UDC 004.9

***M. D. GODLEVSKYI, A. A. GOLOSKOKOVA, A. A. CHIPIZHENKO***

# MEDIUM-TERM PLANNING INFORMATION TECHNOLOGY FOR QUALITY IMPROVMENT OF THE SOFTWARE DEVELOPMENT PROCESS BASED ON THE CMMI MODEL

Розглядається технологія для вирішення задачі середньострокового планування (динамічна постановка задачі) покращення якості процесу розробки програмного забезпечення на основі моделі CMMI. Розглянуто тестовий приклад для другого рівня зрілості організації розробника програмного забезпечення. Плановий період складається з трьох підперіодів, а цільовий профайл може містити довільну кількість практик. Проведено аналіз отриманих результатів, які є вихідною інформацією для статичної постановки задачі.

**Ключові слова:** динамічна постановка задачі, розробка програмного забезпечення, модель CMMI, другий рівень зрілості, цільовий профайл.

Рассматривается технология решения задачи среднесрочного планирования (динамическая постановка задачи) улучшения качества процесса разработки программного обеспечения на основе модели CMMI. Рассмотрен тестовый пример для второго уровня зрелости организации разработчика программного обеспечения. Плановый период состоит из трёх подпериодов, а целевой профайл может содержать произвольное количество практик. Проведен анализ полученных результатов, которые являются выходной информацией для статической постановки задачи.

**Ключевые слова:** динамическая постановка задачи, разработка программного обеспечения, модель CMMI, второй уровень зрелости, целевой профайл.

Information technology is considered for medium-term planning for quality improvement of the software development process based on the CMMI model for the dynamic problem statement. The information system forms a rational plan for the software development process improvement and also allows, within the defined planning period, to determine precisely those practices and focus areas of the maturity model, for which it is necessary to increase the capability level, under conditions of limited resources. The test case for the second level of maturity of the software company is considered. The scheduled period consists of three sub periods, and the target profile may contain any quantity of practices. The analysis of the obtained results, which are the input data of the information system that solves the static problem, is carried out.

**Keywords:** dynamic problem statement, software development, CMMI model, the second level of maturity, target profile.

**Introduction.** The qualitative characteristics of software are paid more and more attention all over the world. The competitiveness of the software company depends directly on the quality of the software development process, and for this reason, software quality assurance issues are one of the top priorities for such companies. The issues related to the organization and improvement of the software development process, the management of the developers team, the development, and implementation of software tools to support the software development lifecycle fall into the field of software engineering. Software development process has a hierarchical architecture, including a number of life cycle processes of the software system. At present, there are various standards governing the sets of lifecycle process of software systems, such as ISO-12207, CMMI [1, 2] and other. The purpose of these standards is to ensure the end product to meet market requirements and to satisfy end users needs. The flexibility of the continuous CMMI model [3] allows the developer company to evaluate the most important stages of the software development process, as well as the maturity level of the entire software development process.

CMMI model was formalized in work [4]. Also, mathematical dynamic model and algorithm for planning the improvement of the software development process quality were developed.

Studies have shown that the problem is NP-complex. Therefore, relevant studies are devoted to the further revision of the model and algorithm according to the adequacy of the software and the speed of work, as well as designing and developing an information system for solving the problem of medium-term planning directly.

**Problem statement.** The task of planning for the quality improvement of the software development process can be presented by a rolling planning problem, which involves the synthesis of static and dynamic models (models of current and medium-term planning). In work [5] static models of planning for the quality improvement of the software development process were developed on the basis of the CMMI maturity model, the initial information for which is the result of solving the dynamic problem.

In a dynamic problem description statement, the task is to determine the first choice of the software development process progress within a planned [0, *T*] to a higher quality level, where T is the duration of the planned period. According to the dynamic problem statement, quality is characterized by alternative options for the progress of the software development process to a higher level of maturity in accordance with the CMMI model, as well as strategies for using different types of resources to achieve a certain level of maturity.

The aim of the research is to develop an information technology of planning for the quality improvement of software development process based on a dynamic model.

**Model and algorithm for medium-term planning under dynamic problem statement.** During the development of information technology for solving the problem of medium-term planning to improve the quality of the software development process, a mathematical model was developed in work [6].

A model analysis of medium term planning for improving the quality of the software development process by the algorithms which can be used in the problem solution showed the relevance of the time-series technique.

This method features a multi-step process, at each *t*-th step of which many uncompetitive options are ignored.

An algorithm "Kiev broom" was implemented by Ukrainian scientists V. S. Mikhalevich and N. Z. Shor on the basis of the time-series technique. The idea of this algorithm was used in the work to solve the problem of quality management process of software development. When using this algorithm for real data it has been shown that modern computers can not cope with the task since it is NP-complex. Therefore, there was a need to improve it in order to reduce the time for the task implementation. It was suggested to formulate and solve individual optimization problems in the course of the works of the "Kiev broom" algorithm.

**Quality management technology of the software development process.** When designing the system, the diagram of the processes IDEF0 (Fig. 1) was used as the basis. This diagram describes the high-level representation of the quality management of the software development process based on the maturity model.

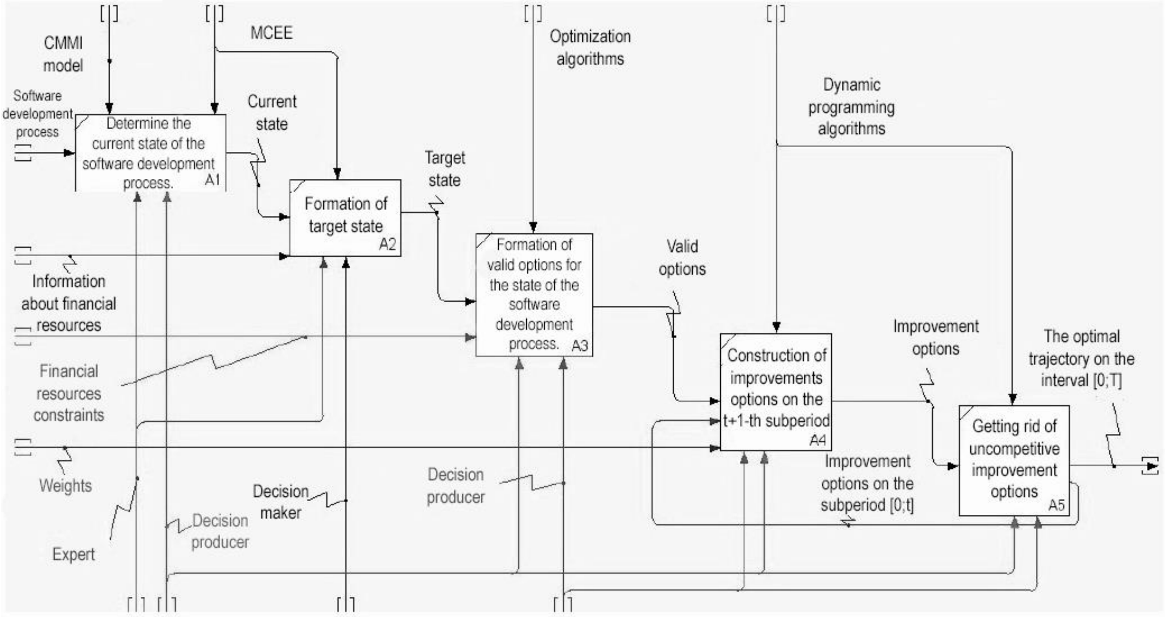
CMMI model
MCEE
Optimization algorithms
Dynamic programming algorithms
Software development process
Determine the current state of the software development process. A1
Current state
Formation of target state A2
Target state
Formation of valid options for the state of the software development process. A3
Valid options
Information about financial resources
Financial resources constraints
Construction of improvements options on the t+1-th subperiod A4
Improvement options
The optimal trajectory on the interval [0;T]
Getting rid of uncompetitive improvement options A5
Improvement options on the subperiod [0;t]
Weights
Expert
Decision producer
Decision maker
Decision producer

Fig. 1 – Process diagram

According to the technology of solving the problem, the first business process A1 determines the current state of the software development process. As an input data, the initial state of the software development process of the investigated software company is proposed. The software development process is evaluated according to the CMMI model. At the same time, the evaluation is carried out at the level of the initial state of the individual practices, which are defined by such a concept as the "level of capability" [7] in the range from zero to three.

In this work, the experts from the investigated software company who assessed the "level of capability" for practices were involved assess the current state of the software development process and the performance testing of the developed technology. The processing of this information was carried out by analysts-individuals who formulate solutions based on the methodology of collective expert evaluation [8]. The result of the business process A1 is the current state of the software development process.

Business process A2 is intended for the formation of a target profile, which defines the software company goals to improve the quality of the software development process. These goals are set by the decision-maker basing on the financial constraints, as well as the expert assessments of the required resources to achieve the goals set separately for each practice of the focus areas.

The aforementioned dynamic model and algorithm for quality management of software development process based on the model of maturity of CMMI are used for the formation of an optimal plan for the improvement of the software development process (business processes A3–A5). The initial state of the software development process, the target profile, the necessary financial resources for each practice belonging to the target profile; and the restrictions on financial resources that can be used in this planning period are used as input data.

The business process A3 is intended to form the sets of valid options of the state of the software development process on each considered scheduled sub period, taking into account the constraints on the financial resources allocated in each planning sub period, these resources can be carried over to the next ($t$+1) sub period, if they are not spent on *t*-th sub period.

Business processes A4, A5 implement the dynamic model of a task, which is an additive target function with financial constraints and variables that determine the level of capabilities for individual practices that are part of the target profile. An algorithm for sequential analysis of options is used for the formation of the optimal plan of the software development process improvement program. At each step of this algorithm, uncompetitive options for the improvement of the software development process are rejected. In the course of the quality management technology for software development process, an optimal trajectory for the software development process is formed.

**Tools for information technology building.** To construct an information system that solves the investigated problem, it was necessary to transform the mathematical model and algorithms of its processing into the corresponding components of the software.

The main programming language was C #, which is an object-oriented, strongly typed programming language for the .NET platform, it was developed by Anders Gailsberg, Scott Wiltamut, and Peter Gold, under the auspices of Microsoft Research.

Most of the calculations necessary for solving the problem are related to the solution of complex optimization problems, therefore all mathematical operations of the information system are performed using the Microsoft Solver Foundation 3.1 package – a set of development tools for mathematical programming, modeling and solving optimization problems.

To construct the information system, an appropriate data model was created. This data model accurately describes the mathematical and algorithmic models which were used in this work. ERWin Modeler [9] is a tool for designing and documenting databases. ERWin Modeler allows to create, document, and maintain databases. This tool was used as a tool for designing the necessary data model.

MS SQL Server served for storing and processing the data. The main advantages of this tool is high-speed of data processing, as well as the presence of built-in debugging capabilities of data models.

The Microsoft Visual Studio 2015 product was taken as an integrated development environment (IDE).

When designing the information system, 3-level "client-server" architecture with a dedicated application server was selected as an architecture model of the software system. This architecture model has another node called Application Server, which hosts the entire business logic, resulting in the database server being unloaded and only used for the data accessing. This model was chosen basing on the following benefits:

1) maximum scalability;

2) unloading the database server, opportunity to increase the performance of the entire system by configuring the application server according to the tasks that are being solved.

Fig. 2 shows the component deployment diagram within the client-server architecture with a dedicated application server. This diagram shows the physical relationships between software and hardware components of the system.

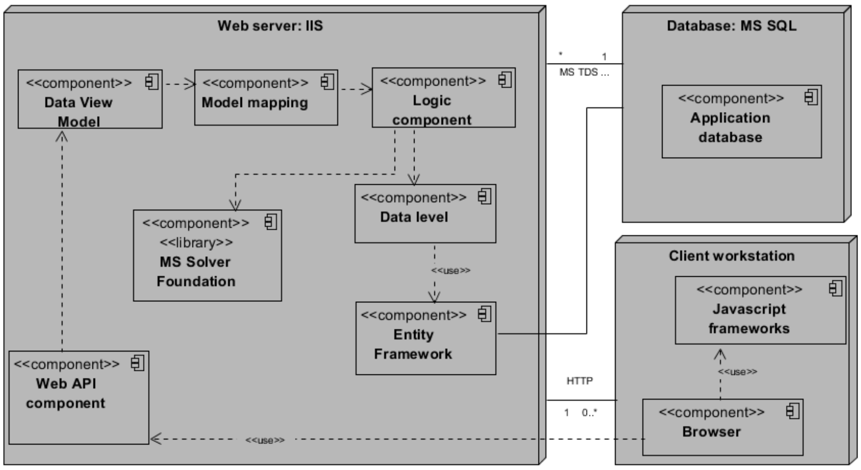
Web server: IIS
<<component>> Data View Model
<<component>> Model mapping
<<component>> Logic component
<<component>> <<library>> MS Solver Foundation
<<component>> Data level
<<use>>
<<component>> Entity Framework
<<component>> Web API component
<<use>>
*
1
MS TDS ...
Database: MS SQL
<<component>> Application database
Client workstation
<<component>> Javascript frameworks
<<use>>
<<component>> Browser
HTTP
1
0..*

Fig. 2 – Component diagram of the information system

We will briefly consider the placement diagram of the software system main components. Within the database server (Database: MS SQL) Application Database component that encapsulates the database and all the necessary configuration components has been dedicated. The application server (IIS Web Server) includes the following components required for the system:

- Entity Framework – ORM library;
- Data Level – this component works with the input data;
- MS Solver Foundation – library for mathematical calculations;
- Logic Component – a component that contains the basic mathematical and algorithmic logic of the software system;
- Data View Model i Model Mapping – infrastructure-level components that act as a middleware between business logic and the presentation level of the system (Web API component).
- The Client WorkStation includes two components:
- JS Frameworks – a set of tools for implementing a part of business logic on the client side;
- Browser – customer-side software, which aims to provide the user with the ability to interact with resources that are located on a web server using a graphical user interface.

**Software testing.** For a test example, the software development process of an "N" IT company is considered. The input data, as well as the target profile of the software development process, are shown in Table 1. The initial and target state of practice is measured in the "level of capability" scale in the range from zero to three. In our case, the initial state of the practice of local areas for the second level of maturity and the target one at the end of the planning period are determined basing on experts' assessments. The "Required Resources" column shows funding needed to move from zero/ first to the second/third level of capability. The decision of the dynamic task of forming a plan for improving the software development process is considered in the planning period, consisting of three sub periods. As it can be seen from the given data, 32200 c.u. of resources are needed for the full transfer of all practices to the required level. For this example, 25700 c.u. of resources (~80 %) were allocated for all sub periods. The investigated period was divided into 3 sub periods for each of them 8000, 10 000 and 7700 units of resources were allocated, respectively.

Table 1 – The target profile of the software development process of an "N" IT company.
The software development process target profile

| Maturity level | 2 | | | | | |
|---|---|---|---|---|---|---|
| Категорія | Project management | | | | | |
| **Focus area (FA)** | **SAM – Supplier Agreement Management** | | | | | |
| Goasl | SAM1 | | | SAM2 | | |
| Practices (P) | 1 | 2 | 3 | 4 | 5 | 6 |
| Initial state | 0 | 0 | 0 | 0 | 0 | 0 |
| Target profile | 2 | 2 | 2 | 2 | 2 | 2 |
| Needed resources, c.u. (first/second levels) | 1000/2200 | 1000/ 2200 | 1000/2200 | 1000/2200 | 1000/2200 | 1000/2200 |
| Weights for P | 0.5 | 0.25 | 0.25 | 0.3 | 0.3 | 0.4 |
| Weights for goals | 0.5 | | | 0.5 | | |
| Weights for FA | 0.4 | | | | | |
| **Focus area (FA)** | **REQM – Requirements Management** | | | | | |
| Goasl | REQM1 | | | | | |
| Practices (P) | 4 | | | 5 | | |
| Initial state | 1 | | | 1 | | |
| Target profile | 2 | | | 2 | | |
| Needed resources, c.u. (first/second levels) | 4300 | | | 4300 | | |
| Weights for P | 0.6 | | | 0.4 | | |
| Weights for goals | 1 | | | | | |
| Weights for FA | 0.3 | | | | | |
| **Focus area (FA)** | **PMC – Project Monitoring and Control** | | | | | |
| Goasl | PMC1 | | | | | |
| Practices (P) | 1 | | | 2 | | |
| Initial state | 2 | | | 2 | | |
| Target profile | 3 | | | 3 | | |
| Needed resources, c.u. (first/second levels) | 2200 | | | 2200 | | |
| Weights for P | 0.8 | | | 0.2 | | |
| Weights for goals | 1 | | | | | |
| Weights for FA | 0.3 | | | | | |

Based on the above initial data, the following results were obtained. As a result of the first sub period, the following practices were promoted: PMC_1_1; SAM_2_5; SAM_2_6; SAM_1_1.

Four practices have moved to a new level of capability, two of them (SAM_1_1, SAM_2_6) have moved from 0 to 2 level of capability. It cost 7600 c.u. of resources, 400 c.u. were moved to the next sub period.

As a result of the second sub period, the following practices were promoted: REQM_1_4; SAM_2_4; SAM_1_2; SAM_1_3.

Four practices have moved to a new level of capability, two of them (SAM_2_4, SAM_1_2) have moved from 0 to 2 level of capability, 10400 c.u. of resources were spent, 700 c.u. were moved to the next sub period.

As a result of the third sub period, the following practices were promoted: REQM_1_5; PMC_1_2.

Two practices have moved to a new level of capability, 8400 c.u. of resources were spent, 1900 c.u. of resources remained. Figure 3 depicts a histogram showing the process of raising the level of capability for specified practices during the three test sub periods.

**Conclusions.** The developed information technology solves the problem of medium-term planning of improving the quality of the software development process based on the CMMI model and forms a rational plan for developing the software development process. This allows within the defined planning period, under conditions of limited resources, to determine precisely those practices and focus areas of the CMMI model for which it is necessary to increase the level of capability. The mentioned above technology compared with the system [10] is characterized by higher productivity by solving individual optimization tasks at each step of the

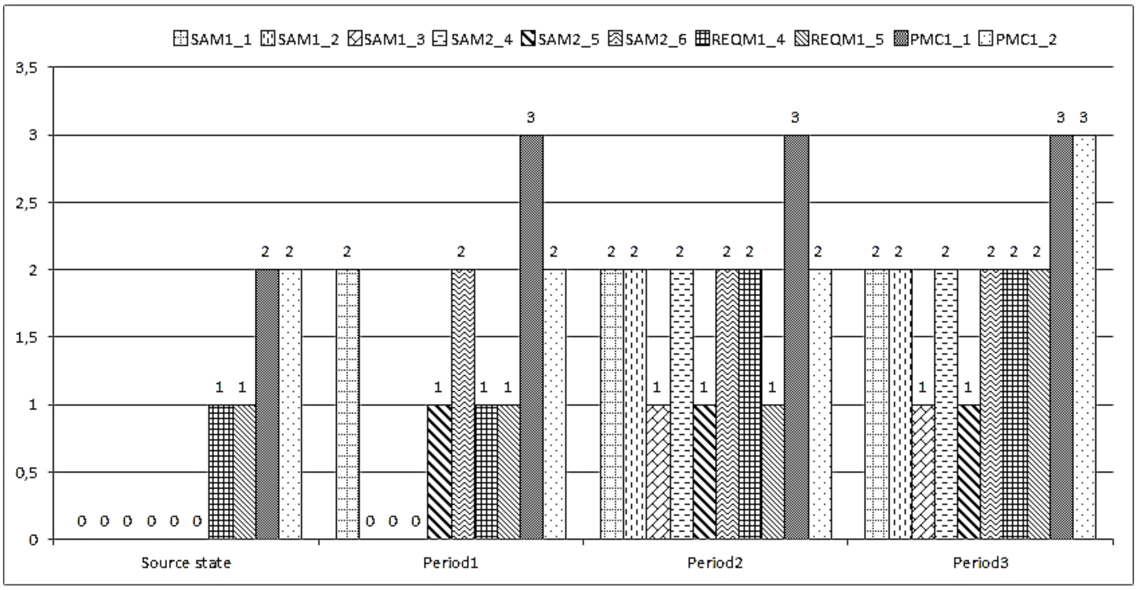

Fig. 3 – Results of calculations for the category "Project Management"

algorithm to identify uncompetitive options for the improvement of the software development process. This question is particularly relevant because the problem under consideration is NP-complete. The results obtained in the first sub period are the input data of the information system that solves the static problem.

The purpose of further research is to test the developed information system on full-scale data volumes, as well as to study the information system in terms of the problem solution speed when increasing its dimensionality.

As part of the further research, it is planned to design and develop an information system that will integrate the developed system and the system for solving static problems.

## References

1. ISO/IEC 12207:2008, Information technology – Software life cycle processes / ISO/IEC 12207:2008, Information technology – Software life cycle processes. International Organization for Standardization. – 2008. – 122 p.
2. *Chrissis M. B.* CMMI: Guidelines for Process Integration and Product Improvement / *M. B. Chrissis, M. Konrad, S. Shrum.* – Addison-Wesley. – 2003. – 688 p.
3. *Mutafelija, B.* Process improvement with CMMI v1.2 and ISO standards / *B. Mutafelija.* – Auerbach Pubs. – 2009. – 406 p.
4. *Годлевский М. Д.* Динамическая модель и алгоритм управления качеством процесса разработки программных систем на основе модели зрелости / *М. Д. Годлевский, И. Л. Брагинский* // Проблемы информационных технологий. – Херсон : ОЛДИ-Плюс, 2012. – С. 6–13.
5. *Годлевский М. Д.* Синтез статических моделей планирования улучшения качества процесса разработки программного обеспечения / *М. Д. Годлевский, А. А. Голоскокова* // Східно–Європейський журнал передових технологій. – 2015. – № 3/2 (75). – С. 23–29.
6. *Годлевский М. Д.* Динамическая модель планирования улучшения качества процесса разработки программного обеспечения / *М. Д. Годлевский, Э. Е. Рубин, А. А. Голоскокова* // Вісник Національного технічного університету "ХПІ". Збірник наукових праць. Серія: Системний аналіз, управління та інформаційні технології. – Х. : НТУ «ХПІ» – 2015. – № 58 (1167). – С. 3–6.
7. *Годлевский М. Д.* Принципы моделирования оценки и управления качеством процесса разработки программного обеспечения / *М. Д. Годлевский, В. А. Шеховцов, И. Л. Брагинский* // Восточно-Европейский журнал передовых технологий. – Харьков. – 2012. – № 5/3 (59). – С. 45–49.
8. *Крючковский В. В.* Интроспективный анализ. Методы и средства экспертн
9. ого оценивания / *В. В. Крючковский, Э. Г. Петров, Н. А. Соколова, В. Е. Ходаков.* – Херсон : Гринь Д. С., 2011. – 168 с.
10. *Павловская Т. А.* C#. Программирование на языке высокого уровня / *Т. А. Павловская* – СПб. : Питер, 2009 – 432с.
11. *Годлевский М. Д.* Информационная технология управления качеством процесса разработки программного обеспечения / *М. Д. Годлевский, И. Л. Брагинский* // Восточно-Европейский журнал передовых технологий. – Харьков. – 2013. – № 2/9 (62). – С. 63–67.

## References (transliterated)

1. *ISO/IEC 12207:2008, Information technology – Software life cycle processes / ISO/IEC 12207:2008, Information technology – Software life cycle processes. International Organization for Standardization*, 2008, 122 p.
2. Chrissis M. B., Konrad M., Shrum S. *CMMI: Guidelines for Process Integration and Product Improvement*. Boston, MA, USA, Addison-Wesley Publ, 2003, 688 p.
3. Mutafelija, B. *Process improvement with CMMI v1.2 and ISO standards.* Boca Raton, FL, USA, Auerbach Publ., 2009, 406 p.
4. Godlevskiy M. D., Braginskiy I. L. Dinamicheskaya model' i algoritm upravleniya kachestvom processa razrabotki programmnykh sistem na osnove modeli zrelosti [Dynamic model and quality management algorithm of software systems development process based on maturity models]. *Problemy informatsionnykh tekhnologiy.* Herson, OLDI-Pljus Publ., 2012, pp. 6–13.
5. Godlevskiy M. D., Goloskokova A. A. Sintez staticheskikh modeley planirovaniya uluchsheniya kachestva protsessa razrabotki programmnogo obespecheniya [Static planning models synthesis of quality improvement of software development process]. *Shidno–Єvropejs'kij zhurnal peredovih tehnologij*. Kharkiv, NPP ChP «Tekhnologicheskiy Tsentr» Publ., 2015, no. 3/2 (75), pp. 23–29.
6. Godlevskiy M. D., Rubin Je. E., Goloskokova A. A Dinamicheskaya model' planirovaniya uluchsheniya kachestva protsessa razrabotki programmnogo obespecheniya [Dynamic planning model of quality improvement of software development process].*Visnyk NTU "KhPI"* [Bulletin of the National Technical University "KhPI"]. Kharkov, NTU "KhPI" Publ., 2015, no. 58, pp. 3–6.
7. Godlevskiy M. D., Shehovcov V. A., Braginskiy I. L. Printsipy modelirovaniya otsenki i upravleniya kachestvom protsessa razrabotki programmnogo obespecheniya [Modeling principles of assessment and control of quality in software development process]. *Vostochno-Evropejskij zhurnal peredovyh tehnologij.* Kharkiv, NPP ChP «Tekhnologicheskiy Tsentr» Publ., 2012, no. 5/3 (59), pp. 45-49.

8. Krjuchkovskiy V. V., Petrov Je. G., Sokolova N. A., Hodakov V. E. *Introspektivnyy analiz* [Introspective analysis]. *Metody i sredstva ekspertnogo otsenivaniya* [Methods and techniques of the expert estimation]. Herson, Grin' D. S. Publ., 2011. 168 p.
9. Pavlovskaya T. A. *C#. Programmirovanie na jazyke vysokogo urovnya* [C#. High-level language programming]. SPb, Piter Publ., 2009. 432 p.
10. Godlevskiy M. D., Braginskiy I. L. Informatsionnaya tekhnologiya upravleniya kachestvom protsessa razrabotki programmnogo obespecheniya [Information technology for the quality management of the software development process]. *Vostochno-Evropeyskiy zhurnal peredovykh tekhnologiy* Kharkiv. NPP ChP «Tekhnologicheskiy Tsentr» Publ., 2013, no 2/9 (62), 63–67pp.

*Received 15.09.2017*

*Бібліографічні описи / Библиографические описания / Bibliographic descriptions*

**Інформаційна технологія середньострокового планування покращення якості процесу розробки програмного забезпечення на основі моделі CMMI / М. Д. Годлевський, А. О. Голоскокова, А. О. Чипіженко** // Вісник Національного технічного університету «ХПІ». Серія: Системний аналіз, управління та інформаційні технології. – Х. : НТУ «ХПІ», 2017. – № 51 (1272). – С. 3–7. – Бібліогр.: 10 назв. – ISSN 2079-0023.

**Информационная технология среднесрочного планирования улучшения качества процесса разработки программного обеспечения на основе модели CMMI / М. Д. Годлевский, А. А. Голоскокова, А. А.Чипиженко** // Вісник Національного технічного університету «ХПІ». Серія: Системний аналіз, управління та інформаційні технології. – Харків : НТУ «ХПІ», 2017. – № 51 (1272). – С. 3–7. – Библиогр.: 10 назв. – ISSN 2079-0023.

**Medium-term planning information technology for quality improvement of the software development process based on the CMMI model / M. D. Godlevskyi, A A. Goloskokova, A. A. Chipizhenko** // Bulletin of National Technical University "KhPI". Series: System analysis, control and information technology. – Kharkov : NTU "KhPI", 2017. – No. 51 (1272). – P. 3–7. – Bibliogr.: 10. – ISSN 2079-0023.

*Відомості про авторів / Сведения об авторах / About the Authors*

***Годлевський Михайло Дмитрович*** **–** доктор технічних наук, професор, Національний технічний університет «Харківський політехнічний інститут», завідуючий кафедри «Програмна інженерія та інформаційні технології управління»; тел.: (057) 707-64-74; e-mail: god_asu@kpi.kharkov.ua

***Годлевский Михаил Дмитриевич*** – доктор технических наук, профессор, Национальный технический университет «Харьковский политехнический институт», заведующий кафедры «Программная инженерия и информационные технологии управления»; тел.: (057) 707-64-74; e-mail: god_asu@kpi.kharkov.ua

***Mykhailo D. Godlevskyi*** – Doctor of Technical Sciences, Full Professor, National Technical University "Kharkiv Polytechnic Institute", Head of the department of "Software Engineering and Management Information Technologies"; tel.: (057) 707-64-74; e-mail: god_asu@kpi.kharkov.ua

***Голоскокова Анна Олександрівна*** – Національній технічний університет «Харківський політехнічний інститут», старший викладач кафедри «Програмна інженерія та інформаційні технології управління»***;*** тел.: (057) 707-64-74; e-mail: semit.department.16@gmail.com.

***Голоскокова Анна Александровна*** – Национальный технический университет «Харьковский политехнический институт», старший преподаватель кафедры «Программная инженерия и информационные технологии управления», тел.: (057) 707-64-74; e-mail: semit.department.16@gmail.com

***Anna A. Goloskokova*** – National Technical University "Kharkiv Polytechnic Institute", senior lecturer of the department of "Software Engineering and Management Information Technologies"; tel.: (057) 707-64-74; e-mail: semit.department.16@gmail.com

***Чипіженко Олександр Олександрович*** – Національній технічний університет «Харківський політехнічний інститут», студент кафедри «Програмна інженерія та інформаційні технології управління»; тел.: (095) 692-80-04; e-mail: alex.chipizhenko@hotmail.com

***Чипиженко Александр Александрович*** – Национальный технический университет «Харьковский политехнический институт», студент кафедры «Программная инженерия и информационные технологии управления», тел.: (095) 692-80-04; e-mail: alex.chipizhenko@hotmail.com

***Alexandr A. Chipizhenko*** – National Technical University "Kharkiv Polytechnic Institute", student of the department of "Software Engineering and Management Information Technologies"; tel.: (095) 692-80-04; e-mail: alex.chipizhenko@hotmail.com