Green software.

Software can be considered green if it helps save resources. Resources in this context is a very broad concept. In addition to direct energy, material, human resources, as well as natural resources should be included here

The program is information, that is, an immaterial object that has a direct impact on the objects of our material world. Such influence is carried out indirectly through various technical means. It is from this that it follows to proceed with regard to the problems of green software. The program during its functioning should ensure the minimization of resources consumed by these technical means.

One of the reasons for the incorrect functioning of the software is the defects introduced into the program at the stage of its creation, modification or during the correction of other defects. The process of manifestation of these defects over time is the subject of a separate area of study - software reliability. In the case of software systems of critical purpose, the correctness of their functioning is the main and decisive factor.

If we want to create green software, certain metrics should be introduced, based on which it will be possible to judge that the created program is truly green. Based on these metrics, it will be possible to compare two programs with each other to say which one is greener to a greater degree. That is, in fact, such metrics will show shades of greenness of the software. And having determined the range of acceptable values of these metrics, we will be able to judge whether the program under study is green.

The main characteristics that determine the quality of software:

- Functionality - includes a set of attributes that show the existence of certain software functions with their given properties. Attributes show how these functions satisfy stated or anticipated needs.

- Reliability - a set of attributes showing the software's ability to maintain its level of performance under specified conditions for a specified period of time.

- Security - a set of attributes characterizing the degree of confidentiality and preservation of information, as well as fault tolerance and the possibility of confirming the authenticity of the software.

- Usability - a set of attributes that indicate both the effort required to use the software and the individual evaluations of such use by potential users.

- Performance - a set of attributes that are related to the relationship between the level of software performance and the amount of resources used under given conditions.

- Maintainability - a set of attributes showing the effort and resources required to make the necessary changes during the operational phase.

- Mobility - a set of attributes that show the ability of software to be transferred from one operating environment to another.

Research directions in the field of security and sustainability of green software can be:

1. Development of methods of self-monitoring of software for the integrity of its code and the authenticity of this code of an uninfected sample. At each run, mission-critical green software can compare its code against a benchmark. Such a comparison can be organized, for example, by calculating the CRC code of the current software version and comparing it with the reference CRC code. The reference code can be stored directly in the program body using the principles of steganography.

2. Development of methods for diagnosing intermediate calculations and final results of green software during its operation. Here you can go by using well-developed methods of diagnosing the operation of the hardware

3. "Handwriting of the creator" - establishing the authorship of the source or executable code of the programs that attacked the computer system. Knowing the author of the malicious code will allow us to apply legal measures of influence to him, thereby stopping his (the author's) criminal activity.

An energetic approach to software development and selection.

The words "energy", "energy" has many meanings. In the context of green technologies, they often indicate the problem of saving physical energy consumption (the solution of which depends on reducing the rate of undesirable irreversible changes in the environment).

In the context of software technologies, these words can be associated with decision-making based on metrics that characterize software systems by analogy with how physical systems and processes are characterized using the concepts of energy and work.

The general, essentially philosophical, concept of energy conservation and transformation suggests that in the field of green information technologies (Green IT) the mentioned interpretations of energy should be interconnected and, due to this, may suggest new ways of solving known problems.

Recognized ways of saving natural resources in connection with IT are:

1) improvement ("greening") of all IT systems and their use;

2) application of IT to support ecological balance;

3) using IT to achieve a new level of understanding of environmental problems ("green understanding").

Information and energy cost estimation of the software development process

The number of parameters for a homogeneous and practically isolated program module in its internal content was discovered by M. Halstead in the 1970s.

One of these parameters is named scope (simple program or program module). Similar to the usual volume, this value is calculated on the basis of extremely simple measured values:

$$V = V(N, \eta) = N \log \eta$$

where V is the volume of the program module;

η - the dictionary of the program module (the number of different program symbols used in the source text of this module, a program symbol is one or more lexical units), N is the length of the program module (the total number of program symbols, taking into account their repetition).

Another parameter was called by Halsted potential by volume:

$$V^* = V(\eta^*, \eta^*) = \eta^* \log \eta^*$$

where V* is the potential volume;

$$\eta^* = 2 + \eta_2^*$$

η^* **- potential dictionary of the program module,**

η_2 **- input-output parameters.**

The difficulty of developing the module
where W is the volume of module development.

$$D = W/V^*$$

Using metric estimates for software comparison for degree of "greenness".

In order to become green, software must meet the following requirements:

1. The requirement to minimize the resources required for the software to function. Such resources include the amount of computer RAM, clock frequency, and speed of the central processor.

2. The requirement to save resources consumed by peripheral devices controlled by the software. Such resources directly include the amount of energy consumed and the associated emissions of harmful substances into the atmosphere.

3. Requirement for reliability of operation and stability in relation to external targeted influences aimed either at the termination of operation or at obtaining control over the software.

Memory efficiency metrics.

Metric	Index	Explanation
M01	The number of class methods not used in the program	Reserved, but not used by the program, RAM is evaluated
M02	Size of unused memory in statically defined arrays	
M03	The size of the never-executed sections of the program code	Occupied but unused memory
M04	The number of reused text or non-integer constants of the same value	Memory that is not being used efficiently

Metrics of the efficiency of the use of processor time

Metric	Index	Explanation
T01	The number of operators of multiplication and division of integral variables by the power of two	Unjustified use of resource-intensive environments
T02	The number of integer exponentiation operators using built-in functions	
T03	Total number of measurements in all static arrays	Tools that potentially increase execution time
T04	The total number of loop statements in the program	

Execution time performance metrics

Metric	Index	Explanation
TM01	Number of arrays in procedure parameters passed by value	An array is being copied
TM02	Number of recursive procedures	Recursion requires additional time and memory
TM03	Total level of recursion (if it can be determined by static text analysis)	

Для кожного програмного коду оцінюються чисельні значення показників метрикам, наведеним у таблицях.

For each software code, numerical comparisons of indicators to the metrics listed in the tables are evaluated.

For each indicator, the value of the corresponding normalizing coefficient is selected and calculated.

For each studied program, the value of the general criterion of the degree of "greenness" of the software is calculated:

$$G = \sum_{k=1}^{M} \frac{E_k}{N_k}$$

where M is the number of metrics,

E_k - the value of the indicator, which is evaluated according to the k-th metric,

N_k - the value of the corresponding normalizing coefficient.