ISSN No:-2456-2165

Analysis of Software Performance Enhancement and Development of Algorithm

Shafagat Mahmudova
Institute of Information Technology of ANAS
Baku, Azerbaijan

Abstract:- **The article provides information about software performance. The software performance is analyzed and problems are explored. Mathematical and engineering approaches and models for solving performance issues are highlighted. The ways to increase software performance are explained. Importance of considering a number of factors in this area is emphasized. Performance of employees in enterprises is compared and an algorithm is developed for its enhancement. The developed algorithm will increase the performance of the enterprise software. Experiments are conducted with the developed algorithm and the proximity of the performance of the enterprises is identified and illustrated in the bar chart. A list of the best software to improve the performance is compiled and analyzed. The issue of software optimization is highlighted. Information on the requirements to the software optimization projects is provided. The article also presents the ways to increase the performance of programmers and offer supportive recommendations in this area. A conceptual model of software performance is developed.**

Keywords:- *Software, Productivity, Optimization, Increase Productivity, Requirements.*

I. INTRODUCTION

In the modern era, information society is increasing rapidly. Computers affect all processes in society, including scientific research, economy, and generally change the way people operate and penetrate new areas of practice. The study of new information technologies and their application in different areas lead to the creation and development of modern systems and programming languages [1]. The main purpose of automation is to save the civilization from inertia and to free human from deadlines set for the execution of tasks. This is one of the main trends in the evolution of computer technology.

When a programmer has a solid knowledge of a programming language used for automation issues, he/she can easily manage on the other knowledge. Moreover, a programmer understands the structure of the program more easily due to the knowledge gained. It should also be noted that the study of computer science, which has been a field of mathematics for a long time, is easy at least due to the fact that programming allows the program to be applied to computers for solving specific problems. Here, the software performance is one of the key issues.

Software products are constantly improving: new features are added, user interface changes, etc. Software performance is an important aspect in developing any software product.

Performance: ability to produce a certain number of products. In other words, it is an ability to release a certain amount of product.

The relevance of this issue is explained by constantly increasing difficulty and importance of software tools. Performance is particularly important in the following cases [2]:

- In engineering and scientific studies, where complex and long-term calculations are performed, and processing time in cluster systems is expensive and limited;
- In web applications, thus, a page generation time is critical to user and directly depends on power volume of server;
- In software products used and so on.

A precise analysis of software performance can be important in reducing the cost of maintenance and equipment.

The concept of performance is either software performance or reactivity:

- performance - amount of information processed by the system within a time unit;
- reactivity - interval between data input to system and generation of relevant data output.

In other words, performance is a capability of software product to be less dependent on the resources of device: processing time of processor, transmission capability of communication channels of capacity occupied in internal and external memory, and so on [3]. Many manufacturers are seriously engaged in performance issues and focus on optimization in this area more.

There is a common tendency that the performance of the tasks available in industry is not evaluated appropriately: "Make it right before speeding up" and "computer model of following year will be 50% faster anyway."

The factors affecting software performance are [4]:

- computer memory volume;
- hard drive access speed;
- maximum frequency of work and processor overload;
- software upgrade and so forth.

- What is the difference between performance and effectiveness?

Effectiveness is an execution of right actions. In other words, effectiveness is an accumulation of knowledge, tools, and set of techniques, thus it allows for more effective work. Performance is an accuracy of the actions performed. Performance measures the effectiveness of work. To evaluate performance of software and information systems, it should be analyzed. In this regard, special approaches and models are available for it.

II. ANALYSIS AND PROBLEMS OF PERFORMANCE OF INFORMATION SYSTEMS

Information System (IS) - is a data processing system working together with the organization resources, such as people, technical facilities and financial resources. Thus, it provides and distributes data for the system [5]. A large number of mathematical and engineering approaches and models of performance analysis are available. They can be structured into three fields:

- analytical modeling;
- imitation modeling;
- experimental analysis of performance.

The first two fields are based on the development of mathematical models, though its application in large-scale distributed systems is very complicated. This is due to the complexity of developing mathematical models for real systems. It is not easy to get reliable values for multiple parameters of all these models. In practice, the third group of approaches is employed more: existing systems or their prototypes are explored by using the measure of performance indicators in accordance with the response to systems controlled by a variety of specialized software tools.

Information systems are mainly used for:

- reduction of enterprise costs;
- increase in labor performance;
- increase in competitiveness;
- quick delivery of order services;
- acceleration of standardization and production process;
- reduction of warehouse resources;
- standardization of personal information, etc.

To this end, computer systems of survey and regulatory information are used in organizations. These systems enable managers and employees to perform document circulation in a few minutes, which previously required a few weeks.

However, some problems arising from the introduction of the system have been well-studied and formulated having effective solution methodologies.

Preliminary study of these problems and preparations greatly simplify the process of application and increase the effectiveness of further utilization of the system.

As noted, performance is one of the main characteristics of software. Inefficiency of users' work can lead to lower performance and result in excessive costs and loss of product, income and customers.

Apparently, most serious problems related to software performance occur at initial stages of processing, especially when it is not properly estimated in scheduling phase.

The main application problems of information systems are as follows:

- Lack of management in enterprise;
- Need for partial or complete reorganization of entity's structure;
- Need for technology change for data processing and business management principles;
- Resistance of enterprise employees;
- and so forth.

Following failures can be encountered when applying information technologies:

- System is incompatible with the needs of organization;
- Work is impossible to be performed in a specific time, since the application requires more time;
- Application requires more cost than scheduling.

Given all this, new methods should be developed to increase software performance, which requires serious implementations.

III. METHODS FOR INCREASING SOFTWARE PERFORMANCE

There are three ways to increase software performance:

- Using additional programs to increase software performance;
- Using software capabilities to increase its performance;
- Increasing programmers' performance to increase software performance.

To achieve high performance in software area, the following factors need to be considered:

- Reducing the length of tasks;
- Reducing the volume of tasks;
- Reducing the number of tasks.

There are three ways to achieve this:

- Faster execution of tasks;
- Parallelize tasks;
- Exclusion of any task, that is, manages on without it.

Key techniques for acceleration of tasks:

- Cashing;
- Initial calculations;
- Initial initialization;
- Package operations.

There are various ways to increase software performance.

Since 2013, there have been several well-known standards. They are used for software size measurement based on functional points:

- COSMIC: ISO/IEC 19761:2011 Software engineering. A functional size measurement method;
- FiSMA: ISO/IEC 29881:2010 Information technology – Systems and software engineering – FiSMA 1.1 functional size measurement method;
- IFPUG: ISO/IEC 20926:2009 Software and systems engineering – Software measurement – IFPUG functional size measurement method;
- Mark-II: ISO/IEC 20968:2002 Software engineering – Ml II Function Point Analysis – Counting Practices Manual;
- NESMA: ISO/IEC 24570:2005 Software engineering – NESMA function size measurement method version 2.1 – Definitions and counting guidelines for the application of Function Point Analysis.

Function Points is the most popular and widely spread method. It was developed by American scientist Alan Albrecht in 1979 (IBM). This method identifies the users' functional requirements for software, and each of them is referred to one of the five types: output, queries, input, internal files, and external interfaces. Once the function is identified and categorized according to its type, it is evaluated for complexity, and several functional points are defined. Each of user's functional requirements is compared to end-user's business function. This difference is important because it is capable to simplify the functions measured at the functional points based on the tendency in user's demand. It can also enable the concealment of internal functions (e.g. algorithms), the implementation of which require resources.

Function Points is a "measurement unit" used to indicate the volume of business functionality that is provided as a product to a user by the information system. Functional points are used to calculate the functional software size. Value is calculated (in dollars) [6]. Function Points are used to measure software size.

This method is used in the following countries:

- US, in 30% of projects;
- Brazil, in state agreements related to software development;
- In 13,000 projects, and its statistics is available;
- Measures software volume in terms, and presented in an understandable form to customer - display forms, reports and so forth;
- Number of function points is correlated with the number of lines of code;
- Function points encourage a manufacturer to increase the amount of software products in terms of customer rather than increasing the number of lines of code;
- The method is standardized in accordance with ISO/IEC20926: 2009;
- Fast Function Points - reduced the number of objects by 9 times compared to function points, accuracy - for large tasks is +/- 3% with Function Points, and for minor tasks - varies widely;
- Effectiveness of software products manufacturing – complexity of cost.

The introduction of Fast Function Points has increased software performance and improved its quality.

In [7, 8], five methods for increasing the programmer's performance are shown:

- mastering and improving development tools;
- effective file management;
- more monitors;
- automation;
- parallelism.

NTSPEI P-Modeling Framework, reverse semantic traceability, and other methods optimize software development stages, where automation is complicated [9].

Software designed for enhanced performance is applied software for data generation, such as documents, presentations, databases, graphs, digital images, and digital videos [10]. They increase performance [11].

On average, 78% of professionals use certain software to increase performance [12]. In 2010, more software was developed for performance enhancement.

There are a number of ways to enhance software performance. One of these methods first takes any listed company and compares an average value of performance of employees of other entities with an average value of current performance of employees of that entity. Then, performance of the company's software is relatively identified. Accordingly, the necessary measures are taken.

Various methods and algorithms for evaluating performance of employees (programmers) working in companies are available. An algorithm given by the author is explained below.

Assume that there is n number of companies. Companies are denoted by $S_{1,}S_{2,}S_{3,...,}S_n$.

Employees working in the *i*-th company are denoted by$P_{i1,}P_{i2,}P_{i3,...,}P_{im\,,}$. Performance of employees working in the *i*-th company is compared with performance of employees working in other companies, and performance of the companies is identified. For this, performance of employees working in the *i*-th company is denoted by $M_{i1,}M_{i2,}M_{i3,...,}M_{im\,,}$.

To calculate the average performance of employees of the *i*-th company the following formula is used:

$$\overline{M_\iota} = \frac{\sum_{k=1}^{m} M_k}{m}, \text{ i=}\overline{1, n}, \qquad (1)$$

Here, $\overline{M_i}$ is an average value of performance of employees of the *i*-th company.

$\overline{M}$ is an average value of an average value of performance of the rest of the companies, with the exception of the *i*-th company, and calculated with the following formula:

$$\overline{M} = \frac{\sum_{i=1}^{n-1}\overline{M_i}}{n-1}. \qquad (2)$$

$\overline{M_i}$ and $\overline{M}$ are compared. If $\overline{M_i} \geq \overline{M}$, $i = \overline{1,n}$, then performance of company is considered satisfactory. Otherwise it is not considered satisfactory and necessary measures are taken. For example, there are 10 companies and each company has 8 employees. Employees of each company and the average value of performance (in $) of employees of all companies are shown in table 1.

Companies	**1st employee**	**2nd employee**	**3rd employee**	**4th employee**	**5th employee**	**6th employee**	**7th employee**	**8th employee**	**Average value of performance of employees**
1st company	60	70	80	90	100	50	70	80	**75**
2ndcompany	70	80	60	100	50	80	60	50	**68,75**
3rd company	90	80	60	60	50	80	70	50	**67,5**
4th company	60	70	90	100	70	60	40	100	**73,75**
5th company	70	60	80	90	60	70	50	100	**72,5**
6th company	50	80	70	90	60	80	40	90	**70**
7th company	100	70	60	100	50	70	60	50	**70**
8th company	80	70	90	60	70	90	50	70	**72,5**
9th company	70	80	80	100	50	90	50	100	**77,5**
10th company	90	60	70	80	60	90	100	50	**75**
Average value of performance of employees of all companies									**71,94**

Table 1:- Average Value of Performance by Companies

Using the average value of performance of employees the degree of their proximity can also be defined. Euclidean distance was used in this regard. The average value of performance of employees of the random company is denoted by $\overline{M^*}$. Whereas, the average value of performance of employees of other companies is denoted by $\overline{M}_{ik}$.

Denoted Euclidean distance by S_i.

$$S_i = \sqrt{\sum_{k=1}^{n-1}(\overline{M}^* - \overline{M}_{ik})^2}, \ i = \overline{2,n} \qquad (3)$$

Using formula (3), S_i –s are found and arranged in ascending order, and proximity of random company to other companies is determined.

Using formula (3) and Table 1, the value of $S_i - $ s are calculated. The values found by the companies are shown in Table 2.

5th company	9,64
9th company	9,64
4th company	10,25
6th company	11,36
7th company	11,36
8th company	11,36
1st company	13,75
10th company	13,75
2nd company	15,65
3rd company	18,25

Table 2:- Proximity of Values of Companies Performance

This table shows the proximity of values of companies' performance.

A bar chart corresponding to these values is drawn up and shown in fig. 1.

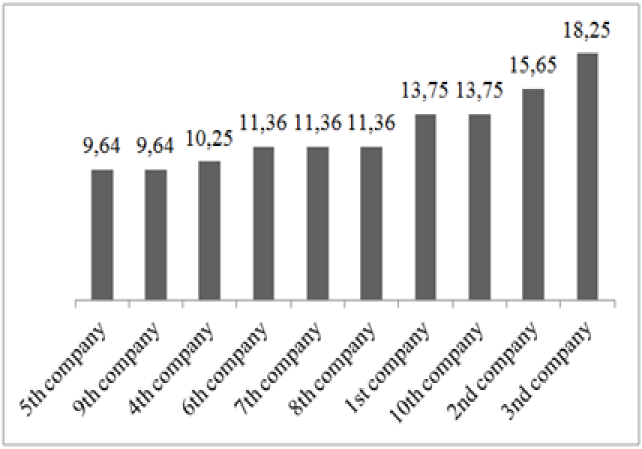
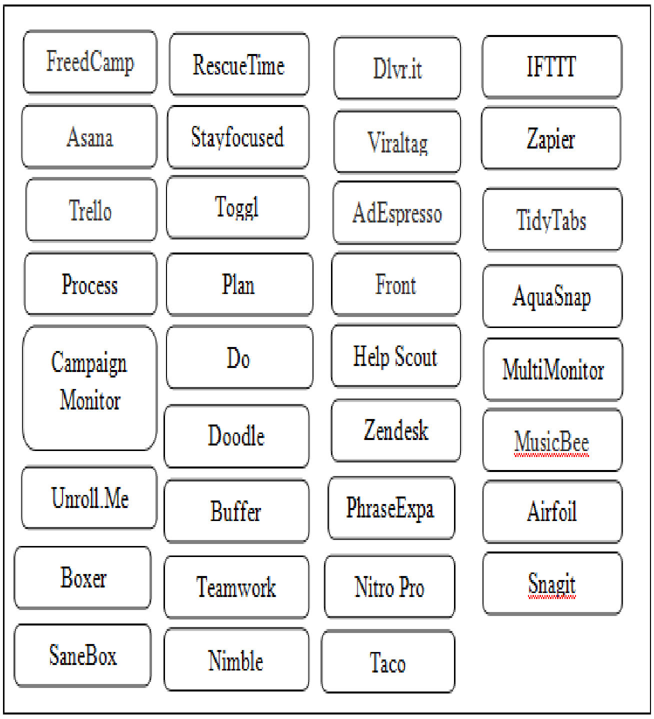


Fig 1:- Bar chart illustrating the proximity of values of companies' performance



FreedCamp	RescueTime	Dlvr.it	IFTTT
Asana	Stayfocused	Viraltag	Zapier
Trello	Toggl	AdEspresso	TidyTabs
Process	Plan	Front	AquaSnap
Campaign Monitor	Do	Help Scout	MultiMonitor
	Doodle	Zendesk	MusicBee
Unroll.Me	Buffer	PhraseExpa	Airfoil
Boxer	Teamwork	Nitro Pro	Snagit
SaneBox	Nimble	Taco	

Fig 2:- A list of best software for performance enhancements

The best 34 software for performance enhancements in 2017 are shown in fig. 2 [13]. A brief description of each of them is given below.

- FreedCamp - a new, comfortable and perfect tool for increasing performance. More than 160,000 people use it. It includes daily registrations and effective reports used to implement more effective and efficient commands. Its functions include job listings, notes, and calendars. It serves to platforms as invoices, project templates, and so forth. It also has an intuitive clear interface making it easy to watch. It is easy to use and convenient for all command types.
- 2. Asana - a multifunctional tool for increasing software performance. It enables to assign emails or tasks to commands and then converts them into executable tasks. The projects can be manages, the representatives can be sent or collaborated through them, and the examination can be executed during their compliance.
- Trello - a perfect tool for managing projects and sending delegates. Each task or project is placed in card so that it can be further arranged according to the panel and the columns. These cards allow team members to make records in accordance with their time.
- Process Street is an ideal tool for creating documentation for entire company. It is used for the repeated procedures of employees. It distributes documents by sections, adjusts them, and monitors the parallel lists on the screen. This is also useful for deployment of new employees, and for adoption of new policies and software.
- Campaign Monitor - a valuable email service with the help of which quality emails can be created. Thus, they can run on any device and verify emails for their optimization.
- Unroll.Me – enables subscription to e-mails and thus prevents time loss. The Rollup function prevents the bulletins to load mailbox and re-organize e-mail. Thus, messages can be read in any order when they are received.
- Boxer - a mobile application that allows re-establishing email on mobile phone. It enables to change the volume of information, adjust the stored responses, and categorize the marks. The application also integrates with Evernote, Drive, and Salesforce, thus the content of email can be sent using other tools.
- SaneBox – records data to e-mails over time and filters insignificant messages and saves them in separate folders. Thus, not only unnecessary e-mails are prevented, but also any difficult decision making is avoided.
- RescueTime - Statistical data is automatically compared and the list of established categories is properly visualized on Rescue Time information panel. It tracks the time spent by the user on different websites and programs. The most productive time and attractive types of pages can be seen here. The events notifications can also be scheduled, when a large amount of time is spent on a page or blocked pages.
- Stayfocused - takes the user's work on computer and allows him/her to spend time on websites in the course of work. The user can track the most productive time and then plan to work in different temporary windows all day long.
- Toggl – If a user does not need a program to control the time, then Toggle can be used. It is a simple mechanism to track time. It enables the user to execute personal tasks and projects. Click the "start" or "stop" button for the project that controls the user. The function of the report is to inform a user about employee habits and advise them to make adjustments to maximize performance. It also deals with issues such as Trello and Asana for quality management.

- Plan - the most recent management tool. It focuses on the user's life and work balance. A user can track whether he/she has a healthy lifestyle or not. It also provides notes, scheduling functions, and works perfectly with Google Calendar.
- Do - is software for meetings. It simplifies the process before and after the whole meeting. It does everything to maximize the meetings of consumer, from scheduling to drawing up results list.
- Doodle - deals with diagrams for large groups. A user sends diagrams in a short time and recipients promptly respond about their delivery.
- Buffer - a tool for managing social networks, it simplifies a joint use of content on different platforms. Here, a series of news (posts) can be ordered and scheduled a few weeks in advance. Buffer also optimizes the content, thus the most important thing is to spend less time on social networks.
- Nimble - is a customer relationship management system for social networks (CRM), thus it filters all network interactions. It constantly updates information providing the most important information about prospects, and there is no need to search for several sites to find them.
- Dlvr.it - a tool for social network automation. It ensures automatic content sharing without the need for journal management.
- Viraltag - a tool for social networks specifically designed for sharing visual effects. Once in social networks, each site has its own dimensions for images it enables to edit the news (messages) for its better visualization on the site. Moreover, it is possible to plan and edit the content of the news (messages) in advance.
- AdEspresso - a tool for increasing performance allowing a user to create ads on Facebook. It enables the automatic optimization of advertisement and even guarantees that a user will only spend a few minutes to create several versions of advertisement, and they will work normal.
- Front - a generic entry for the news (messages) entered. It provides instant access to email, Twitter, Facebook Messenger and SMS.
- Help Scout - enables a user to manage more customers efficiently from a common e-mail address with several mailboxes. It provides collaboration with team members. Scout allows the client to automatically set up the news (messages) without opening it. It saves time and personal news (messages).
- Zendesk - offers functional opportunities of Help Scout, though processes more news. It provides adding phone, video and document to its settings and managing workflow efficiently enabling the user to establish feedback with clients. It also provides a space to keep all data about clients, thus a user do not need another place to store data.
- Teamwork Desk - has unified mailboxes and customer portal. As with Help Scout, user emails do not appear to customers, however they can get access and support through a high-level portal. This portal can be a center for customers, thus they can access documents, estimate user's support and interact with other customers.
- Taco - removes user issues from several applications and places them in a nearby channel. It can be integrated with more than 30 different programs as Trello, GitHub, Basecamp. It frees a user from re-examining the program.
- IFTTT - a platform that allows combining hundreds (357) of programs. To locate various program details on personal platform, Taco sends any information from one program to another. It provides using formula.
- Zapier - helps a user to automate working processes and can be considered analogue to IFTTT, nonetheless this is more complicated. For example, someone sends an e-mail to user and it can be automatically saved in Dropbox, and then notification is sent.
- TidyTabs - If a user is not satisfied with a few windows of programs, then Nurgo TidyTabs is installed from Software. Lightweight utilities organize the desktop allowing putting together multiple windows in as bookmarks, even for unrelated programs.
- AquaSnap - offers many opportunities for further enhancement of desktop and windows processing. For example, it closes windows easily and arranges them side-by-side combining them with each other as magnets.
- MultiMonitorTool - designed to deal with multiple monitors as seen from its title. It is used to turn off or disconnect several monitors, change the default monitor and the position of windows on monitor from the user's graphical interface.
- MusicBee – a perfect music player that supports the reproduction of music from different sources. With the help of different skins, appearance of musical device can be changed and displayed in an efficient way.
- Airfoil - a powerful digital device that works with a variety of devices including AirPlay and Bluetooth device dynamics combined to computers. It is capable to provide audio from many music service devices.
- Nitro Pro - software for reading and editing PDF files. This powerful and fast editor easily modifies PDF documents and is equipped with an OCR integrated mechanism that converts scanned documents into edited documents.
- Snagit - a utility to capture images placed on computer monitor. It is powerful software for screen and video. The program is capable to capture not only desktop or program window, but also screensaver of web page.
- PhraseExpander - a useful utility that collects large amounts of information on computer display. Although it is advertised as a tool for medical professionals, the functions as spell check, phrasing and editing make it perfect.

Today, businesses, including small and medium enterprises (SMEs) are dependent on software to reach their targets and to operate and maintain their business. Although earlier research focused on the quality of software, the correlation between software quality and performance remains a problem as before. [31] touches upon this problem and shows its solutions. In that study the surveys held in Malaysia has been used. The research shows that efficiency, enhancement, functionality, reuse, safety and ease of use are the key factors in terms of management. Based on the results obtained, a software performance model for quality and SMEs has been developed. This model can be used as standard and basic principles in the process of software upgrade and for quality assessment in organizations.

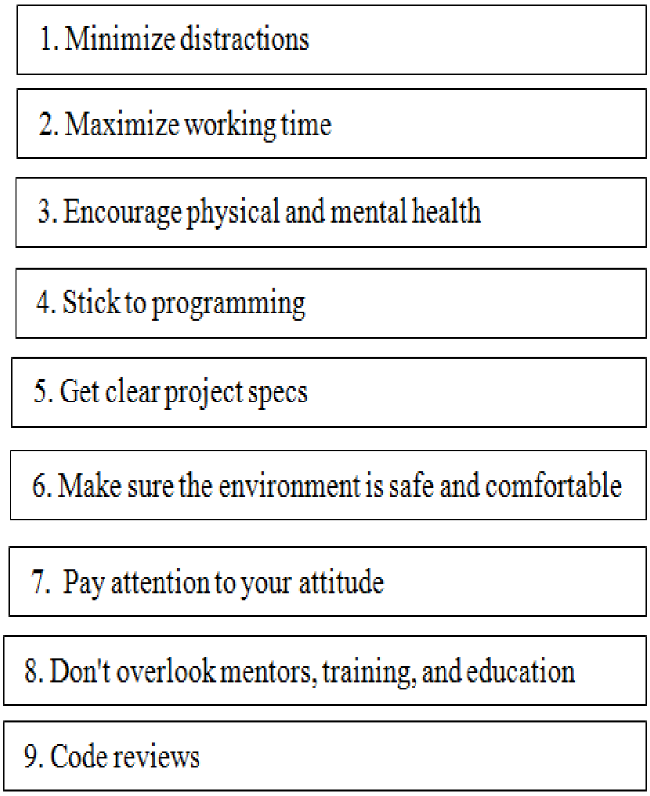
SMEs are the main part of economy. Additionally, some specialists working in small and medium enterprises do not comprehend the benefits of knowledge for software performance quite well. Undeniably, knowledge and skills constitute a part of employees' ability. In [32], analysis is based on available literature and key performance characteristics are identified.

IV. WAYS TO INCREASE PRODUCTIVITY OF PROGRAMMERS

As mentioned above, there are three ways to increase software performance:

- Using additional software to increase performance;
- Increasing performance of software itself;
- Increasing productivity of programmer, which directly affects software performance.

The previous section outlines the use of additional software to increase performance. In this section, the ways to increase productivity of programmer is analyzed. Some publications show 9 ways to increase productivity of programmer [14] (fig. 3):



1. Minimize distractions

2. Maximize working time

3. Encourage physical and mental health

4. Stick to programming

5. Get clear project specs

6. Make sure the environment is safe and comfortable

7. Pay attention to your attitude

8. Don't overlook mentors, training, and education

9. Code reviews

Fig 3:- 9 ways to increase productivity of programmer

Programmer's performance [15]:

- Do not deal with optimization only for optimization (Glyn Williams);
- Use simple abstraction (Glyn Williams);
- Do not write a generalized code complicating the program (Glyn Williams);
- Always reserve a few small projects and use them as needed (Punit Labma);
- Always implement code refactoring (Punit Lamba);
- Learn work environment and programming language tools (Punit Lamba);
- Learn other software codes (Punit Lamba);
- Divide algorithms into smaller parts (Bernard Stoker);
- Survey new technologies (Bernard Stoker);
- Use short program leaps (Con Robertson);
- Learn how to work with team (Eric Rak);
- Automate everything as possible (Peter Shaw).

In the overall process of problem solving, if the program response to user query in a short time, the end user calls it productivity [16].

The number of queries, which is program's ability to perform in a minute (hours/days), serves the number of users (for example, 10,000) for required business.

Checking the performance of program, it is possible to understand whether the requirements for the program are fulfilled, they should be improved or not, and so on.

The number of users is also the access term for test. It is only needed to conduct a test on the possible download required by the customer, and:

- If the system has high performance with the maximum number of users - it does not mean it has no problems;
- It is necessary to know how much reserve the performance has in the system.

Calculated indicators for downloaded test:

- Latency - What is the average latency? Which is the maximal and minimal?
- Bandwidth - how many requests are handled per minute?
- Error rate - how many requests are failed?

The programmers with high productivity use the effective processes in module testing [29]. One of the factors affecting the software performance is how effective the programmers handle the tasks. Programmers take a few steps to handle them. The fulfillment of these steps is called the process of tasks. Although the process of general programs has been studied well, their impact on the process has been remained unstudied. [29, 30] provides detailed information on these issues. In this study, the process of modeling is modeled as a Markov chain, and then the high productivity and low productivity of programmer are studied and decision is made respectively.

V. SOFTWARE PERFORMANCE PROBLEMS

To understand the problems of performance more clearly, they are divided into several groups [17].

First group – includes easily acceptable problems. They are met everywhere. They may include requests optimization, non-optimal algorithm, various indexing of fields, and so forth. Such problems are quickly resolved by the enterprise specialists.

Second group - includes variable and unexpected problems. They are relatively less encountered. These problems occur unexpectedly. For example, failure of programs on server or of system at a certain moment on any node. Generally, this problem can always be. Typically, a particular methodology should be used to address such problems so that they can be detected during multi-user operating mode.

Third group - includes the problems previously known, however difficult to solve. For solution of such problems numerous technologies, such as parallel computing are used.

Reasons for performance reduction problems. Analysis of situation shows that if any system normally operates at least at the beginning, the main reasons for product degradation (deterioration of object’s characteristics) may include the followings:

- Poor quality of control - failure to track the changes in its parameters after exploitation of the information system;
- Insufficient IT infrastructure for rapid growth of company products;
- Failure to check the effects of new functions on performance, etc.

VI. SOFTWARE OPTIMIZATION METHODS

Despite the ever-increasing effectiveness of computers, productivity of human labor is rising deliberately, which is particularly relevant to productivity of programmers. Condition can be improved through INTSPEI P-Modeling Framework, reversible semantic tracing and other methodologies that optimize the process of software development stages [18].

For beginning, the average productivity of labor of employees of a company is compared with the current productivity of labor of employees of other companies.

If the indicators of the company’s performance exceed Fortune Global 500, this is an increase in performance.

Productivity does not only deal with circulation and revenue. Labor is the key point here. Labor is a result and benefit, as it is generated by an employee.

When evaluating the business of software developer, its ecosystem is analyzed. It is impossible to distinguish between productivity of labor in company and its productivity in business-environment.

The key issue facing a researcher is to increase performance. Criteria for optimization function are as follows:

$y = \varphi(x_1, x_2,..., x_n) \rightarrow min$,

thus, y - time of data processing;

$x_1, x_2,..., x_n$ - all parameters (all affecting factors), thus, they can directly or indirectly affect performance;

$x_i \epsilon [a_i, b_i]$ - assigned field of the i-th factor.

The problem of low performance of information systems can be solved by performing a number of inspections and modifications of processes [19].

Increasing the performance of existing systems may avoid the purchase of additional server equipment and save considerable funds to the budget. In this regard, the followings should be implemented:

- troubleshooting the system;
- performance analysis;
- performance inspection;
- performance engineering;
- hardware optimization, etc.

VII. REQUIREMENTS TO THE PROJECTS FOR SOFTWARE OPTIMIZATION

A number of requirements are specified to the project to optimize software. These requirements are as follows.

➢ The first requirement - conducting uninterrupted quality monitoring. If this criterion is not followed, then the source of problem will not be revealed. That is, there was a problem, and now it does not exist.

- Definiteness in the course of analysis is very important. This requirement is important and the maximum number of information should be collected. Thus, it will be possible to determine what exactly is happening in the system.

➢ The second requirement - Step by Step,

- Optimization of projects, unlike other projects, cannot be paralleled. The first period is performed and if the resource is N number of people, then the distribution of problems among them will be complicated. Because the system is a complex organism, people share the first and other problems among themselves, and the problem may disappear even at that time;
- Optimization of all rows does not make sense since, after the first period, their problems may disappear. The complexity of parallel work should be considered.

➢ The third requirement - it should be realized that finding the first reason of problem can take more time rather than its optimization.

- Events should be sequenced in a way leading to the determination of the performance problem. In this case, it is important to find the first problem and it will be the first reason. If the first reason is not found, it means that the time for problem finding has not been used effectively.

- The fourth requirement - substantiation.
- A rule should be taken, so that in all decisions the system is based on current situation. It should be ensured that the system is requested as the type of variables.
- If knowledge base is taken as a basis, the probability of errors becomes high enough.

- The fifth requirement - this is a necessity rather than requirement.

- What is the difference of this requirement?

At the early stage of project optimization, the main objectives should be identified – “Do not deal with optimization only for optimization”.

The growing complexity of all software and its daily use has increased the interest in software analysis. This is mainly about the evaluation of functional features of software systems (their structure, etc.). Software performance analysis has recently become topical. This analysis is aimed at evaluating the behavior of software, i.e. from planning to code, through a thorough analysis of its structure and behavior [20].

A systematic approach is required to performance control throughout the lifecycle of software.

Development of software systems meeting performance objectives is one of the main tasks. Performance is an indicator of software systems, how good the system is, or how appropriate the software components are to requirements, and so on. Latency is the time required to respond to this request. For example, online system may be required to show the result within a half second after a user presses the key. This is the intermediate time for integrated systems to respond to the events needed. Bandwidth of system is referred to the number of requests that can be processed for a certain period of time. For example, a telephone switch may be required to process 100,000 calls per hour [21].

Thus, performance is any characteristic of software product and the user can principally measure the time in front of the computer with a stopwatch. In other words, performance is a software quality indicator [22].

Software performance problems may include:

- Insufficient theoretical bases of methods to increase software performance;
- Contradiction between automation and adaptation of system, thus it's difficult to change it;
- Correlation of events in distributed systems.
- The key approach to analysis of the worst-case occurrence in the program has been used so far. This analysis is divided into two stages [23]:
- program is analyzed to find the worst possible route. This problem is sometimes called route analysis.
- worst route implementation time is found. This problem is temporarily called synchronization.

- Testing software performance - checking the capacity of program to perform during loading period in terms of scalability and vulnerability. Vulnerability is bandwidth capacity of a specific program to achieve predetermined targets, and scalability is the number of types of activities currently being processed. Implementation of this type of test is a key factor for determining the quality of program [24].

Purpose of performance testing. The entire process of software performance testing is implemented to achieve four goals:

- Determination of bandwidth or transaction speed;
- Determination of response time of server;
- Inclusion of functional test scenarios into test scenario to determine render (visualization) time;
- Determination of technical specifications, and their documentation based on test.

Six basic types of software performance testing are distinguished:

- download;
- tension;
- robustness;
- deviation;
- configuration;
- isolation.

In modern times, human activity with computers depends on automated applications more. These applications provide people with plenty daily activities. Programs are therefore of great importance in engineering society. In recent years, many software prediction models have been built to develop high-quality software. Earlier software development methods are also used in developing countries, especially in Pakistan. Using these outdated methods will result in lower incomes of investment. These methods also lead to malfunctions in timely delivery of products [25].

Software lifecycle models in Pakistan are as follows:

- Waterfall;
- Spiral;
- Periodic;
- Agile.

Many software development projects focus on performance problems after functional testing [26]. Early planning of requirements for performance and performance testing can prevent failure. Engineering methods can be integrated into all phases of software lifecycle: from system concept to specifications of requirements, inspection and production. Effectiveness monitoring can be carefully analyzed. Discussion of system architecture covering its development and implementation reduces the risk of its poor quality. Performance modeling can be used to verify architectural and design decisions, including performance testing projection. The results of performance tests enable a user to identify problems early enough.

- Conceptual Model – a set of elements and presentations (ideas) about the objects. It represents goals of projected system and the relationships between terminologies of this or that theory. Figure 4 illustrates a scheme of conceptual model to increase software performance. This scheme visualizes how to increase performance.

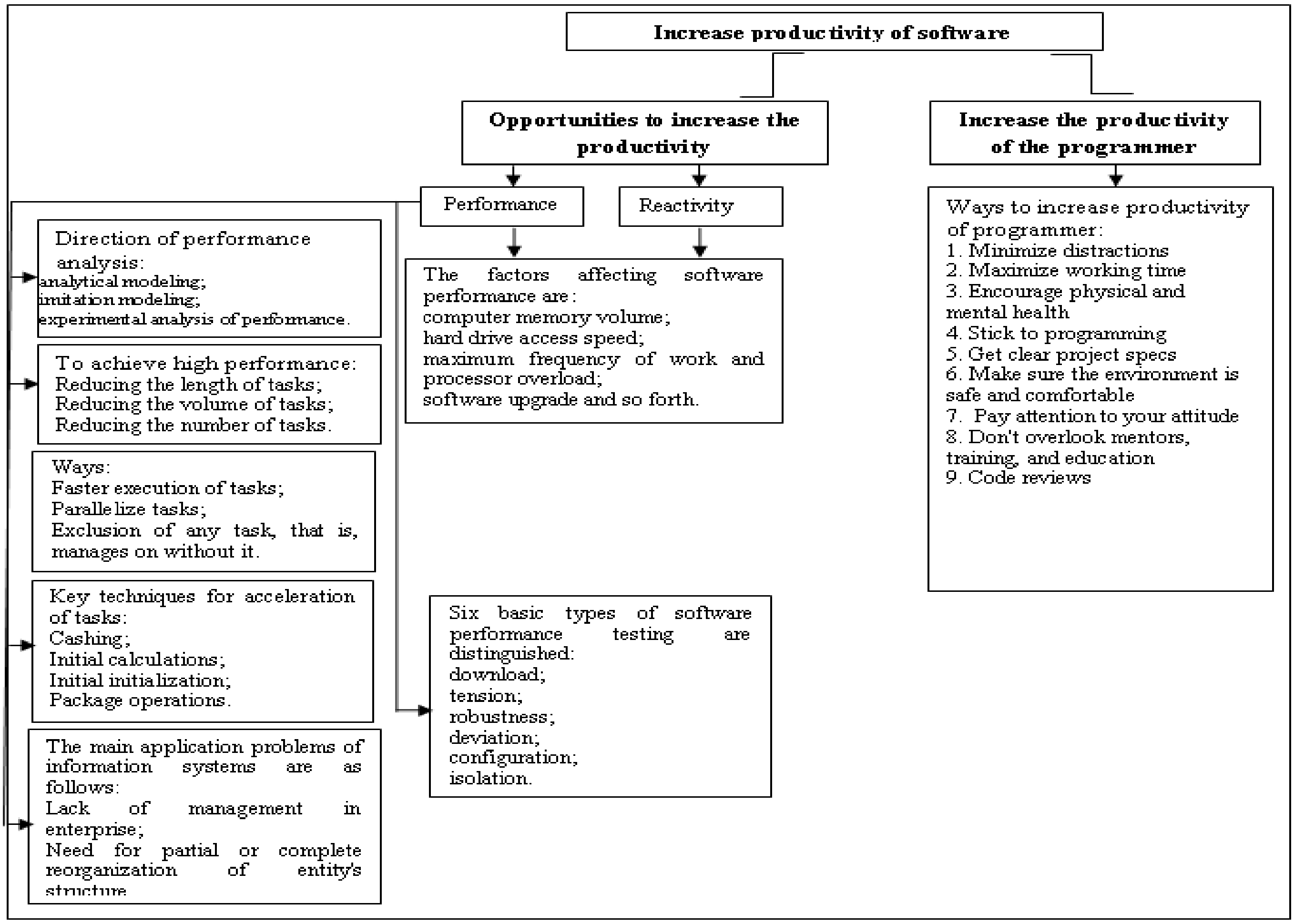

Increase productivity of software
Opportunities to increase the productivity
Increase the productivity of the programmer
Performance
Reactivity
Direction of performance analysis:
analytical modeling;
imitation modeling;
experimental analysis of performance.
To achieve high performance:
Reducing the length of tasks;
Reducing the volume of tasks;
Reducing the number of tasks.
Ways:
Faster execution of tasks;
Parallelize tasks;
Exclusion of any task, that is, manages on without it.
Key techniques for acceleration of tasks:
Cashing;
Initial calculations;
Initial initialization;
Package operations.
The main application problems of information systems are as follows:
Lack of management in enterprise;
Need for partial or complete reorganization of entity's structure
The factors affecting software performance are:
computer memory volume;
hard drive access speed;
maximum frequency of work and processor overload;
software upgrade and so forth.
Six basic types of software performance testing are distinguished:
download;
tension;
robustness;
deviation;
configuration;
isolation.
Ways to increase productivity of programmer:
1. Minimize distractions
2. Maximize working time
3. Encourage physical and mental health
4. Stick to programming
5. Get clear project specs
6. Make sure the environment is safe and comfortable
7. Pay attention to your attitude
8. Don't overlook mentors, training, and education
9. Code reviews

Fig 4:- Scheme of conceptual model to increase software performance

As in other areas of science, many tools are available in programming to improve productivity. Practice does not always have a positive effect. It sometimes even worsens the performance of professionals [28].

VIII. CONCLUSION

The article dealt with software performance and productivity of a programmer, challenges and other issues in this field. High performance of software can be achieved by solving these problems. Other important issues also need to be addressed.

To increase software performance, the followings should be provided: [27]:

- Responding to market trends;
- Quality improvement and stability;
- Monitoring relevant norms and standards;
- Coordination with suppliers;
- Reducing operational costs;
- Managing global operations;
- Transition to higher standards of business management;
- And so forth.

Given all abovementioned, it was concluded that new methods should be developed to maximize software performance, and significant implementations should be done in this area.

REFERENCES

[1]. S.J. Mahmudova and K.K. Mamtiyev, Programming and its development stages, Baku: Informasiya Texnologiyalari, 2011.

[2]. S.A. Dubakov, Information technology performance analysis in the software development process, Tomsk: Politekhnicheskiy Universitet, 2005.

[3]. P.E. Shlender and Y.L. Kokin, Labor Economics (2nd ed.), Moscow: Maqistr, 2002.

[4]. V. Mikhaylov, How productivity is measured by the quality of software development in a system forming financial organization, The Conf. Razrabotka PO, Moscow, 2015, pp. 22-26.

[5]. T.A. Serebryakov and Y.N. Parshin, Analysis of the performance of information systems in the enterprise, Elektronnoye nauchnoye izdaniye, 2014, vol. 5, no. 4, p. 913-921.

[6]. T. Runcie, The Agile Project Manager: Understanding Estimating in Agile PM. Retrieved June 28, 2016, from https://www.mpug.com/articles/the-agile-project-manager-understanding-estimating-in-agile-pm/.

[7]. F. Nil, Productive programmer, "How to make complex simple and impossible possible", St.Petersburg: Simvol-Plyus, 2009.

[8]. G. Bakunov, Five ways to increase productivity for the programmer. Retrieved iyun 7, 2015, from http://itnotesblog.ru/note.php?id=158.

[9]. V. Pavlov, Methods to optimize the process of creating software, Otkrytyye sistemy. SUBD, 2007, vol. 10, p. 1-11.

[10]. D. Ziff, Office Productivity software. PC Magazine Encyclopedia. Retrieved November 30, 2014, from https://en.wikipedia.org/wiki/Productivity_software.

[11]. G. Burning, The Digital Skills Gap in the Workforce. Retrieved Mart 1, 2015, from https://www.burning-glass.com/wp-content/uploads/2015/06/Digital_Skills_Gap.pdf

[12]. R. Sasha, The 27 Best Productivity Software of 2018. Retrieved June 12, 2018, from http://blog.idonethis.com/productivity-software/.

[13]. J. Justin, 10 ways to increase the productivity of your programmers. Retrieved July 30, 2008, from https://www.techrepublic.com/blog/10-things/10-ways-to-increase-the-productivity-of-your-programmers/.

[14]. D. Khokhlova, How to increase the efficiency of development - tips for programmers, Retrieved December 31, 2014, from https://vc.ru/flood/6285-tips-for-devs.

[15]. L. James, Proizvoditelnost programmnogo obespecheniya, Software performance, Retrieved March 31, 2015, from http://www.codenet.ru/webmast/php/perfomance/.

[16]. P. Barketov, Optimization of the performance of information systems, Retrieved March 12, 2012, https://infostart.ru/public/333695/.

[17]. J. Bentley, Writing Efficient Programs, London: Prentice-Hall, 1982.

[18]. D. Knuth, The Art of Computer Programming (3th ed.), Moscow: Williams, 2006.

[19]. C. Vittorio, D.M. Antinisca and I. Paola, What Is Software Performance? Model-Based Software Performance Analysis, 2011, p. 1-7.

[20]. U.S. Connie and G.W. Loyd, Introduction to Software Performance Engineering, London: Addison-Wesley, 2001.

[21]. W.C. Murray, F. Greg and C.P. Dorina, The Future of Software Performance Engineering. Proceedings of the International Conf. on Software Engineering, ISCE 2007, Minneapolis, USA, pp. 171-187.

[22]. W. Wayne, Performance management strategies. How to create and deploy effective metrics, Renton: TDWI, 2009.

[23]. Q. Segue, What is Software Performance Testing? Retrieved August 25, 2014, from https://www.seguetech.com/what-is-software-performance-testing/.

[24]. F.K. Muhammad, A.Q. Khurram and A.S. Khurram, Performance Evaluation of Software Development Models. Software Engineering, 2013, vol. 3, no.1, p. 1-4.

[25]. B. B. André, Incorporating Software Performance Engineering Methods and Practices into the Software Development Life Cycle, ACM Digital Library, 2016, p. 327-330.

[26]. T. Tom, Improving the performance of thermal power systems using software Workflow, Invensys, Informatsionnyy document, 2015, pp. 1-8.

[27]. O. Dieste, A. Aranda, F. Uyaguari, B. Turhan, A. Tosun, D. Fucci, M. Oivo, and N. Juristo, Empirical evaluation of the effects of experience on code quality and programmer productivity: an exploratory study, Empirical software engineering, 2017, vol. 22, no. 5, p 2457-2542.

[28]. D. Kamma, and P. Jalote, High Productivity Programmers Use Effective Task Processes in Unit-Testing, Proceedings of the 22nd asia-pacific software engineering conf., 2015,. pp. 32-39.

[29]. D. Kamma, Study of task processes for improving programmer productivity. Proceedings of the 36-th international conf. on software engineering, 2014, pp. 702-705.

[30]. J.H. Yahaya, A. Deraman, A. Tareen and A.R. Hamdan, Software Quality and Productivity Model for Small and Medium Enterprises, International journal of advanced computer science and applications, 2017, vol. 8, no. 5, p. 316-320.

[31]. M.Z Yusoff, M. Mahmuddin, and M. Ahmad, The effect of knowledge work productivity factors on software development, Journal of engineering science and technology, 2017, vol. 12, p. 64-73.