The Information Community of the Arctic in Russia: Evaluation of the Expenses for the IT Projects Development, Characteristics of the Labour Costs Calculating

Ivan V. Evdokimov, Alexander S. Khaluimov, Nikita V. Sokolov & Sergey E. Golokhvastov

In the Arctic conditions of northern Siberia, the IT-industry represents an important platform for providing globally competitive employment. Hence, evaluation of the expenses related to IT-development is a highly important question for the information community of the Arctic. Nowadays, software solutions provided by the 1C *company are leading in the fields of public administration, municipal board and business in the aforementioned region. Adequate assessment of the cost and development time has an important role in the software development. In the field of information technology (IT) specialists often use different metrics based on the software functionality – function-oriented metrics. The models used for evaluation contain a number of parameters. Each of these parameters has a special coefficient, which is based on the company standard. Their values have a direct impact on the software developing cost calculation. Among all of the functionally-oriented assessing methods we can give a special credit to the Function Points (FP) method. The basis of its use is the correlation of parameters of future programming with tables which include special coefficients. To calculate the number of function points, the cost, and the time of IT project development we use special formulas which are based on varieties of the COCOMO model and FP-tables. A special feature of the FP method is a table including coefficients of the empirical complexity for each programming language and IDE, based on the number of operators for one function point. Consequently, this method allows us to estimate the value of the product development not only in terms of its functionality, but also in terms of applied tools. Thus, the subject of this research will be the definition of the value factors which are used to calculate the FP-evaluations on the* 1C v.8.3 *platform. It will be based on statistical analysis of several regional IT projects. To improve the adequacy of FP-models, we will consider stakeholders of the* 1C*-based IT-projects as objects of our research. Recent software engineering developments allow us to move away from clichés about the High North, which has been considered only as a supplier of natural resources for many years.*

Necessity & Perspectives of the Arctic Information Community's Development: Clichés about the Far North Exclusively as a Resource Provider

As history shows us, the fast exploration and development of new territories comes very soon after the time of its discovering: it's about the building of infrastructure, which differs from one age to another, the migration of a people to new lands, and the advent of a modern civilization there as a consequence of all those things. The Far North is an exception to these "exploration traditions." The main reason is the location of these territories. It is difficult just to reach them,

Ivan V. Evdokimov is Associated Professor and Alexander S. Khaluimov, Nikita V. Sokolov & Sergey E. Golokhvastov are students at the Siberian Federal University, Krasnoyarsk, Russian Federation.

therefore the challenge of exploring and developing these areas are even more pronounced, even more so a few centuries ago.

Eventually many of these barriers had been overcome. People got the possibility to explore the North, which they couldn't do before. They began establishing research centers and attracted scientists to the Far North. Generally, they started a serious development of the Far North.

If we look on the continental Far North sides, we will see many similarities: living there was quite hard for a common citizen, especially because of the climate, so few people sought to develop it. But with the growth of new technologies, people started making expeditions to these lands to determine whether profits were possible from their development.

Here we come to the main reason for Far North exploration – the exploitation of resources. We examine the territories of the Russian Federation, as our country is very well known as one of the main providers of resources in the world. Furthermore, it is not a secret that Russia has many territories which have very weak infrastructure but high resource potential, including the Far North and Far East of Russia e.g. Norilsk, where we have copper and nickel deposits, or Chukchi Peninsula with its gold and tin.

Maybe after finding a legitimate reason to pursue Far North exploration it became apparent that we did not really need to make these lands attractive for permanent settlement, because we can just build some bases which will only be used for resource production. Fortunately, these days most have put this strategy aside, so we can see how people on these lands are trying to build a high quality of living as they would have in more comfortable territories.

Nevertheless, this colonial model of territorial expansion has left its mark on the Far North, and especially on the public perception of it – that the region is just a resources provider. More needs to be understood about this model.

The colonial model is a forcible model of territorial expansion, so we can make a decision that the main issue is that this model is directed towards "deriving" every possible resource. Human resources are no exception. Territories for whom this model is applied are often referred to as colonies. Their main characteristics are:

- Lack of political autonomy, and a special status which usually differs from the status of common provinces;
- Economic abuse of natural and human resources for the benefit of the metropolis. It often slows down economic growth and results in the degradation of the colony;
- Capturing of territory by the metropolis, and occupation; and
- Changing the situation is an objective of the territory's inhabitants (Makarenko, 2012).

The last point is still relevant today. The successful introduction of information technologies would make a big difference in developing the Far North and improving the quality of life in these territories.

But, despite all this, stereotypes about the Far North as a resource provider settled hard in people's minds. It was quite normal when resource was the main value in the world, but now, in post-industrial society, priorities have changed. Now human resource is the main value: knowledge, technologies, professional of their field – now these are the things, which cost a lot. So it is not wondering that IT-industry has its own important place in modern world.

In this way, when the direction of the Far North development became definite, we can take a look on IT-industry itself.

Figure 1*. The image shows us which regions and regions of Russia belong to the territory of the Far North. These territories are taken as a basis in this research work.*

Sources of the maps: Russian public organization "Russian Geographical Society" ("RGS") *is a geographic public organization of Russia, founded on August 18, 1845. One of the oldest geographical societies of the world after the Paris (1821), Berlin (1828) and London (1830). Their maps are used in this article.*

Figure 2. *On the map of Russia the Siberian Federal District is marked with red borders, which directly relates to the SFU mission. Inside it, orange colored areas are marked, related to our research work. Territories that partially belong within the Far North will be specifically identified in paragraph 3.*

Figure 3. *Here are the settlements belonging to the SFU mission, where* 1C *projects were implemented. Cities are noted not only in areas that are completely controlled by the Far North, but also those that are partly its territories.*

Source: *http://narfu.ru/aan/news.php?ELEMENT_ID=233183*

IT Industry: Concepts, Current Trends and Prospects of Development

Assessment of Costs of Development of IT projects & Assessment Methods

The discussion of such a thing as an 'IT Industry' should begin with the concept of information technology in general. Information technology is understood as processes, methods of collection, processing, distribution and storage of information, as well as the creation and operation of computers that perform these functions. So, in a broad sense, IT encompasses all processes that take place with information.

Despite the fact that information technologies, in their essence, are not limited to computer technologies alone, most people rightly associate them with this sphere, since the emergence of computers has brought information work to an entirely new level, not comparable even with the influence that, in its time, the appearance of television had on IT.

Nowadays, the IT industry is primarily concerned with the creation and further development of information systems, through which IT can more effectively cope with its main task – reducing time, labour, energy and material costs in all spheres of human activity.

Today, the total volume of the global IT market exceeds two trillion USD. Moreover, the most dynamically developing segment in the global IT market is software development, whose annual growth is about 6%, and this trend has been maintained for the past several years. More than half of this segment is composed of applications of various categories and the remaining parts are formed by development tools and system software. The fastest development is found in the category of applications for collaborative work, specifically for intra-company social networks and for sharing files, with their volume increasing by more than 20% annually. The second fastest growing developments are in applications in the categories of solutions for database management and analytics, with annual growth of about 8%. High demand is also maintained for categories of enterprise resource management applications, customer relations and security assurance (IDC, 2014).

In addition, a special place in the further strategic development of information is occupied by so-called cloud technologies, in the analysis of large amounts of data, and in the integration of mobile devices and social networking technologies in a corporate environment. The combination of these technologies is combined into the so-called 'Third Platform,' the development of which promises a significant change in business models in most industries in the next few years.

The concept of so-called 'platforms' should be discussed in more detail. These terms denote the stages of development of the IT industry market:

- **First Platform:** formed by a combination of mainframes and terminals, together with a set of applications and users.
- **Second Platform:** formed by a set of concepts familiar to any modern user, such as the Personal Computer and the Internet. Among the more professional terms there are: client-server architecture, LAN and others. The number of applications is measured in the hundreds of thousands, users – in the millions.
- **Third Platform:** formed by a huge, extremely fast growing number of different mobile devices, with a constant connection to the Internet, widespread social networks and cloud infrastructure. The number of users is already measured in the billions, and they, thanks to the already mentioned clouds, can use a huge number of applications, services and content in general. Moreover, due to the simplicity and convenience of working with cloud services, and also due to the high activity of users in social networks, the amount of content in the network is growing very rapidly.

In this way, we can conclude that in the near future in the IT industry will be further developed cloud services, social technologies and mobile devices.

In the process of software development, the problem of estimating the material and time costs for the successful completion of IT projects remains extremely urgent, since it is clearly logical that the customer needs to know the exact figures of a project's cost even before the project is ready. Moreover, this problem extends not only to the initial stage of choosing the most optimal projects in terms of costs, but also to later stages of adjusting the decisions that have already been taken. Of course, carrying out such assessments is absolutely impossible without the use of modern economic and mathematical methods.

However, most of the currently available evaluation methods are aimed at business projects, and they do not take into account a number of certain features of the IT sphere related to the availability of special characteristics in projects in this area that require the involvement of deeper and more sophisticated methods of research. Among the existing methods of evaluation in the IT field, the most effective are the so-called parametric estimates that use the relationships between various project-related data and parameters in order to obtain a specific estimate by means of some mathematical formula as a result. In addition, there is the so-called **COCOMO model (COnstructive COst MOdel)** – an algorithmic model for estimating the cost of software development, which uses collected data on projects being developed that will later be used in calculation formulas of this model (Sheta, 2006: 118-123).

COCOMO is divided into three levels, depending on the detailing and speed of making an estimate of the project's cost:

- **COCOMO Model 1: Basic** – using this type of model, you can perform a quick cost estimate, which, however, will not take into account many factors, such as: hardware limitations, staff experience, using of modern techniques and development tools;
- **COCOMO Model 2: Intermediate** – according to this model, the calculation of the cost estimate is a function that depends on the software's size, as well as on a number of so-called "cost factors", including subjective assessments of product, project, personnel and hardware characteristics;
- **COCOMO Model 3: Advanced/Detailed** – this, the most developed model, is a modified COCOMO of the middle level, which takes into account the same characteristics, but with the evaluation of the influence of each of them on the stages of software development.

Among the methods related to parametrics, the following are distinguished:

The Use Case Point (UCP) (Clemmons, 2006: 18-23) method, based on the using of examples from the so-called **Unified Modeling Language (UML)** (Booch, 2005). The general formula is:

$UCP = (UUCW + UAW) \times TCF \times ECF$, where:

- $UUCW$ - Uncorrected use case weight;
- UAW - Uncorrected artist weight;
- TCF - Technical complexity factor;
- ECF – Environmental complexity factor.

The three-point estimation method, a modified method of **PERT (Program Evaluation and Review Technique)** (Cottrell, 1999: 16-22), that removes the uncertainty of the estimate by this method. The general formula is:

$E = \frac{(a+4m+b)}{6}$, where:

- a - Optimistic scenario for the best case;
- b - Pessimistic scenario for the worst case;
- m - The most likely scenario.

The Function Points (FP) method, (Abran, Robillard, 1996: 895-910; Behrens, 1983: 648-652; Evdokimov, Makeev, Koktashev, 2017: 141-146) which we believe is best for estimating the cost of IT projects, since it is independent of the programming language and can be done at any stage of the project development.

This method uses a number of parameters, the numerical values of which determine the amount of labour and are the standards adopted at the enterprise. These numerical values depend on the following factors:

- Developers qualification;
- Development tools used;
- Experience accumulated by the enterprise in software production.

In this method, it is necessary to justify the numerical values of the five corresponding characteristics:

1. **External Input** – Elementary process that moves data from the external environment to the application. Data can come from the input screen or from another application. Data can be used to update internal logical files. Data can contain both management and business information.
2. **External Output** – Elementary process that moves the data computed in the application to the external environment. In addition, internal logical files can be updated in this process. Conclusions mean reports, screens, printouts, error messages, or output files sent to other applications. Reports and files are created based on internal logical files and external interface files.
3. **External Inquiry** – Elementary process that works with both input and output data. Its result is data returned from internal logical files and external interface files;
4. **Internal Logical File** – A user-defined group of logically linked data that is located inside the application and is serviced through external inputs;
5. **External Interface File** – A user-defined group of logically linked data that is located and supported by another application. The external file of this application is an internal logical file in another application.

All outputs, inputs and requests refer to the so-called **transaction** category**.** A **transaction** is an elementary process that is distinguished by the user and moves data between the external environment and the software application.

After calculating these parameters, each of them is assigned a complexity – low, average or high rank, and then a numerical rating of rank.

How exactly the ranking is formed is shown in the following tables 1 to 5.

Table 1. Rank and assessment of external inputs complexity

Amount of links to files	**Amount of data elements**		
	1 – 4	5 - 15	> 15
0 - 1	Low (3)	Low (3)	Average (4)
2	Low (3)	Average (4)	High (6)
> 2	Average (4)	High (6)	High (6)

Table 2. Rank and assessment of external output complexity

Amount of links to files	**Amount of data elements**		
	1 – 4	5 - 19	> 19
0 - 1	Low (4)	Low (4)	Average (5)
2 - 3	Low (4)	Average (5)	High (7)
> 3	Average (5)	High (7)	High (7)

Table 3. Rank and assessment of external inquiry complexity.

Amount of links to files	**Amount of data elements**		
	1 – 4	5 - 19	> 19
0 - 1	Low (3)	Low (3)	Average (4)
2 - 3	Low (3)	Average (4)	High (6)
> 3	Average (4)	High (6)	High (6)

Table 4. Rank and assessment of internal logical file complexity

Amount of links to files	**Amount of data elements**		
	1 - 19	20 - 50	> 50
0 - 1	Low (7)	Low (7)	Average (10)
2 - 5	Low (7)	Average (10)	High (15)
> 5	Average (10)	High (15)	High (15)

Table 5. Rank and assessment of external interface file complexity

Amount of links to files	**Amount of data elements**		
	1 - 19	20 - 50	> 50
0 - 1	Low (5)	Low (5)	Average (7)
2 - 5	Low (5)	Average (7)	High (10)
> 5	Average (7)	High (10)	High (10)

In addition, it should be noted that if an external reference uses a file reference both at the input and output stages, it is counted only once. The same rule applies to data elements.

After collecting all the informational characteristics of the product, you need to proceed to the calculation of metrics – Function Points (FP).

The initial data for the calculation from Tables 1 to 5 are summarized in Table 6.

Table 6. Data for the calculation of FP-metrics.

Characteristics name	**Rank, complexity, amount**			
	Low	Average	High	Total
External input	? * 3 =	? * 4 =	? * 6 =	= ?
External output	? * 4 =	? * 5 =	? * 7 =	= ?
External Inquiry	? * 3 =	? * 4 =	? * 6 =	= ?
Internal Logical File	? * 7 =	? * 10 =	? * 15 =	= ?
External Interface File	? * 5 =	? * 7 =	? * 10 =	= ?

Total amount (further – TA) = ?

Where:

- ? – quantitative characteristics' numerical values of each species for all levels of complexity;
- Numbers – complexity assessment.

The quantitative values of the characteristics (?) are multiplied by numerical estimates of the complexity. The values obtained in each row are summed, giving a full value for this characteristic. These total values are summed vertically, forming the total amount (TA).

Finally, you can start calculating the function pointers (FP) by the following formula:

$FP = \mathrm{TA} \times (0.65 + 0.01 \times \sum_{i=1}^{14} F_i)$, where:

$F_i -$ The complexity adjustment factor ($i = \overline{1, 14}$), taking values from 0 to 5, according to the following expressions:

- 0 – No influence
- 1 – Incidental
- 2 – Moderate
- 3 – Average
- 4 – Significant
- 5 – Essential

The values are selected as a result of the answers to the questions in Table 7, which characterize the system parameters of the application:

Table 7. Defining system application parameters:

№	**System parameter**	**Description**
1	Data transmission	How much data is required to transfer or exchange information with the application or system?
2	Distributed data processing	How are distributed data and processing functions handled?
3	Performance	Does the user need to fix the response time or performance?
4	Prevalence of the used configuration	How prevalent is the current hardware platform on which the application will run?
5	Transaction speed	How often are transactions performed? (Every day, every week, every month)?
6	Online data entry	What percentage of information need to be entered online?
7	Efficiency of the end user	The application was designed to ensure the effective operation of the end user?
8	Operational update	How many internal files are updated in online transaction?
9	Complexity of processing	Does the application perform intensive logical or mathematical processing?
10	Reusability	The application was designed to meet the requirements of one or many users?
11	Easy installation	How difficult is the conversion and installation of the application?

12	Easy operation	How effective and / or automated are the startup, backup and recovery procedures?
13	A variety of accommodation conditions	Was it possible to design, develop and support the possibility of installing the application in different places for different organizations?
14	Simplicity of changes	Was the simplicity of the change been designed, developed and supported in the application?

We can, using this FP value, generate metrics for some estimates. For example:

- $\text{Performance} = \frac{\text{FP}}{\text{Expenses}}$;
- $\text{Quality} = \frac{\text{Amount of errors}}{\text{FP}}$;
- $\text{Unit cost} = \frac{\text{Cost}}{\text{FP}}$;
- $\text{Documentation} = \frac{\text{Amount of document's pages}}{\text{FP}}$.

To calculate the labour intensity and development time characteristics, you initially need to calculate the expected number of source code lines (SLOC). SLOC is one of the most important attributes of software. This is not only a key indicator of the cost and time of software development, but also the base unit for obtaining other indicators for assessing the status and quality of software. The development of standards for the calculation of SLOC is carried out by the organizations SEI and IEEE. In accordance with them, logical lines of code are used for calculation, and not physical ones, since the number of physical lines varies greatly depending on the style of writing the code. The standards also specify how to read logical strings for each programming language.

To calculate the estimated number of lines of source code, we use a table with empirically derived values based on existing projects and based on determining the number of operators per functional point, and showing the complexity of development for different programming languages.

Based on the previously selected programming languages, the ratio of the number of operators to a functional point for various projects in the previously described business areas was evaluated. Table 8 shows the average, median, minimum and maximum values of this ratio, ordered by the popularity of programming languages.

Table 8. Recalculation of FP-estimates to LOC-estimates:

Programming languages	Average	Median	Minimum	Maximum
Java	53	53	14	134
Python	24	15	15	60
J2EE	46	49	15	67
C#	54	59	29	70
ASP	51	54	15	69
ABAP	28	18	16	60
JavaScript	47	53	31	63
C++	50	53	25	80
HTML	34	40	14	48
SQL	37	35	13	60

This table is based on the classification of languages into high-level and low-level, proposed by C. Jones, who classified programming languages by the number of operators that they need to implement a single functional point.

Further, by multiplying the functional size of the software and the average value from the table for the selected development language, you can get the predicted value of the project source code. And using SLOC you can calculate any required LOC-estimate, for example, using the COCOMO model.

$$\text{LOC} = a \cdot \text{UFP} + b$$

Parameters 'a' and 'b'' can be obtained using linear regression based on completed projects available data (Evdokimov, Makeev, Koktashev, 2017, p. 141, table 2).

To calculate the nominal labour intensity (without taking into account the coefficients of labour costs, cost factors and complexity), the following formula is used, obtained with the help of the COCOMO model:

$T = N_1 \times KSLOC^{N_2}$, where:

- $N_1, N_2 -$ are determined according to table 9;
- $KSLOC - \frac{LOC}{1000}$ of lines.

Table 9. Coefficients N_1, N_2:

Software type	N1	N2
Common	3,2	1,05
Semi-independent	3,0	1,12
Embedded	2,8	1,20

- **Common software** − software of small volume (not more than 50 KSLOC), developed by a relatively small group of experienced specialists in stable conditions;
- **Semi-independent software** − software of medium size (no more than 300 KSLOC), developed by a group of specialists of medium qualification;
- **Embedded software** − software with strict limitations (air ticket reservation system, air traffic control system, etc.).

The development time is calculated by the formula:

$t = 2.5 \times T^{N_3}$, where:

N_3 – determined according to Table 10.

Table 10. Coefficient N_3:

Software type	N3
Common	0,38
Semi-independent	0,35
Embedded	0,32

One serious issue of every method of calculating labour costs is that you can't show a specific example if you are not part of the development team. Despite that we still can make a summary about which programming language is optimal, if our target is to minimize labour costs of the IT-project.

As mentioned earlier, FP-method is based on the number of operators the language needs to implement a single functional point. For *1C* programming language, this number is noticeably lower than for other languages (about 10 operators for 1 functional point). This fact makes this language a priority in development for corresponding subject areas.

It also shows that the Function Point method allows us to evaluate the labour costs even if we are not involved in its development, but its accuracy makes it applicable even on later development stages, whereas, for example, Use-Case method, despite its accuracy on early development stages, doesn't show the real cost situation in later stages.

Current Condition of the Information Community of the Arctic: Features of Calculating Labour Costs & *1C* as a Leader in IT project development

Returning to the problems of working in the Far North, it is important to account for the impact of the IT industry. It is not a secret that things such as navigation systems, weather stations, and means of communication, which are taken for granted nowadays, are all IT industry products to a certain degree. So it would be clever to say that all these things are making living conditions in the Far North easier than it would be without those technologies. In almost every aspect of life the impact of software engineering is evident.

Before looking at examples of IT development in the Far North, including using and distributing IT products, we should identify some features of working in these territories, especially in Russia.

Working in Far North regions is associated with many risks: weather conditions which can cause injury and affect health; the loss of high-cost equipment; and weak and expensive infrastructure, for both governments and private companies. Hence, this often means that nobody wants to come North just for the sake of improving the region, which is so distant.

For today, the financial problems are partly solved. According to the 146th and 148th Articles of the Labour Code of the Russian Federation, the remuneration of labour for workers is increased when they are engaged in working in areas that are seen as potentially harmful and/or dangerous. The same rules are applied for workers employed in jobs in areas with special climatic conditions.

Here we should also notice that IT development itself is not the cause of resource exploitation. In fact, as we stated previously, the high costs of IT in northern conditions are the human resources that are expensive. In IT up to 60% of the expenses are assigned to the salaries of specialists, meaning that it can be quite expensive for employers, especially in the Far North, where there are salary implications.

Imports are almost always more expensive than domestic production – this is one of the main aspects of developing economy. Software development is not an exception here. In every country we can find specialists who can develop and improve the IT industry. So the question is not about the presence of domestic software development but about its quality.

In Russia *1C* is a leader in software supply. Their solutions, based on '*1C: Enterprise v. 8.3*,' are very attractive for many public and private companies in Russia because of their cost and quality, which is comparable with software developed abroad.

In the Far North regions their solutions have taken root as well. For example, the Russian Telephone Company has bought *1C: Enterprise* software packages for their branches in Tomsk, Lesosibirsk and Eniseisk. It helped them to automate a lot of paperwork and bring their productivity to a new level. Furthermore, they also got qualified support from the distributor, which would be less effective if software had been bought abroad.

The table of introduced software solutions by *1C* shows us that the average number of operators for a single functional point is about 50. It means that using *1C* technologies may be a lot cheaper than using other development instruments (Evdokimov, Makeev, Koktashev, 2017: 14, *table 2*).

The leadership of solutions from *1C* explains itself not only by import costs, but also, as we mentioned in the previous part, by its features, which have an impact on the calculation by using one of the most accurate methods of labour costs calculating – Function Points –, which can be used even for primary estimating of labour costs. This fact shows that by improving our national software we can support our IT industry without using expensive software and technologies from abroad.

While the perspective of relatively inexpensive national software can help improve the national IT industry, this increases the possibility of expanding the information society in the Far North, for successful cooperation between scientists, IT developers and entrepreneurs, who might be ready to make the Far North a better place for working and living.

References

Abran A., Robillard P. N. Function points analysis: An empirical study of its measurement processes //IEEE Transactions on Software Engineering. – 1996. – T. 22. – №. 12. – C. 895-910;

Basili V. R. et al. Understanding the high-performance-computing community: A software engineer's perspective //IEEE software. – 2008. – T. 25. – №. 4. – C. 29;

Behrens C. A. Measuring the productivity of computer systems development activities with function points //IEEE Transactions on Software Engineering. – 1983. – №. 6. – C. 648-652;

Booch, G. (2005). *The unified modeling language user guide*. Pearson Education India.

Byers, Michael. (2013). International law and the Arctic. Vol. 103. Cambridge University Press.

Clemmons, R. K. (2006). Project estimation with use case points. *The Journal of Defense Software Engineering*, 18-22.

Cottrell, W. D. (1999). Simplified program evaluation and review technique (PERT). *Journal of construction Engineering and Management*, *125*(1), 16-22.

Cvecinskij A.S., Tuzhilkin V.S., Voroncov A.A. Sovremennoe sostojanie razrabotok specializirovannyh informacionno-analiticheskih sistem dlja obespechenija osvoenija prirodnyh resursov shel'fa morej Rossii i perspektivy ih ispol'zovanija [Modern condition of infoanalisys systems for procuring of developing of natural resources of Russian sea shelf and perspectives of their using] // Trudy Gosudarstvennogo okeanograficheskogo instituta. 2008. № 211. S. 369-383;

Doble M. J. et al. Through-ice AUV deployment: Operational and technical experience from two seasons of Arctic fieldwork //Cold Regions Science and Technology. – 2009. – T. 56. – №. 2. – C. 90-97;

Evdokimov I.V., Bojarchuk N.Ja. Osobennosti strategicheskogo planirovanija razvitija regionov severa Vostochnoj Sibiri [Features of strategic planning of North East Syberia development] // Cennosti i interesy sovremennogo obshhestva materialy Mezhdunarodnoj nauchno-prakticheskoj konferencii. 2013. S. 268-271;

Evdokimov I.V., Makeev V.V., Koktashev V.V. Ekonomicheskoe obosnovanie effektivnosti IT-proektov v regione Krajnego Severa na osnove metoda Function Points [Economic reasoning of effectivity of IT-projects in Far North based on Function Points methodology] // Mezhdunarodnyj zhurnal gumanitarnyh i estestvennyh nauk. 2017. T. 2. № 3. S. 141-146;

Evdokimov I.V., Mikhalev A.S., Kitskalov A. E., Yashevsky D.E. (2017) Osobennosti primenenija metoda Use-Case Points v jekonomike programmnoj inzhenerii s otkrytym ishodnym kodom [Features of application of the Use-Case Points method in economy of program engineering with an open source code.]

Forbes Global 2000 List (2014). Retrieved from, https://www.forbes.com/sites/liyanchen/2014/05/07/the-worlds-largest-companies-china-takes-over-the-top-three-spots/#2faa7bb66f61.

Gao G. X. et al. (2012). Breaking the Ice: Navigation in the Arctic //Global Navigation Satellite Systems: Report of a Joint Workshop of the National Academy of Engineering and the Chinese Academy of Engineering. – National Academies Press. – C. 229.

IDC, IDC CEE Black Book, Q2 2014.

IDC, IDC Worldwide Black Book, Q2 2014.

IDC, Russia Cloud Services Market 2014-2018 Forecast and 2013 Analysis, 2014.

IDC, Russia IT Services Market 2014-2018 Forecast and 2013 Analysis, 2014.

IDC, Russia Vertical Markets 2013-2018 IT Spending Forecast, 2014.

IDC, Worldwide and U.S. Mobile Applications Download and Revenue Forecast; Worldwide Quarterly Mobile Phone Tracker, 2014.

IDC, Worldwide Integrated Systems 2014-2018 Forecast; State of the Market and Outlook, 2014.

IFPUG: Function Point Counting Practices Manual, Release 4.1.1.

Kemerer, Chris F. (1993). Reliability of function points measurement: a field experiment. Communications of the ACM. 36(2): 85-97;

Kofinas G. et al. (2005). A research plan for the study of rapid change, resilience, and vulnerability in social-ecological systems of the Arctic //The Common Property Resource Digest. – T. 73. – C. 1-10;

Low, Graham C., & D. Ross Jeffery (1990). Function points in the estimation and evaluation of the software process. IEEE Transactions on Software Engineering 16(1): 64-71.

Makarenko V. P. (2012). Raspad imperij i problema kolonializma». [The collapse of empires and the problem of colonialism] // Politicheskaja konceptologija: Zhurnal metadisciplinarnyh issledovanij №1.

Matson, Jack E., B.E. Barrett, & J. M. Mellichamp. (1994). Software development cost estimation using function points." IEEE Transactions on Software Engineering. 20(4): 275-287.

R. S. Pressman (2000) Software Engineering: *A Practitioner's Approach.* McGraw-Hill.

Rajkov A.N. (2015). Informacionnye tehnologii na severe i v Arktike Rossii [Information technologies in the north and the Russian Arctic] // Mezhotraslevaja informacionnaja sluzhba. 3 (172): S. 23-28;

Sheta, Alaa F. (2006). Estimation of the COCOMO model parameters using genetic algorithms for NASA software projects. *Journal of Computer Science.* 2(2): 118-123;

Washburn A. L., & G. Weller. (1986). Arctic research in the national interest. *Science*. 233(4764): 633-639.

Worldwide Big Data Technology and Services 2014-2018 Forecast (2014). IDC #250458.