AJEMA GLOBAL RESEARCH NETWORK

AMERICAN Journal of Engineering, Mechanics and Architecture
Volume 01, Issue 02, 2023 ISSN (E): XXX-XXX

CASE Technologies

Ergashboyev Javohirbek Shavkatjon o'g'li, Sattorov Muhammadjon Foziljon o'g'li, Isoqhonova Sarvinoz Ilhomjon qizi
Students of the Fergana branch of TUIT named after Muhammad al-Khorazmi

Abstract: In three articles, the term "CASE" (Computer Aided Software Engineering) is defined in a very broad sense, and the original meaning of the term CASE has now acquired a new meaning that interrupts the production process of complex IT. information about.

Keywords: CASE, IT-based support, hardware management, control documents, TRW, IP, IC design.

Trends in the development of modern information technologies lead to a constant increase in the complexity of IP created in various sectors of the economy. Modern large projects, as a rule, are characterized by the following features:

- the complexity of the description, which requires careful modeling and analysis of data and processes (a sufficient number of functions, processes, data elements and complex relationships between them);
- the existence of a set of closely related components (subsystems) that have their own local tasks and performance goals (for example, an related to processing transactions and solving routine tasks Analytical processing applications using traditional applications and specialized software (decision support) queries to big data);
- lack of direct analogues, limiting the use of any typical design solutions and practical systems;
- the need to integrate existing and newly developed applications;
- working in a heterogeneous environment on several hardware platforms;
- dispersion and homogeneity of certain groups of developers in terms of skill level and established traditions of using certain tools;
- the critical time frame of the project implementation is related, on the one hand, to the limited capabilities of the development team, and on the other hand, to the scale of the ordering organization and the different levels of preparation of its individual departments. Implementation of IS.

The listed factors contributed to the emergence of a special category of software and technology tools - CASE-tools that implement CASE-technology to create and support IS.

Currently, the term "CASE" (Computer Aided Software Engineering) is used in a very broad sense. The original meaning of the term "CASE", which was limited only to the issues of software development automation, has now acquired a new meaning that encompasses the entire process of complex IT development. Now, the term "CASE-tools" refers to software that supports IT creation and maintenance processes, including requirements analysis and

formulation, design of application software (applications) and databases, code generation, testing, documentation, quality assurance, means configuration. management and project management, as well as other processes. CASE tools together with system software and hardware form a complete IT development environment.

Before the appearance of CASE-technology and CASE-tools, research in the field of programming methodology was carried out. Programming has adopted the characteristics of a systems approach with the development and implementation of high-level languages, structured and modular programming methods, design languages and their support tools, and formal and informal languages for describing system requirements and specifications. and others. In addition, the following factors contributed to the emergence of CASE technology:

- prepare analysts and programmers who accept concepts of modular and structured programming;
- widespread introduction and continuous growth of EHM productivity, which allowed the use of effective graphic tools and the automation of most of the design stages;
- the introduction of network technology that allowed to combine the efforts of individual performers into a single design process using a common database containing the necessary information about the project.

CASE-technology is an IT design methodology, as well as a set of tools that visually simulate the subject area, analyze this model at all stages of IT development and maintenance, and develop applications in accordance with the information needs of users. . Most of the existing CASE tools are based on systematic (mainly) or object-oriented analysis and design methodologies, which use diagrams or text to describe external requirements, relationships between system models, system behavior dynamics, and software architecture. ' use specifications of the form. 16].

A number of limitations appeared when using structural analysis methodologies (complexity of understanding, high labor intensity and cost of use, inconvenience of making changes to design specifications, etc.). From the beginning, CASE technologies evolved to overcome these limitations by automating analysis processes and integrating tools. They have the following advantages and capabilities.

1. Unified graphics language. CASE-technologies provide all participants of the project, including customers, with a single strict, clear and intuitive graphic language, which allows to get visible components with a simple and understandable structure. At the same time, the programs are represented by two-dimensional diagrams (easier to use than multi-page descriptions), which allows the client to participate in the development process, and the developers - to communicate with subject matter experts. the activities of system analysts, designers and programmers, which makes it easier for them to defend the project in front of management, as well as facilitate system maintenance and changes.

2. The single database of the project. The basis of CASE technology is the use of a project database (repository) to store all information about the project, which can be shared among developers according to the right of use. The content of the repository includes not only various types of information objects, but also the relationships between their components, as well as the rules for using or processing these components. The repository can store more than 100 types of objects: structure diagrams, screen and menu definitions, report projects, data descriptions, processing logic, data models, their organization and processing, source codes, data elements etc.

3. Integration of funds. Integration of CASf-tools and separation of system data between developers is performed on the basis of the repository. At the same time, the capabilities of the repository provide several levels of integration: a common user interface for all tools, data transfer between tools, integration of development stages through a single system for

representing life cycle stages, communication between different tools. data and media transfer. platforms.

4. Team development and project management support. CASE-technology is a project that provides the ability to work on the network, export-import any project fragments for their development and (or) modification, as well as planning, control, management and interaction supports group work, ie. functions required in the development and maintenance of projects. These functions are also implemented on a repository basis. In particular, security controls (restrictions and access privileges), version control and changes, etc. can be implemented through the repository.

5. Prototyping. CASE technology makes it possible to quickly build mock-ups (prototypes) of the future system, which allows the client to assess how acceptable and suitable it is for future users in the early stages of development.

6. Formation of documents. All project documents are created automatically based on the repository (as a rule, in accordance with the requirements of current standards). The undoubted advantage of CASE technology is that the documentation always corresponds to the current state of affairs, because any changes in the project are automatically reflected in the repository (as you know, with traditional approaches to software development, documentation is best delayed, and a line changes are not reflected in it at all).

7. Project verification. According to the statistical data of the analysis of five large projects of TRW (USA), CASE-technology provides automatic verification and control of the completeness and consistency of the project in the early stages of development, which affects the success of the development as a whole. , design and coding errors are 64% and 32%, respectively. The total number of errors and design errors are 100 times more difficult to detect in the maintenance phase than in the requirements analysis phase.

8. Automatic generation of object code. Machine code generation is repository-based and allows for automatic construction of up to 85-90% of object code or text in high-level languages.

9. Maintenance and reengineering. Within the framework of CASE technology, system maintenance is characterized by maintaining the project, not the program codes. Reengineering and reverse engineering tools create a system model from its codes and integrate the resulting models into the project, automatically update documents when changing codes, etc. allows.

When using CASE technologies, all stages of the IT life cycle change, the biggest changes are related to the analysis and design stages. Table 3.1 shows the main changes in the IT life cycle when using CASE-technologies compared to traditional development technology.

Table 3.7. Comparative descriptions of the main changes in the IP life cycle

Traditional development technology	Development using CASE technologies
The main effort is to code and test	The main effort is focused on analysis and design
"Paper" specifications	A fast repetitive routine

End of table. 3.1

Traditional development technology	Development using CASE technologies
Manual coding	Automatic generation of machine code
Software testing	Automatic project management
Saving the program code	Project support

Table 3.2 shows an estimate of labor costs by stages of the software life cycle, depending on the development technology used.

Let's move on to the features of modern CASE-systems, which cover a wide range of support for many IC design technologies: from simple analysis and documentation tools to full-scale automation tools that cover the entire software life cycle.

Table 3.2. Estimation of labor costs by software life cycle stages depending on the development technology used, %

Technology	Analysis design programming test	Analysis design programming test	Analysis design programming test	Analysis design programming test
development	20	15	20	45
Traditional	30	30	15	25
Structural methodology (manual)	40	40	5	15

Let's move on to the features of modern CASE-systems, which cover a wide range of support for many technologies for IC design: from simple analysis and documentation tools to full-scale automation tools that cover the entire software life cycle [16].

The category of CASE tools includes both relatively inexpensive systems for PCs with very limited capabilities and expensive systems for heterogeneous computing platforms and operating environments. Thus, there are about 300 different CASE-tools on the modern software market, the most powerful of which are used by almost all leading Western companies in one way or another.

References

1. Kodirov, E., Turgunov, B., & Muxammadjonov, X. (2019). IN THE WORLD REFUSES TO USE FACE RECOGNITION TECHNOLOGY. *Мировая наука*, (9 (30)), 34-36.
2. Turgunov, B., Komilov, A., Abdurasulova, D., & Umarov, X. (2018). Security of a smart home.
3. Тохиров, Р., Тургунов, Б., & Мухаммаджонов, Х. (2019). СТРУКТУРНАЯ СХЕМА БЛОКА РАСПОЗНАВАНИЯ РЕЧИ В АВТОМАТИЗИРОВАННОЙ СИСТЕМЕ УПРАВЛЕНИЯ. *Форум молодых ученых*, (7), 322-324.
4. Абдурахмонов, С. М., Кулдашов, О. Х., Тожибоев, И. Т., & Тургунов, Б. Х. (2019). Оптоэлектронный двухволновый метод для дистанционного контроля содержания метана в атмосфере. *Письма в Журнал технической физики*, *45*(4), 11-12.
5. Тургунов, Б., Комилов, А., Абдурасулова, Д., & Асроров, С. (2018). Применение беспроводных сетевых технологий в медицинских измерительных системах.
6. Kodirov, E., Muxammadjonov, X., & Turgunov, B. (2019). INDUSTRIAL" INTERNET OF THINGS": THE BASIS OF DIGITAL TRANSFORMATION. *Теория и практика современной науки*, (9), 3-5.