UDC 004.9: 004.05
DOI: 10.31891/CSIT-2021-4-8

DMYTRO MEDZATYI, TETIANA HOVORUSHCHENKO
Khmelnytskyi National University

METHOD OF SELECTION OF SOFTWARE DESIGN TECHNOLOGY

The paper further develops the mathematical model of the software design technology (SDT) and the criteria for evaluating the SDT, which allow experts to evaluate each considered software design technology more accurately, taking into account all its components. The method and production rules of the selection of the software design technology proposed by the authors give the organization the opportunity to make a motivated and reasonable choice of the design technology for its further implementation.
Keywords: software, software design technology, technological process, technological operation.

ДМИТРО МЕДЗАТИЙ, ТЕТЯНА ГОВОРУЩЕНКО
Хмельницький національний університет

МЕТОД ВИБОРУ ТЕХНОЛОГІЇ ПРОЄКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Технологія проєктування (розроблення) програмного забезпечення (ТППЗ) – це впорядкована сукупність взаємозв'язаних технологічних процесів в межах життєвого циклу ПЗ. ТППЗ представляє собою інженерний підхід до розроблення програмних засобів, який охоплює методологію програмування, проблеми забезпечення надійності програм, оцінки робочих характеристик та якості проектів.

Початковими даними для оцінювання технології проєктування програмного забезпечення є набір параметрів (техніко-економічних характеристик) ТППЗ функціональні характеристики, орієнтовані на процеси життєвого циклу ПЗ (керування проєктом, керування вимогами, керування конфігурацією та змінами, аналіз та проектування ПЗ і т.і.); функціональні характеристики застосування (середовище функціонування, сумісність з іншими ТППЗ, відповідність технологічним стандартам); характеристики якості (надійність, зручність використання, ефективність, супроводжуваність, можливість переносу); загальні характеристики (витрати на технологію, ліцензійна політика, оціночний ефект від впровадження ТППЗ, потрібна для впровадження інфраструктура, доступність та якість навчання, сертифікація постачальника, підтримка постачальника). На основі даного набору параметрів аналізуються та класифікуються існуючі ТППЗ.

Сьогодні процеси оцінювання та вибору ТППЗ залишаються незабезпеченими математичним підґрунтям. Тому актуальною задачею наразі є побудова математичного апарату для підтримки процесів оцінювання та вибору ТППЗ. Враховуючи вищевикладене, метою даного дослідження є побудова критеріїв оцінювання та продукційних правил вибору технології проєктування ПЗ.

У статті набули подальшого розвитку математична модель ТППЗ та критерії оцінювання ТППЗ, які дають можливість експертам оцінити кожну розглядувану технологію проєктування ПЗ більш точно, з врахуванням всіх її складових частин. Запропоновані авторами метод та продукційні правила вибору технології проєктування ПЗ надають організації можливість мотивованого та обгрунтованого вибору технології проєктування для її подальшого впровадження.
Ключові слова: програмне забезпечення, технологія проєктування програмного забезпечення, технологічний процес, технологічна операція.

Introduction

Software design technology includes all specific facts, levels, important knowledge, activities, operations, techniques for managing software development.

Software design (development) technology (SDT) is an ordered set of interconnected technological processes within the software life cycle. SDT is an engineering approach to software development that covers programming methodology, software reliability issues, performance evaluation, and project quality. SDT defines the professional culture of specialists, which provides a given level of productivity and quality of the resulting software products [1-4].

The main requirement for modern SDTs is their compliance with standards and regulations related to software life cycle processes and assessment of technological maturity of development organizations [5-7].

The implementation of the SDT is understood as all actions - from the assessment of initial needs to the full use of the SDT in various departments of the organization [1-4]. The SDT implementation process consists of the following stages [1-4]: 1) identification of the SDT necessity, the characteristics of the object of implementation and software development projects; 2) determination of the requirements for the SDT (analysis of the characteristics of the object of implementation and projects, justification of the requirements for the SDT, determination of the priorities of the requirements); 3) evaluation of SDT variants – preliminary expert evaluation, which consists in the analysis of available SDT for compliance with the requirements, and detailed evaluation, which consists in forming a detailed description of each SDT-applicant; 4) identification of the needs of the SDT, the characteristics of the object of implementation and projects; 5) the choice of SDT on the basis of comparative analysis of technologies and taking into account expert assessment; 6) adaptation of the SDT to the conditions of application by forming a specific working configuration of the SDT, adapted to the conditions of the object of implementation.

The initial data for evaluation are a set of parameters (technical and economic characteristics) of SDT [1-4]: 1) functional characteristics focused on the software life cycle processes (project management, requirements management, configuration and change management, software analysis and design, etc.); 2) functional characteristics of application (operating environment, compatibility with other SDTs, compliance with technological standards); 3) quality characteristics (reliability, usability, efficiency, traceability, portability); 4) general characteristics (technology costs, licensing policy, evaluation effect of SDT implementation, infrastructure required for implementation, availability and quality of training, supplier certification, supplier support). Based on this set of parameters, existing SDTs are analyzed and classified.

Today, the processes of evaluation and selection of SDT have not mathematical basis. Therefore, *the actual task* now is to build a mathematical apparatus to support the processes of evaluation and selection of SDT. Given the above, *the purpose of this study* is to build evaluation criteria and production rules for the selection of software design technology.

Mathematical model of software design technology

Given the fact that the SDT includes the technological process, technological operation, working product, role, leadership, and CASE-tool, we will build a mathematical model of software design technology [8]:

$$TSD = \{TO, TP, WP, R, G, CT, TOR, WPTOR, GTO\}, \quad (1)$$

where: $TO = \{to_1,..., to_f\}$ – the set of technological operations (f – the quantity of SDT technological operations); $TP = \{tp_1,...,tp_k\}$ – the set of technological processes (k – the quantity of SDT technological processes), each of which is a subset $\{to_1,...,to_m\}$ of interconnected technological operations (m – the quantity of technological operations that make up this technological process), i. e. $TP = \{\{to_1,...,to_o\}_1,...,\{to_1,...,to_e\}_k\}$, where o – the quantity of technological operations that make up the technological process tp_1, e – the quantity of technological operations that make up the technological process tp_k; $WP = \{wp_1,...,wp_x\}$ – the set of working products wp_c (x – the quantity of working products of technology); $R = \{r_1,...,r_{lp}\}$ – the set of roles r_y, which SDT contains (lp – the quantity of SDT roles); $G = \{g_1,...,g_f\}$ – the set of practical guides g_q (f – the number of technological operations and, accordingly, technology manuals); $CT = \{ct_1,...,ct_w\}$ – the set of CASE-tools ct_a (w – the quantity of tools of design technology); $TOR = \{<to_1, r_1>,...,<to_n, r_n>\}$ – the set of pairs to_{ji} and r_{ji}, where to_{ji} – technological operation, which is part of the software design technology and is performed by the role r_{ji} (n – the number of interrelated operations and roles of technology); $WPTOR = \{<wp_1, to_1, r_1>,...,<wp_s, to_s, r_s>\}$ – the set of triplets wp_d, to_d, r_d, where wp_d – the working product that is created, modified or used in a technological operation to_d and defines the area of responsibility of the role r_d (s – the number of interconnected work products, process operations, and technology roles); $GTO = \{<g_1, to_1>,...,<g_f, to_f>\}$ – the set of pairs g_u, to_u, where g_u – practical guide to performing a technological operation to_u (f – the quantity of technological operations and, accordingly, technology manuals).

Criteria for evaluating the software design technology

We offer the following criteria for evaluating the SDT [8]:

1) the labor intensity of creating software – the quantity of man-months that are projected to be spent on creating software using SDT, or the number of functional points (this criterion should be minimized);

2) productivity – the amount of work (number of lines of code), which is per unit of labor intensity (man-month) when using this SDT (criterion requires maximization);

3) the number of defects in the generated software when using this SDT (this criterion must be minimized);

4) the level of return on investment – this criterion requires maximization and is calculated by the formula:

$$return\_on\_investment = \frac{(profit - costs)}{costs}, \quad (2)$$

where $profit$ – profit from the use of software, $costs$ – software creation and maintenance costs;

5) software maintenance costs – this criterion requires minimization and is calculated by the formula:

$$maintenance\_costs = \frac{cost\ of\ support}{total\ costs}, \quad (3)$$

where $cost\ of\ support$ – the expected cost of the software maintenance phase when using this SDT, $total\ costs$ – the cost of implementing this SDT in the organization;

6) time of SDT implementation – the projected time interval from the beginning of SDT implementation to the full implementation and use of SDT by all participants in the development process, which requires minimization;

7) the cost of implementing the SDT – the expected total cost of acquisition, study, and maintenance of the SDT (this criterion should be minimized);

8) the payback period for the implementation of the SDT – the time interval from the beginning of the implementation of the SDT to the full payback of the costs of its implementation (the criterion requires minimization).

Therefore, the group of experts should evaluate each of the considered software design technologies, taking into account its components (formula (1)), according to all the above criteria. As a result of such estimation, the following set of parameters for i -th considered SDT will be received:

$$EVAL = \{l, p, nd, ri, mc, it, ic, re\}, \tag{4}$$

where l – the projected labor intensity of software using the i-th SDT (for example, the number of functional points), p – productivity when using the i-th SDT, nd – the predicted number of defects in the generated software when using the i-th SDT, ri – the expected level of return on investment when using the i-th SDT, mc – the cost of software maintenance, which was developed using the i-th SDT; it – expected time of implementation of the SDT (months); ic – expected costs for the implementation of the i-th SDT; re – payback period for the implementation of the i-th SDT (months).

Then, if experts evaluate j software design technologies, we obtain j sets $EVAL$ (equation (4)), i.e. $EVTSD = \{EVAL_1,..., EVAL_j\}$, where $EVTSD$ – the set of subsets of estimates for all considered SDT.

The set of subsets $EVTSD$ is more appropriate to represent in the form of the following matrix (taking into account formula (4)):

$$EVTSD = \begin{vmatrix} l_1 & p_1 & nd_1 & ri_1 & mc_1 & it_1 & ic_1 & re_1 \\ ... & ... & ... & ... & ... & ... & ... & ... \\ l_j & p_j & nd_j & ri_j & mc_j & it_j & ic_j & re_j \end{vmatrix}, \tag{5}$$

where the first line contains estimates for software design technology №1, the second line – estimates for SDT №2, etc., and the last line – estimates for the latest design technology № j.

Method and production rules for selection of software design technology

The selection of SDT includes the following actions [1-4, 8]: 1) formulation of selection tasks, including goals, assumptions, and limitations; 2) implementation of all necessary selection actions, including the definition of criteria, identification of candidate technologies, collection of necessary data and application of criteria to evaluation results to determine the means with the best indicators; 3) performing the required number of iterations in order to select (or reject) technology that has similar indicators.

Based on the formula (5) and the description of the criteria for evaluating the SDT, we will develop *the method of selection of the optimal SDT for software organization*:

Step 1. Find the minimum elements of the 1st (l), 3rd (nd), 5th (mc), 6th (it), 7th (ic) and 8th (re) columns of the matrix $EVTSD$ and save their row numbers in the variables $\min_1$, $\min_3$, $\min_5$, $\min_6$, $\min_7$, $\min_8$ respectively.

Step 2. Find the maximum elements of the 2nd (p) and 4th (ri)columns of the matrix $EVTSD$ and save the numbers of their rows in the variables $\max_2$, $\max_4$ respectively.

Step 3. Using production rules to determine the best software design technology for each criterion, assign 1 point to the technology-"winner" for each criterion, calculate the sum of points for each SDT and save the sum of points of technology № ki in element № ki of the set ST, where $ST = \{st_1,...,st_{nt}\}$ – the set of points for each technology design, and st_1 – the number of points scored by technology №1, st_{nt} – the number of points scored by technology № nt, nt – the number of considered SDT.

Крок 4. Find the maximum element of the set $ST = \{st_1,...,st_{nt}\}$ and save its number in the variable not.

Production rules for the choice of software design technology have the form:

1) if the variable $\min_1 = ti$ (i.e. technology number ti is the best by the criterion of "labor intensity of creating software"), then technology number ti receives 1 point: $st_{ti} = st_{ti} + 1$ (st_{ti} – the number of points of the ti -th technology, where $st_{ti} \in ST$);

2) if the variable $\max_2 = ti$, then technology number ti receives 1 point: $st_{ti} = st_{ti} + 1$;

3) if the variable $\min_3 = ti$, then technology number ti receives 1 point: $st_{ti} = st_{ti} + 1$;

4) if the variable $\max_4 = ti$, then technology number ti receives 1 point: $st_{ti} = st_{ti} + 1$;

5) if the variable $\min_5 = ti$, then technology number ti receives 1 point: $st_{ti} = st_{ti} + 1$;

6) if the variable $\min_6 = ti$, then technology number ti receives 1 point: $st_{ti} = st_{ti} + 1$;

7) if the variable $\min_7 = ti$, then technology number ti receives 1 point: $st_{ti} = st_{ti} + 1$;

8) if the variable $\min_3 = ti$, then technology number ti receives 1 point: $st_{ti} = st_{ti} + 1$;

9) if the variable $not = toi$, then the technology number toi is the most optimal for implementation in this organization.

Example of selection of software design technology

For example, the group of experts considered such software design technologies as:: Rational Unified Process (RUP); Oracle; Borland; Computer Associates (CA).

Thus, a group of experts evaluated each of the considered software technologies, taking into account its components, according to all the above criteria, as well as taking into account the characteristics of the IT company of Khmelnytskyi, for which the selection of SDT is carried out. As a result of this assessment, the following sets of parameters were obtained:

– for Rational Unified Process (RUP): $EVAL_{RUP} = \{572,\ 16588,\ 25,\ 0.25,\ 0.375,\ 15,\ 3000,\ 23\}$,

– for Oracle: $EVAL_{Oracle} = \{785,\ 13456,\ 41,\ 0.12,\ 0.467,\ 18,\ 4200,\ 27\}$,

– for Borland: $EVAL_{Borland} = \{814,\ 12769,\ 47,\ 0.1,\ 0.518,\ 21,\ 5100,\ 31\}$,

– for Computer Associates: $EVAL_{CA} = \{803,\ 13016,\ 45,\ 0.105,\ 0.479,\ 20,\ 5000,\ 29\}$.

Then the matrix $EVTSD$ has the following form:

$$EVTSD = \begin{vmatrix} 572 & 16588 & 25 & 0.25 & 0.375 & 15 & 3000 & 23 \\ 785 & 13456 & 41 & 0.12 & 0.467 & 18 & 4200 & 27 \\ 814 & 12769 & 47 & 0.1 & 0.518 & 21 & 5100 & 31 \\ 803 & 13016 & 45 & 0.105 & 0.479 & 20 & 5000 & 29 \end{vmatrix}.$$

According to step 1 of the proposed method of selection of SDT, we find the minimum elements of the 1st, 3rd, 5th, 6th, 7th and 8th columns of the matrix $EVTSD$ and save their row numbers: $min_1 = 1$, $min_3 = 1$, $min_5 = 1$, $min_6 = 1$, $min_7 = 1$, $min_8 = 1$.

According to step 2 of the proposed method of selection of SDT, we find the maximum elements of the 2nd and 4th columns of the matrix $EVTSD$ and save the numbers of their rows: $max_2 = 1$, $max_4 = 1$.

According to step 3 of the proposed method of selection of SDT, we determine the best SDT for each criterion, we assign 1 point to the technology-"winner" for each criterion, we calculate the sum of points for each SDT and save the sum of points of each technology in the corresponding element of the set $ST = \{8,0,0,0\}$.

Obviously, the maximum element of the set ST is the 1st element, therefore, the technology №1 (Rational Unified Process) is the most optimal for implementation in this organization.

Conclusions

Today, the processes of evaluation and selection of SDT have not mathematical basis. Therefore, the actual task now is to build a mathematical apparatus to support the processes of evaluation and selection of SDT. Given the above, the purpose of this study is to build evaluation criteria and production rules for the selection of SDT.

The paper further develops the mathematical model of the SDT and the criteria for evaluating the SDT, which allow experts to evaluate each considered software design technology more accurately, taking into account all its components. The method and production rules of the selection of the software design technology proposed by the authors give the organization the opportunity to make a motivated and reasonable choice of the design technology for its further implementation.

REFERENCES

1. Savenko I. Software development technology: lecture notes / I. Savenko. – Tomsk: Tomsk Polytechnic University Publishing House, 2014. – 67 p.

2. Bakhtizin V., Glukhova L. Software development technology: a textbook / V. Bakhtizin, L. Glukhova. – Minsk: BGUIR, 2010. –267 p.

3. Patsey N., Shiman D., Sukhorukova I. Software development technology: a textbook on course design / N. Patsey, D. Shiman, I. Sukhorukova. – Minsk: BSTU, 2011. –130 p.

4. Rudakov A., Fedorova G. Technology of software development: workshop / A. Rudakov, G. Fedorova. – Moscow.: Publishing Center "Academy", 2014. – 192 p.

5. ISO/IEC 12207:2017. Systems and software engineering — Software life cycle processes. [Introduced 01.11.2017]. – Geneva (Switzerland), 2017. – 60 p. (International standard).

6. ISO/IEC 9000:2015. Quality management systems — Fundamentals and vocabulary. [Introduced 01.03.2015]. – Geneva (Switzerland), 2015. – 84 p. (International standard).

7. Paulk M., Curtis B., Chrissis M., Weber C. Capability Maturity Model fro Software (Version 1.1): Technical Report – Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University // [Electronic resource] / M. Paulk, B. Curtis, M. Chrissis, C. Weber. – Access mode: http://www.sei.cmu.edu/reports/93tr024.pdf

8. Hovorushchenko T., Kuras E., Krasovskyi M. Evaluation and selection of system software design technology / T. Hovorushchenko, E. Kuras, M. Krasovskyi // Proceedings of III All-Ukrainian scientific-practical conference of young scientists and students "Intelligent technologies in system programming". – Khmelnytskyi: PP Gonta, 2014 – pp. 357-365.

Dmytro Medzatyi **Дмитро Медзатий**	PhD, Associate Professor, Associate Professor of Computer Engineering & System Programming Department, Khmelnytskyi National University, Khmelnytskyi, Ukraine, e-mail: medza@ukr.net. orcid.org/0000-0002-1879-2945, Scopus Author ID: 56008190300 https://scholar.google.com/citations?user=p7GyWqQAAAAJ&hl=uk&oi=ao	кандидат технічних наук, доцент, доцент кафедри комп'ютерної інженерії та системного програмування, Хмельницький національний університет, Хмельницький, Україна.
Tetiana Hovorushchenko **Тетяна Говорущенко**	DrSc, Professor, Head of Computer Engineering & System Programming Department, Khmelnytskyi National University, Khmelnytskyi, Ukraine, e-mail: govorushchenko@gmail.com. orcid.org/0000-0002-7942-1857, Scopus Author ID: 54420153900 https://scholar.google.com.ua/citations?user=aIJyKc8AAAAJ&hl=uk	доктор технічних наук, професор, завідувач кафедри комп'ютерної інженерії та системного програмування, Хмельницький національний університет, Хмельницький, Україна.