AMERICAN JOURNALS
Open Access,
Peer Reviewed Journals

American Journal of Pedagogical and Educational Research
ISSN (E): 2832-9791| Volume 19, | December, 2023

SOFTWARE PROCESS DESIGN PROBLEMS AND METHODICAL, TECHNOLOGICAL, INSTRUMENTAL AND ORGANIZATIONAL SUPPORT

Abdullayev Ravshan Narzulloyevich
Teacher of TATU Karshi Branch

ABSTRACT

designing is now being practiced by software developers. Software design reflects hierarchical decomposition, which consists of dividing a complex project (problem) into several easily solvable subproblems. Design encompasses the issues associated with creating software and shapes the software development activities from designing the architecture of the software. The design consists of a complex of standards, instructions, methods, incorporating the rules of the methodological support of the software.

KEYWORDS

Project, software, method, design, hierarchy, model, design problems, methodology, technology, instrumental support of design, interactivity, information systems, interconnection.

Introduction

Introduction. “Projecting” is defined in the dictionary as "to shape according to plan". The term "software projecting" was created after two NATO conferences in 1968 and 1969.

Software design reflects hierarchical decomposition, which consists of dividing a complex project (problem) into several easily solvable subproblems. Small problems are in turn divided into sub-problems until each detail is clearly defined.

Strict hierarchical decomposition obeys the following rules :

1. At each level of the hierarchy, the plan or project must be completed at the current level of detail;
2. Each part of an arbitrary level of the hierarchy fully covers a particular function or problem corresponding to the current level of detail.

Some forms of hierarchical decomposition of software are called :

- software design from top to bottom;
- software structural design;
- structured plan and details of the product;
- step-by-step software projecting;
- top-down programming;
- modular programming;
- structural programming;

Design covers various activities related to the creation of software - from the functional decomposition of the system , from the definition of the architecture of the software product to the completion of the

software product, and can be divided into the following stages : design , programming and software development. evaluation _
At this stage, the software is created and 90% has its final form.

Design problems for large software products

As a rule, large software tools incorporate all the properties of a complex system. They have a large number (hundreds, thousands) of component-modules and are tightly connected to solve a common problem.
A hierarchical structure with several level groups and modular components is used to ensure the interdependence of the components that make up a single complex.
Each module has its own purpose and quality criteria, which, as a rule, do not coincide with the main purpose and quality criteria of the complete software product.
Especially in a software product with a hundred modules, it is very difficult to organize appropriate use of EHM resources in order to realize the main efficiency criterion while keeping a number of quality indicators within the permissible limits.
Due to the large number and complexity of program execution paths, programs require high robustness to input data errors and output data errors, as well as EXM corruption. This is defined by the concept of "software reliability".
Creating large software products with specified characteristics with limited resources requires a set of activities to achieve the set goal, and this is called a "project".
Goal-oriented project management is the proportional distribution of resources between the activities related to the creation of a software product during the full design cycle.
In order to increase labor productivity in the creation of complex software products, it is necessary to standardize and comprehensively automate the technological process of creating a software product.
A specification for each program in the software must be specified. It describes the functions of the program in detail and makes it possible to fully test its operation.
These measures allow to widely use separate programs in different systems without the participation of the creators, to replace obsolete components without damaging the software product.
Usually, it is appropriate to create large software products in the form of a product set (a group of programs and modules) .
It is necessary to standardize the structure and form of necessary documents for each developed and tested program and software products . Currently, such a standard exists and it is called "Uniform System of Software Documentation".
Another issue is to standardize the structure of programs and their integration rules to ensure the transfer of control and data exchange. Description of variables and their use, rules of memory allocation, requirements for information exchange between individual programs, program complexes and autonomous control systems should be brought to a uniform form.
It is also necessary to standardize the methods and requirements for ensuring and determining the quality of complex software products. These methods should make it possible to control the performance of the software product created in real conditions. .
For many years, attempts to create a universal high-level algorithmic language that provides convenient creation of high-quality programs have not ended successfully.

2-3 programming languages are used within the framework of the system in each class, in particular, in practice, some part of the program is always created in autocode.

It is considered appropriate to standardize programming languages and the structural structure of programs within the framework of a class of programs, while maintaining the possibility of creating programs in autocode.

The mentioned issues of standardization should be combined with a single technological scheme and methodology for the development of large software products.

Designing large and high-quality software products is a complex and long-lasting work process of a team of specialists of various specialties . It is necessary to organize, organize and automate this work process using modern effective methods and tools that support the design of large software products .

The methodological support of software design consists of a complex of standards, instructions and methods that embody the rules of software design .

The documentation defines the development object construction and its creation process. In the methodology and instructions , programming languages, rules for the use of symbols and signs, structural construction of program components and rules for their interaction, and other methodological principles of program complex organization are specified in the methodology and instructions. In addition, the documents contain the methodological basis for creating a software product: rules for programming and debugging components, testing and quality assessment of components, and other rules. A concrete enterprise standard is created to create a concrete software product or a group of software products of one class on the basis of state and industry standards, which consist of the methodological bases of designing software products.

Together, these documents represent the methodological aspects of creating a concrete software tool.

Technological support of software design

The technological support of the software design process consists of detailing the methodological support documents that determine the concrete technology of the software product life cycle.

Technical support documentation of the development is closely related to the technology of monitoring and exploitation of the software product. These documents define the design phase, its results and methods of monitoring compliance with the specified technology. Forms technology implements methods and criteria for evaluating the quantity and quality of a software product at various stages of creating a software product. For each stage of creating a software product component, the cost and duration of the project is determined, taking into account the characteristics of the object.

In the technology of creating a concrete software product, the form of using instrumental means of automating the creation of a software product is determined.

As a result, the technological process is provided with methods, documents and automation tools. Together, these ensure the required quality of the software product.

Instrumental support of the software design process

The instrumental support of the software consists of the software tools of the computing technique that ensure the automation of the process of creating the software product .

The degree of software automation and instrumentation depends on the provision of software development and monitoring. It includes software tools for automation of the technological process of development, preparation and monitoring of the software product, as well as hardware tools for calculation, ignition and circulation techniques.

Software development of software developers is determined by the functional capabilities of software systems that automate the creation of software (DTYAADT) or, as it is called, automated systems of software design (DTLAT).

For each stage of software development, different methods and tools can be used with different effectiveness (depending on the nature of the project).

The level of primary software equipment of developers can be characterized by the size of programs in typical (model) technology.

A high level of software equipment for developers is achieved by automating all stages of development, preparation and monitoring of a software product.

The following software tools are used :

1)translators of program texts and program instructions in a high-level language;

2)planning and control tools for static and dynamic testing of programs;

3)software modeling tools of external environment objects for the software product;

4) automated management and control tools for software product development;

5) tools of modern methods of automation of creation of large software products.

DTLATs, which combine all these methods and tools, have a size of up to 1 million teams, and the cost of their creation is several million soums. However, the average level of developer software is low-cost .

The demand for DT design systems depends on the complexity of the object, the resources available to create the software, and a number of constructive and organizational factors .

The requirements for automation systems, which require the creation of large software tools, are as follows :

— reduction of total labor costs and duration of software development;

- increase the labor productivity of creators;

— ensuring the high quality and reliability of the created and monitored software product;

— complex automation of large-scale and sufficiently complex software creation with the team;

— to provide a generalized technology of creation and monitoring of software products;

— Ensuring effective use of memory and performance resources of ECs.

Software developers' hardware availability is determined by the power of the individual ECs used.

REFERENCES

1. Гецци К., Джазайери М., Мандриоли Д. Основы инженерии программного обеспечения / Пер. с англ. 2-е изд. СПб.: БХВ- Петербург, 2015. 832 с.

2. Камаев В.А., Костерин В.В. Технологии программирования. Учебник, 2-е изд., перераб. и доп. М.: Высшая школа, 2006 454 с.

3. Макконнелл С. Совершенный код. Мастер-класс / Пер. с англ. — М. : Издательство «Русская редакция», 2010. — 896 стр.

4. Мацяшек Л.А. Анализ требований и проектирование систем. М.: Издательский дом "Вильямс", 2002.428с.

5. Diehl S. Software Visualization: Visualizing the Structure, Behaviour, and Evolution of Software. Berlin Heidelberg: Springer-Verlag, 2007. 187 p.

6. Booch, Rumbaugh, Jacobson. "The Unified Modeling Language User Guide", 2nd Edition.

7. Ergashev Nuriddin Gayratovich. “DIDACTIC PROVISION OF THE IMPLEMENTATION OF THE TEACHING MODEL OF INFORMATION TECHNOLOGY IN TECHNICAL SYSTEMS IN TECHNICAL SPECIALTIES OF HIGHER EDUCATION ON THE BASIS OF A HIERARCHICAL APPROACH”. Intent Research Scientific Journal, vol. 2, no. 12, Dec. 2023, pp. 28-40, https://intentresearch.org/index.php/irsj/article/view/272.

8. Ergashev, Nuriddin. "Ergashev Nuriddin G’ayratovich N. G’. Ergashev, A. O’. Shukurov. SN Siradjev. Raqami axborot texnologiyalari. O ‘quv qo ‘llanma. Intelekt, Qarshi 2023. 220-b.: N. G’. Ergashev, A. O’. Shukurov. SN Siradjev. Raqami axborot texnologiyalari. O ‘quv qo ‘llanma. Intelekt, Qarshi 2023. 220-b." E-Library Karshi EEI 1.01 (2023).

9. Ergashev, Nuriddin. "Ergashev Nuriddin G’ayratovich N. G’. Ergashev, XX Nekboyev, ZE Chorshanbiyev. Iqtisodiyotda axborot-kommunikatsion texnologiyalar va tizimlar. darslik. Intelekt, Qarshi 2023. 244-b.: N. G’. Ergashev, XX Nekboyev, ZE Chorshanbiyev. Iqtisodiyotda axborot-kommunikatsion texnologiyalar va tizimlar. darslik. Intelekt, Qarshi 2023. 244-b." E-Library Karshi EEI 1.01 (2023).

10. Ergashev, Nuriddin. "Ergashev Nuriddin G’ayratovich N. G’. Ergashev, ZE Chorshanbiyev, SN Siradjev. Texnik tizimlarda axborot texnologiyalari fanidan masalalar to ‘plami. O ‘quv qo ‘llanma. Intelekt, Qarshi 2023. 160 b.: N. G’. Ergashev, ZE Chorshanbiyev, SN Siradjev. Texnik tizimlarda axborot texnologiyalari fanidan masalalar to ‘plami. O ‘quv qo ‘llanma. Intelekt, Qarshi 2023. 160 b." E-Library Karshi EEI 1.01 (2023).

11. Ergashev, Nuriddin. "Ergashev Nuriddin G’ayratovich ZT Raximov, AA Xo ‘jayev, Ergashev N. G’. Texnik tizimlarda axborot texnologiyalari. Ekologiya va atrof-mahit muhofazasi (sanoat korxonalari) yo ‘nalishi talabalari uchun o ‘quv qo ‘llanma.-Toshkent.-2020.-215 b.: ZT Raximov, AA Xo ‘jayev, Ergashev N. G’. Texnik tizimlarda axborot texnologiyalari. Ekologiya va atrof-mahit muhofazasi (sanoat korxonalari) yo ‘nalishi talabalari uchun o ‘quv qo ‘llanma.-Toshkent.-2020.-215 b." E-Library Karshi EEI 1.01 (2023).

12. Ergashev, Nuriddin. "Ergashev Nuriddin G’ayratovich Texnik tizimlarda axborot texnologiyalari. Darslik: N. G’. Ergashev. Texnik tizimlarda axborot texnologiyalari. Darslik. Intelekt, Qarshi 2023. 259-b." E-Library Karshi EEI 1.01 (2023).

13. Ergashev, Nuriddin. "Ergashev Nuriddin G’ayratovich N. G’. Ergashev, BJ Xoliqulov. Axborot texnologiyalari va jarayonlarni matematik modellashtirish. Darslik. Intelekt, Qarshi 2023. 261-b.: N. G’. Ergashev, BJ Xoliqulov. Axborot texnologiyalari va jarayonlarni matematik modellashtirish. Darslik. Intelekt, Qarshi 2023. 261-b." E-Library Karshi EEI 1.01 (2023).

14. Davronovich, Shodiyev Rizamat, and Ergashev Nuriddin Gayratovich. "ANALYSIS OF EXISTING RISKS AND METHODS OF COMBATING THEM IN CLOUD TECHNOLOGIES." American Journal of Pedagogical and Educational Research 18 (2023): 190-198.