

*M. A. ZARUDNIEV, V. O. HUZHYA*

### **COST AND COMPLEXITY RESEARCH OF SOFTWARE DEVELOPMENT TO SOLVE THE PROBLEM OF INVENTORY MANAGEMENT**

The article describes the process of estimating the cost and complexity of software development for the task of inventory management of a commercial enterprise to improve the decision-making process at the stage of formation of requirements for similar software products. There are different approaches to inventory management of a commercial enterprise. The algorithms implemented in the system are based on the constraint theory tool – dynamic inventory buffer management. Inventory management according to the theory of restrictions allows to provide a high level of availability of goods with minimal inventory in the system. After analyzing the subject area of inventory management and the methodology of dynamic inventory buffer management, a list of functional and non-functional software requirements was formed, and a data model was designed. The information obtained was used to compile a list of logical files and transactions specific to a particular requirement. From the resulting list of files and transactions, it is determined which elementary data, elementary records and links to files they consist of. The information obtained was used to estimate the size of the software at non-aligned functional points. To take into account the impact of non-functional requirements on the complexity of software development, the VAF alignment factor was calculated, for which it is necessary to give a qualitative assessment of the impact of system-wide requirements on the complexity of the software. The estimation of the complexity of the development was obtained using the COCOMO II methodology, namely, using seven multipliers of the complexity for the case of the initial assessment at the stage of approval of requirements. The calculations were obtained for teams with different professional level and equipment, in the conditions of a significant tightening of the development process, a high level of certainty of risks, without a rigid schedule of work. The cost of development was calculated for several common programming languages: J2EE, C#, C++.

**Keywords:** software, evaluation methods of cost and complexity, COCOMO II method, the task of managing vending inventory based on demand.

*M. A. ЗАРУДНІЄВ, В. О. ГУЖВА*

### **ДОСЛІДЖЕННЯ ВАРТОСТІ ТА ТРУДОМІСТКОСТІ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ЗАДАЧІ УПРАВЛІННЯ ЗАПАСАМИ ПІДПРИЄМСТВА**

У статті описується процес оцінки вартості і трудомісткості розробки програмного забезпечення задачі управління запасами торговельного підприємства для поліпшення процесу прийняття рішень на етапі формування вимог до аналогічних програмних продуктів. Існують різні підходи до управління запасами торгового підприємства. Алгоритми, реалізовані в системі засновані на інструменті теорії обмежень – динамічному управлінні буфером запасів. Управління запасами з теорії обмежень дозволяє забезпечити високий рівень наявності товару при мінімальних запасах в системі. Проаналізувавши предметну область управління запасами і методологію динамічного керування буфером товарних запасів був сформований список функціональних і нефункціональних вимог до програмного забезпечення, а також спроектована модель даних. Отримана інформація була використана для складання переліку логічних файлів і транзакцій, характерних для конкретної вимоги. З отриманого списку файлів і транзакцій визначено, з яких елементарних даних, елементарних записів і посилань на файли вони складаються. Отримана інформація була використана для оцінки розміру програмного забезпечення в не вирівняних функціональних точках. Для обліку впливу функціональних вимог на трудомісткість розробки програмного забезпечення був порохований фактор вирівнювання VAF, для чого необхідно дати якісну оцінку впливу загальносистемних вимог на трудомісткість програмного забезпечення. Оцінка трудомісткості розробки була отримана з використанням методології COCOMO II, а саме з використанням семи множників трудомісткості для випадку первинної оцінки на етапі затвердження вимог. Розрахунки були отримані для команд з різним професійним рівнем і устаткуванням, в умовах значного посилення процесу розробки, високим рівнем визначеності ризиків, без жорсткого графіка робіт. Вартість розробки була розрахована для декількох поширених мов програмування: J2EE, C#, C++.

**Ключові слова:** програмне забезпечення, методи оцінки вартості та трудомісткості, метод COCOMO II, задача управління торговим інвентарем на основі попиту.

*M. A. ЗАРУДНІЄВ, В. А. ГУЖВА*

### **ИССЛЕДОВАНИЕ СТОИМОСТИ И ТРУДОЁМКОСТИ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ РЕШЕНИЯ ЗАДАЧИ УПРАВЛЕНИЯ ЗАПАСАМИ ПРЕДПРИЯТИЯ**

В статье описывается процесс оценки стоимости и трудоемкости разработки программного обеспечения задачи управления запасами торгового предприятия для улучшения процесса принятия решений на этапе формирования требований к аналогичным программным продуктам. Существуют различные подходы к управлению запасами торгового предприятия. Алгоритмы, реализованные в системе основаны на инструменте теории ограничений – динамическом управлении буфером запасов. Управление запасами по теории ограничений позволяет обеспечить высокий уровень наличия товара при минимальных запасах в системе. Проанализирован предметную область управления запасами и методологию динамического управления буфером товарных запасов был сформирован список функциональных и нефункциональных требований к программному обеспечению, а также спроектирована модель данных. Полученная информация была использована для составления перечня логических файлов и транзакций, характерных для конкретного требования. Из полученного списка файлов и транзакций определено, из каких элементарных данных, элементарных записей и ссылок на файлы они состоят. Полученная информация была использована для оценки размера программного обеспечения в не выровненных функциональных точках. Для учета влияния нефункциональных требований на трудоемкость разработки программного обеспечения был посчитан фактор выравнивания VAF, для чего необходимо дать качественную оценку влияния общесистемных требований на трудоемкость программного обеспечения. Оценка трудоемкости разработки была получена с использованием методологии COCOMO II, а именно с использованием семи множителей трудоемкости для случая первичной оценки на этапе утверждения требований. Расчеты были получены для команд с разным профессиональным уровнем и оборудованием, в условиях значительного ужесточения процесса разработки, высоким уровнем определенности рисков, без жесткого графика работ. Стоимость разработки была рассчитана для нескольких распространенных языков программирования: J2EE, C#, C++.

**Ключевые слова:** программное обеспечение, методы оценки стоимости и трудоемкости, метод COCOMO II, задача управления торговыми запасами на основе спроса.

© M. A. Zarudniev, V. O. Huzhyia, 2018

**Introduction.** Modern Ukrainian trade enterprises of small and large businesses are faced with the problem of optimal inventory management. Neglect of this type of management activity can lead to significant material losses.

The most common solution to this problem is to manage inventory without or partially using automated tools. However, often trade enterprises are limited in funds allocated for management functions. Therefore, the management of the enterprise has to either neglect careful planning of reserves and suffer losses, or invest significant funds.

To minimize the described costs, the best solution at the moment will be the introduction of a full-fledged automated inventory management system at the trading enterprise. This will reduce the number of employees engaged in accounting, speed up cost planning and reduce the impact of the human factor. The process of developing inventory management software is complex and requires significant labor costs. The cost and complexity of its development depends on the size of the software, the level of competence of the developer, the presence of previous experience in the development of similar systems and many other factors [1].

At the stage of identifying the requirements for the system there is a need to assess the cost and complexity of the implementation of certain of its functionality and qualities, the relevance of which the more complex and large-scale project. Only highly qualified experts in the subject area of inventory management, whose availability requires additional financial costs, are able to carry out such an assessment with high accuracy immediately.

The main task of estimating the cost and complexity of software development is to support decision – making for optimal allocation of project resources at the stage of requirements formation [2].

**Purposes and research problems.** The purpose of this work is to assess the cost and complexity of software development to solve the problem of inventory management.

To achieve this goal, the article solves the following tasks:

- the analysis of methods for assessing the cost and complexity of software development and the choice of the most appropriate;
- the analysis of the subject area of inventory management of the enterprise;
- the methodology of inventory management of commercial enterprise with current demand on the basis of the tool of the theory of restrictions is described;
- detailed functional and non-functional software requirements as well as data model are developed;
- the estimation of the size and on its basis calculated the complexity and development time.

**The subject** of the study is the process of assessing the cost and complexity of software development. The size of the software is estimated according to the method of functional points, and the complexity – according to the

COCOMO II model.

**The object** of the study is the software for solving the problem of inventory management of commercial enterprises. Functional and non-functional requirements to it were put forward on the basis of the analysis of the subject area and methodology of inventory management of commercial enterprise.

**Analysis of methods for estimating the cost and complexity of software development.** Among the considered models for estimating the cost and complexity of software development, the COCOMO II model was singled out as the most popular and having sufficient accuracy to make an adequate decision in the framework of requirements management [3].

The most detailed model that is used when the project is fully ready for development uses 17 multipliers of labor intensity (cost factors) and 5 factors that determine the scale of the project. The model itself has the following form:

$$PM = A \cdot SIZE^E \prod_{i=1}^{17} EM_i,$$

where  $SIZE$  – product size in KSLOC;

$EM_i$  – multipliers of labor intensity.

$$E = B + 0,01 \sum_{j=1}^5 SF_j,$$

where  $SF_j$  – scale factors;

$A = 2,94$ ;

$B = 0,91$ .

Values of  $EM_i$  and  $SF_j$  parameters are tabular [4].

Functional points method (Functional Points Analysis – FPA) is used to estimate software size in functional points. It allows you to estimate the size of the software product based on the functionality demanded by the customer and supplied by the developer. Evaluation of the number of functional points (FT) for the software product is derived from the data that are determined by the analysis of the information area and the study of the features of its future operation.

The evaluation procedure consists of the following steps [5]:

- 1) Definition of the boundary of the product.
- 2) Definition of data and transactions (operations on the data).
- 3) Counting the number of non-aligned function points.
- 4) Determination of factor values align (Value-Alignment Factor – VAF).
- 5) Counting aligned function points.

The complexity of the files depends on the number of logical data groups (Rets) and the number of unique data fields (DETS). The complexity of transactions depends on the number of different files modified or read in a transaction (FTR) and the number of unique data fields (DET). Each RET/FTR or DET component can have one of three difficulty levels: L – low, A – average, H – high. The method of function points of the matrix determines the complexity of the data file and transaction.

To account for the impact of non-functional requirements on the complexity of software development, there is a VAF alignment factor. For its calculation it is necessary to give a qualitative assessment of the impact of system-wide requirements on the complexity of the software.

Having received the necessary estimate of the size of the software, you can calculate the complexity of its development. To do this, you need to transfer the software size to SLOC according to the table 1:

Table 1 – Estimation of the number of lines of code of one functional point

Computer language	Estimation of the number of lines of code of one functional point		
	Likely	Optimistic	Pessimistic
C	148	9	704
C++	60	29	178
C#	59	51	66
J2EE	61	50	100
JavaScript	56	44	65
Visual Basic	50	14	276

**Analysis of the methodology of stock management of the commercial enterprise based on the current demand.** Dynamic buffer management (*eng.* DBM) – a tool of the theory of constraints (THAT), which allows you to effectively manage the reserves of the enterprise, focusing on the actual demand of consumers [6].

The purpose of the DBM algorithm is to track the current buffer level to decide whether the target level is too low or too high.

Key TO ideas for inventory management [7]:

1) Take into account not only the stock in stock, but also the goods in transit, that is, all open purchase orders must be part of the mechanism to ensure the availability of goods. The target level determines the inventory buffer, including both on-hand inventory and open orders.

2) The target level does not change until a clear signal is received that it is unacceptable.

3) Regular replenishment of the target level.

Inventory buffer-a target level of inventory that is set and maintained in the supply chain to meet customer demand, taking into account fluctuations in demand and delivery times. It is necessary in order to maintain the right amount of inventory in the system and to ensure constant availability of the product for the customer. At the same time, the system should not have excess reserves, freeze working capital, that is, the amount of reserves in the system should be minimal.

The buffer determines the quantity of the required goods for sale from delivery to delivery. Its size is analyzed by the system on a daily basis and automatically changes depending on the intensity of the balance change. If the balance dominates in the green zone – the system reduces the buffer in the red-increases to the upper value of the buffer (replenishment level).

When the stock in any link of the supply chain falls below the level of the replenishment, it must immediately

be replenished to this level. Using an immediate response to the actual sale eliminates the need to forecast sales in the short and medium term [8].

The inventory buffer is divided into several zones, which are shown in Fig. 1 [9].

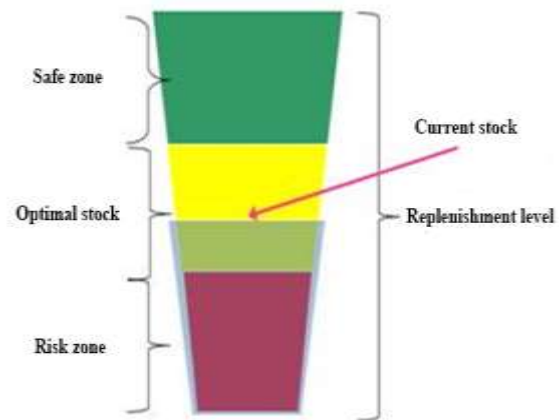


Fig. 1. Structure of buffer stocks

Buffer management is an automated procedure for analyzing the inventory buffer and automatically making changes that reflect changes in demand up or down. Properly configured, buffer management significantly reduces replenishment time and dramatically increases the number of timely deliveries [10].

The initial buffer size is calculated as the maximum consumption per delivery cycle. The delivery cycle takes into account the duration and frequency of delivery. The maximum consumption is determined from previous sales statistics or is set manually. TOC buffer management is able to analyze actual consumption and make recommendations on targets, according to the logical rule above. The rule suggests taking into account the high consumption during the replenishment period, taking into account the variability in the various factors of replenishment time. The higher the target service level, the more variability needs to be considered.

The size of the item buffer can be set individually for each of the warehouses throughout the supply chain, including retail outlets, retail chains and dealers' warehouses.

Individual parameters of dynamic buffer [11]:

- the level of replenishment basic units of measurement;
- red and green zone size as a percentage of replenishment level, replenishment period (in days);
- red and green zone penetration audit periods (in days), minimum inventory level;
- accounting for the supply forecast for the replenishment period.

After the settlement of the target buffer level, the system makes the ordering provider of goods.

**Defining software requirements.** After analyzing the subject area of inventory management and the methodology of dynamic inventory buffer management, you can create a list of functional requirements for the software in General:

- 1) User Authorization in the system.
- 2) Inventory control.
- 3) Dynamic buffer management of the inventory.
- 4) Automatic formation of an order for the supply of goods.

- 5) Export the report the dynamics of buffer stocks.
- 6) Receiving of user manual.

In addition to functional requirements, the product is also subject to system-wide requirements of General Systems Characteristics (GSCs), which restrict developers in choosing a solution and increase the complexity of development. The following list of non-functional requirements is selected for the study:

1) Data exchange. The process of communication between the database server and the application server must be fault-tolerant.

2) Performance. Operations available in the software must not be performed for more than 5 seconds.

3) Hardware resource limits.

4) Transactional load. The time to recover from a failure caused by a hardware power failure (other external factors), not a fatal operating system failure, should not exceed the time required to restart the server and refresh the connection page.

5) Intensity of user interaction. Time of work with is limited to the user's working schedule.

6) Ergonomics. The information system must meet the standards of the common user access (CUA) IBM.

7) Ease of administration. The functionality of the database administration needs to be made to a custom toolbar.

**Results of calculation.** As a result of the analysis of functional requirements to the inventory management software of the trading enterprise on the basis of the FPA method, the size of the software in the functional points was calculated:  $FP = UAF * VAF = 340,24$ .

As a result of the study of the complexity of software development using the COCOMO II method, the following dependencies of development complexity on labor input factors were obtained. The calculations were obtained for teams with different professional level and equipment, in the conditions of a significant tightening of the development process, a high level of certainty of risks, without a rigid schedule of work. The complexity of the development was calculated for several of the most common programming languages – J2EE, C#, C++ and listed in the table 2.

Based on these data, the estimated cost of developing a commercial enterprise inventory management software was calculated on the condition of 100% employment of all team members throughout the development period using the formula:

$$\sum_{i=1}^7 v_i \cdot \frac{PM}{7},$$

where  $v_i$  – salary of the  $i$ -th team member (UAH/month);

$PM$  – the complexity of software development for this team (man-months).

Table 2 – Estimation of labor input for different programming languages

Team of developers	Estimation of labor intensity (man-months)		
	J2EE	C#	C++
Very weak	296,6	285	290,8
Weak	145,2	139,8	142,5
Satisfactory	77,2	74,5	75,9
Average	44,3	42,8	43,6
Strong	22	21,3	21,7
Very strong	11,5	11,1	11,3

The average salary of the worker as a programmer in terms of UAH (at the rate of NBU from 01.12.2018 [12]) is listed in the table 3.

Table 3 – Average salary of a worker as a programmer

	J2EE	C#	C++
Junior Software Engineer	19670	18265	19670
Software Engineer	53390	50580	59010
Senior Software Engineer	98350	94135	95540
Technical Lead	122235	121673	108888
System Architect	167195	134880	137690

The results are listed in the table 4.

Table 4 – Estimated cost of software development for inventory management of commercial enterprise (UAH)

Team of developers	Development cost		
	J2EE	C#	C++
Very weak	171410	160170	177030
Weak	205130	192485	216370
Satisfactory	283810	268355	292240
Average	386375	371763	381458
Strong	646300	596563	614688
Very strong	781180	727228	724278

Thus, it can be concluded that:

1) Development of projects in C++ for qualified teams in Ukraine is the cheapest, but for the least qualified teams on the contrary – the most expensive;

2) The cheapest development tool for all skill levels of teams in Ukraine is the use of the C language#.

3) J2EE is practically the most expensive development tool (except for C++ for teams with small qualifications).

The presence of a huge number of factors that directly or indirectly affect the process of software development, does not allow to give an accurate assessment of the software at the stage of requirements formation.

**Conclusion.** In this research work, based on the analysis of the subject area of the problem of inventory management of a trading enterprise, a study of the cost

and complexity of the development of appropriate software was conducted.

#### References

1. Ньютон Р. *Управление проектами от А до Я*. Москва: Альпина Бизнес Букс, 2007. 180с.
2. Архипенков С. *Лекции по управлению программными проектами*. URL: [http://citforum.ru/SE/project/arkhipenkov\\_lectures/13.shtml](http://citforum.ru/SE/project/arkhipenkov_lectures/13.shtml) (дата обращения 12.10.2018).
3. Садовский И. Д. Применение модели СОСОМО II для оценки разработки программного обеспечения в Windows проектах. *Экономика и бизнес: теория и практика*. 2016. № 10. С. 102–106.
4. Шанченко Н. И. Оценка трудоемкости разработки программного продукта: методические указания. Ульяновск: УлГТУ, 2015. 40 с.
5. Сидоров Н. А., Баценко Д. В., Василенко Ю. Н., Щebetин Ю. В., Иванова Л. Н. Методы и средства оценки стоимости программного обеспечения. *Проблемы системного подходу в економіці*. 2004. № 7. С. 113 – 118.
6. Нагапетьянц Н. *Экономика предприятия: Учебник для вузов*. Москва: ЮНИТИ-ДАНА, 2007. 670 с.
7. Коуэн О., Федурко Е. *Основы Теории Ограничений*. Таллинн: TOC Strategic Solutions, 2012. 331 с.
8. Детмер У. *Теория ограничений Голдратта: Системный подход к непрерывному совершенствованию*. Москва: Альпина Бизнес Букс, 2008. 444 с.
9. Речкалов В. *Динамическое управление буфером*. URL: <http://tocpeople.com/2013/10/dinamicheskoe-upravlenie-buferom/> (дата обращения 10.10.2018).
10. Правук О. *Оптимизация запасов и ассортимента. Подход теории ограничений*. URL: <https://uppravuk.net/optimizatsiya-zapasov-i-assortimenta-podhod-teorii-ogranichenij/> (дата обращения 10.10.2018).
11. Корбетт Т. *Учёт прохода: Управленческий учёт по ТОС*. Киев: Необхідно і достатньо, 2009. 240 с.
12. *НБУ, Офіційний курс доллара на 01 грудня 2018*. URL: <https://minfin.com.ua/currency/nbu/usd/> (дата обращения 01.12.2018).
13. [http://citforum.ru/SE/project/arkhipenkov\\_lectures/13.shtml](http://citforum.ru/SE/project/arkhipenkov_lectures/13.shtml) (accessed: 12.10.2018).
3. Sadovskiy I. D. Primenenie modeli COCOMO II dlja ocenki razrabotki programmnogo obespechenija v Windows proektah. *Ekonomika i biznes: teoriia i praktika* [Application of COCOMO II model for evaluation of software development in Windows projects. Economics and business: theory and practice]. 2016, no. 10, pp. 102–106.
4. Shanchenko N. I. *Ocenka trudoemkosti razrabotki programmnogo produkta: metodicheskie ukazaniya* [Evaluation of the complexity of software development: guidelines]. Ulyanovsk, UIHTU Publ., 2015. 40 p.
5. Sydorov N. A., Batsenko D. V., Vasilenko Yu. M., Shchetin Yu. V., Ivanova L. N. *Metody i sredstva ocenki stoimosti programmnogo obespechenija* [Methods and tools of cost estimation software]. *Problemy systemnogo pidkhodu v ekonomitsi*. 2004, no. 7, pp. 113 – 118.
6. Nahapetians N. *Ekonomyka predpriyatiya: Uchebnyk dlja vuzov* [Enterprise Economics: Textbook for universities]. Moscow, YUNYTY-DANA Publ., 2007. 670 p.
7. Koujen O., Fedurko E. *Osnovy Teorii Ogranichenij* [Fundamentals of the theory of Constraints]. Tallinn, TOC Strategic Solutions Publ., 2012. 331 p.
8. Dettmer H. William. *Goldratt's Theory of Constraints: A Systems Approach to Continuous Improvement*. Asq Pr, 1997. 378 p. (Russ. ed.: Detmer W. *Teorija ogranichenij Goldratta: Sistemnyj podhod k nepreryvnomu sovershenstvovaniyu*. Moscow, Alpina Biznes Buks Publ., 2008. 444 p.).
9. Rechkalov V. *Dinamicheskoe upravlenie buferom* [Dynamic buffer management]. Available at: <http://tocpeople.com/2013/10/dinamicheskoe-upravlenie-buferom/> (accessed: 10.10.2018).
10. Pravuk O. *Optimizacija zapasov i assortimenta. Podhod teorii ogranichenij* [Optimization of stocks and assortment. The approach of the theory of constraints]. Available at: <https://uppravuk.net/optimizatsiya-zapasov-i-assortimenta-podhod-teorii-ogranichenij/> (accessed: 10.10.2018).
11. Corbett T. *Throughput Accounting*. North River Press, 1998. 174 p. (Russ. ed.: Corbett T. *Uchjot prohoda: Upravlencheskij uchjot po TOC*. Kiev, Neobkhdno i dostatno Publ., 2009. 240 p.).
12. *NBU, Ofitsiyni kurs dolara na 01 hrudnia 2018* [NBU, the Official dollar exchange rate on December 01, 2018]. Available at: <https://minfin.com.ua/currency/nbu/usd/> (accessed: 01.12.2018).

#### References (transliterated)

1. Niuton R. *Upravlenie proektami ot A do Ja* [Project Management A to Z]. Moscow, Alpina Biznes Buks Publ., 2007. 180 p.
2. Arkhpenkov S. *Lekcii po upravleniju programmnyimi proektami* [Lectures on the management of software projects]. Available at:

Received 04.12.2018

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

**Заруднев Максим Андрійович (Заруднев Максим Андреевич, Zarudniev Maksym Andriiovych)** – бакалавр, Національний технічний університет «Харківський політехнічний інститут», студент програмної інженерії та інформаційних технологій управління; м. Харків, Україна; ORCID: <https://orcid.org/0000-0002-1061-1249>; e-mail: [zarudnev.maxim@gmail.com](mailto:zarudnev.maxim@gmail.com)

**Гужва Віктор Олексійович (Гужва Виктор Алексеевич, Huzhva Viktor Oleksiiovych)** – кандидат технічних наук, доцент, Національний технічний університет «Харківський політехнічний інститут», професор кафедри програмної інженерії та інформаційних технологій управління м. Харків, Україна; ORCID: <https://orcid.org/0000-0001-6832-4480>; e-mail: [guzhva.v.a@gmail.com](mailto:guzhva.v.a@gmail.com)