|,$$

where *m* is the number of steps in the resources allocation strategy.

An equation set (5) is fuzzy and stiff. Some of its parameters are set statistically. For its solution with the aim of the work completion time prediction, the application of special methods and algorithms is required [8 – 11].

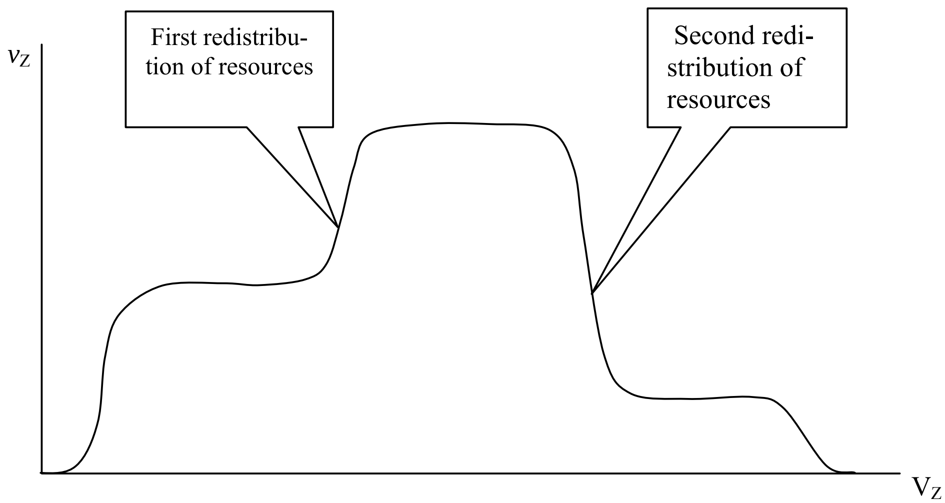

Fig. 4. Example of the multi-step strategy of resources redistribution

Conclusions

In this paper dynamic model of the management system for software development production processes is developed. This system takes into account the uncertainty of a number of parameters of executives and works, supervisor interference and resources redistribution between the tasks as well as risky events that may occur in the course of work. The system also makes it possible to increase accuracy and validity of the works completion time prediction.

REFERENCES

1. ISO – Management standards. – Режим доступу : http://www.iso.org/iso/iso_catalogue/management_standards.htm.

2. Бушуев С. Д. Управление проектами. Основы профессиональных знаний и система оценки компетенции проектных менеджеров (National Competence Baseline, NCB UA Version 3.0) / С. Д. Бушуев, Н. С. Бушуева. – К. : ІРІДІУМ, 2006. – 208 с.

3. Типовые нормы времени на программирование задач для ЭВМ. – М. : НИИ труда, 1980. – 28 с.

4. Спайдер Проджект: Управление Проектами|Project Management|консалтинг|обучение|Spider Project – 2008. – Режим доступу : http://www.spiderproject.ru/.

5. Project Home Page – Microsoft Office Online – 2008. – Режим доступу : http://office.microsoft.com/en–us/project/FX100487771033.aspx.

6. Yousefli A. A. New Heuristic Model for Fully Fuzzy Project Scheduling / A. Yousefli, M. Ghazanfari, K. Shahanaghi, M. A. Heydari. – Режим доступу : http://www.worldacademicunion.com/journal/jus/jusVol02No1paper07.pdf.

7. Орловский С. А. Проблемы принятия решений при нечеткой исходной информации / С. А. Орловский. – М. : Радио и связь, 1981. – 286 с.

8. Раскин Л. Г. Нечеткая математика. Основы теории. Приложения / Л. Г. Раскин, О. В. Серая. – Х. : Парус,

2008. – 352 с.

9. Москвіна С. М. Аналіз алгоритмів розв'язання систем жорстких диференціальних рівнянь у пакетах моделювання систем автоматичного керування / С. М. Москвіна, Т. О. Голубєва // Вісник Вінницького політехнічного інституту. – 2004. – № 2. – С. 21–25: іл. 5. – ISSN 1997–9266.

10. Москвіна С. М. Підхід до підвищення стійкості чисельних алгоритмів при моделюванні систем автоматичного керування / С. М. Москвіна, Т. О. Голубєва // Вісник Вінницького політехнічного інституту. – 2003. – № 6. – С. 92–95: іл. 1. – ISSN 1997–9266.

11. Москвіна С. М. Про підвищення стійкості методів моделювання жорстких систем / С. М. Москвіна, О. М. Москвін, Т. О. Голубєва // Вісник Вінницького політехнічного інституту. – 2007. – № 1 – С. 84 – 88. – ISSN 1997–9266.

Golubieva Tatiana – Post-graduate student of the Computer control systems department. gtat@bigmir.net

Dubovoy Volodimir – Dc. Sc. (Eng.), Prof. of the Computer control system department.

Vinnytsia National technical University.