

Impact Factor:

ISRA (India) = 4.971
ISI (Dubai, UAE) = 0.829
GIF (Australia) = 0.564
JIF = 1.500

SIS (USA) = 0.912
ПИИИ (Russia) = 0.126
ESJI (KZ) = 8.997
SJIF (Morocco) = 5.667

ICV (Poland) = 6.630
PIF (India) = 1.940
IBI (India) = 4.260
OAJI (USA) = 0.350

SOI: [1.1/TAS](#) DOI: [10.15863/TAS](#)

International Scientific Journal Theoretical & Applied Science

p-ISSN: 2308-4944 (print) e-ISSN: 2409-0085 (online)

Year: 2020 Issue: 09 Volume: 89

Published: 26.09.2020 <http://T-Science.org>

QR – Issue



QR – Article



M.R. Malikov

Samarkand State Medical Institute

Associate Professor of the Department of Information Technology

A.A. Rustamov

Samarkand State Medical Institute

Assistant of the Department of Information Technology

N.I. Ne'matov

Samarkand State Medical Institute

Assistant of the Department of Information Technology
Republic of Uzbekistan, Samarkand

STRATEGIES FOR DEVELOPMENT OF MEDICAL INFORMATION SYSTEMS

Abstract: This article, aimed primarily at professionals, contains a description of methods, models of technologies for the development of complex information systems, which include MIS. The main focus is on the organization and design of the storage subsystem.

Key words: MIS, technology, strategy, medicine, system, model, method, principle, integration.

Language: English

Citation: Malikov, M. R., Rustamov, A. A., & Ne'matov, N. I. (2020). Strategies for development of medical information systems. *ISJ Theoretical & Applied Science*, 09 (89), 388-392.

Soi: <http://s-o-i.org/1.1/TAS-09-89-49> **Doi:**  <https://dx.doi.org/10.15863/TAS.2020.09.89.49>

Scopus ASCC: 1710.

Introduction

1. PROBLEMS AND ERRORS IN THE USE OF COMPUTER TECHNOLOGY

The lack of strategic and tactical planning at the stage of developing a medical information system usually leads to the following problem situations:

- incompatibility of interfaces of some systems;
- Lack of integrated access to medical, administrative or reference information;
- inadequacy of the system to the requirements of the end user;
- lack of expected performance;
- lack of necessary support for standards;
- insufficient or exhaustion of system resources;
- discrepancy between the applied information technologies and the strategy of the medical organization.

Most of these situations arise not because of technological errors, but because of deficiencies in control. Moreover, the problem lies in the absence or

inadequacy of the methodology for using and managing existing technologies [Bourke, 1994]. Most of the failures in the development of information systems projects are caused not by technological failures, but by methodological and organizational errors, among which the following can be distinguished:

- incorrect prioritization in the organization of work;
- selection of standards and technologies that are not adequate for the tasks set;
- inability to achieve consensus and agreed vision of problems;
- non-observance of organizational and technical requirements;
- lack of provision of technical personnel with appropriate tools, skills and authority;
- lack of clearly set goals, methods for assessing efficiency and control and accounting policies;

Impact Factor:

ISRA (India) = 4.971
ISI (Dubai, UAE) = 0.829
GIF (Australia) = 0.564
JIF = 1.500

SIS (USA) = 0.912
PIIHQ (Russia) = 0.126
ESJI (KZ) = 8.997
SJIF (Morocco) = 5.667

ICV (Poland) = 6.630
PIF (India) = 1.940
IBI (India) = 4.260
OAJI (USA) = 0.350

- incorrect organization of access and secrecy of information. For the successful implementation of the information system, it is necessary to adhere to the accepted standards and models for supporting the software life cycle.

2. MODELS AND METHODS OF ORGANIZING SOFTWARE DEVELOPMENT PROJECT MANAGEMENT MODEL:

The requirements analysis defines the objectives of the system being developed and specifies the requirements of future users. Figuratively speaking, this stage should formulate the answer to the question: "WHY is the system needed?"

The specification phase defines the requirements of the users in terms of the functionality of the computer system as that functionality would appear from the outside. The question to be answered is "WHAT is a system?"

The design phase provides an accurate model of the system and a detailed description of its implementation ("HOW to build the system?"). This phase is often divided into two steps: architectural design and detailed design, the result of which should be a kind of formalism, on the basis of which further coding of programs will be carried out.

The implementation and development phase corresponds to writing programming code.

The validation phase is the verification of the adequacy of the system to the specified requirements. It implies installation and testing of the system in real life situations.

During the maintenance and support phase, system updates and improvements are carried out in accordance with the modified requirements.

A feature of this model is the following: not a single step can begin until the previous step is completed and its compliance with the requirements is checked at a certain checkpoint.

PROJECT MANAGEMENT METHODS

RACINES (an acronym for RAtionalization des Choix INformatiquEs) is some action guide for project organizers that was first published in 1988 by the French Ministry of Industry. This method formalizes the definition and implementation of a strategic plan for organizing projects in accordance with the following five steps:

The first step is the stage of evaluating the possibilities and preparing the project. Involves the implementation of work structures, usually organized around a management committee that acts as a decision-making body, a user group or advisory body, and a project team that plays the role of the main manufacturer.

Assessment and orientation step - analysis of existing information systems and resources, identification of needs, priority boundaries and determination of the direction in which to move.

Consideration step for possible scenarios. A scenario is a strategy that fills a specified set of objectives with a strategic plan. Each proposed scenario should include a conceptual solution, an organizational solution, a technical solution, a financial calculation, a sequence of actions that is determined by the established priorities, and an assessment of the potential impact on the organization if this scenario occurs. However, scenarios are only a qualitative, not quantitative, model. As a result, only one scenario is selected, which turns out to be optimal taking into account all critical conditions.

The next step is to draw up an action plan. Includes the development of a progressive detailed quantitative description of the scenario chosen in the third step. At the same time, the sequence of actions and organizational measures is indicated along with the necessary technical and human resources.

The final step is the implementation of the strategic plan. Typically, scenario selection takes 6 to 18 months, while an action plan can take 5 to 10 years. Disadvantages of the method:

- The method does not always identify the required resources or quality control processes for the products obtained at each step.

- The method is not well suited for large projects, where the specification may change.

Software development technologies are progressing very quickly and currently allow significant results to be obtained in a short time. The result can be obtained in the form of prototypes, which are provided directly to the user.

The spiral method is based on the principle of incremental development. New features are added at each step (increment). Each turn of the spiral includes requirements analysis, specification, design, implementation, and validation.

At the end of each loop, a new version of the software is produced, which will be operated until the next phase ends.

Spiral method and rapid prototyping help improve the management of project resources. Explicit specification of system versions allows checking and verification of each version. While not guaranteeing that cascading errors (initial errors that multiply in the next stages) are eliminated, this method guarantees that they are eliminated, allowing you to return to the latest acceptable version of the system at any time.

However, system analysis gives rise to many intractable problems, including:

- Inability to obtain comprehensive information to assess the system requirements from the point of view of the customer.

- The customer does not have sufficient information about the data processing problem.

- The specification of the system is often incomprehensible to the customer due to the scope and technical terms.

Impact Factor:

ISRA (India) = 4.971	SIS (USA) = 0.912	ICV (Poland) = 6.630
ISI (Dubai, UAE) = 0.829	PIHII (Russia) = 0.126	PIF (India) = 1.940
GIF (Australia) = 0.564	ESJI (KZ) = 8.997	IBI (India) = 4.260
JIF = 1.500	SJIF (Morocco) = 5.667	OAJI (USA) = 0.350

The severity of these problems can be significantly reduced through the use of modern structural methods, among which structural analysis methodologies are central.

3. PRINCIPLES OF STRUCTURAL ANALYSIS

It is customary to call a structural analysis a method for studying a system, which begins with its general overview and then details it, acquiring a hierarchical structure with an increasing number of levels. All structural analysis methodologies are based on some general principles and guidelines.

The following two basic principles are used: the principle of "divide and conquer" and the principle of hierarchical ordering. The first principle means breaking complex and time consuming tasks into many smaller independent subproblems. The second declares the organization of parts of the system into tree-like hierarchical structures, i.e. the system can be understood and built in levels, each of which adds new details.

In addition to two basic principles, there are a number of others, no less important. Some of them are: the principle of abstraction, formalization, conceptual generality, consistency, etc.

DATA-FOCUSED STRUCTURAL ANALYSIS METHODOLOGIES

There are many methodologies for structural analysis designed to solve specific problems. Since we are interested in information systems development, we will only consider data-driven methodologies. The development of an enterprise-wide information system is a rather laborious and lengthy process. The combination of these two factors - the laboriousness of the development process and the information content (data orientation) of the software being developed - most often determines the choice of one of the existing DBMS as a development tool and method of presenting information.

4. DATA DESIGN

When designing data-driven systems, the inputs and outputs are the most important, not processing and computation. Therefore, the order of analysis and design is somewhat different from the traditional order when developing, say, real-time systems. Namely: data structures are defined first, procedural components are built as derived from data structures. In fact, the design process consists in defining data structures, merging them into the prototype of the program structure and filling this structure with detailed data processing logic.

Jackson's structural design is a classic example of this approach. Its basic design procedure is intended for "simple" programs (a "complex" program is broken down into simple ones using traditional methods) includes the following 4 stages:

- Data design phase.

- Program design stage.
- Operations design phase.
- The stage of designing texts.

At the design stage of data, it is necessary to define the structures of input and output data and, on their basis, build some model of the internal data representation. This requires:

Build a model of the functional requirements of the system at the level of control processes (analyzing the subject area).

Then build a data model at the entity-relationship level (by analyzing processes).

DATA FLOW DIAGRAMS

Data flow diagrams (DFDs) are the most popular tool for building a system model at the level of control processes. With their help, these requirements are broken down into functional components (processes) and presented as a network connected by data streams. The main goal is to demonstrate how each process transforms its inputs into outputs, as well as to reveal the relationships between the processes. DFDs are represented using Jordan notation.

The following concepts are basic:

Data stream. A mechanism for transferring information from one part of the system to another. They are depicted by named arrows, the orientation of which indicates the direction of movement of information.

Process. Produces output streams from input streams according to some rule. It is depicted as a circle, inside which the name of the process and its number are placed (for links to it inside the diagram).

Data storage (storage). Allows you to save data between processes. The information it contains can be used at any time after it has been defined, and the data can be selected in any order. It is depicted by two parallel horizontal lines, between which the name of the repository is written.

External entity. An entity outside the context of the system, which is a source or receiver of data (does not participate in data processing). It is represented by a rectangle with a name inside.

When constructing diagrams, it is advisable to observe the following sequence:

Dividing many requirements into groups.

Identification of external and internal system objects. Extract information about objects from the requirements for primary (input) and secondary (output) documents.

Identification of the main types of information circulating between the system and external objects.

Development of a common DFD. At the same time, proceed from the recommendation: place from 3 to 6-7 processes on each diagram (the limit of human capabilities of simultaneous perception). Combine more complex processes into one process, and group threads.

Formation of DFD of lower levels by decomposing complex processes into parts (in parallel

Impact Factor:

ISRA (India)	= 4.971	SIS (USA)	= 0.912	ICV (Poland)	= 6.630
ISI (Dubai, UAE)	= 0.829	ПИИИ (Russia)	= 0.126	PIF (India)	= 1.940
GIF (Australia)	= 0.564	ESJI (KZ)	= 8.997	IBI (India)	= 4.260
JIF	= 1.500	SJIF (Morocco)	= 5.667	OAJI (USA)	= 0.350

to decompose data streams) until functions can be split into parts.

Construction of specifications of elementary processes (at the level of algorithms).

If necessary - making changes to the DFD of any levels as the details of the requirements are clarified.

The result of completing all of these stages is a ready-made model of the functional requirements of the system and a clear understanding of the structure of the subject area. At the end of this stage, you can proceed to building the data model.

ESSENCE-CONNECTION DIAGRAMS

One of the most commonly used tools for developing data models are Entity-Relationship Diagrams (ERDs). ERDs provide a standard way to define data and the relationships between them. With their help, the detailing of the data stores of the designed system is carried out, the system components are determined, the ways of their interaction, including the identification of objects important for the subject area (entities), the properties of these objects (attributes) and their relations with other objects (links).

This notation was introduced by Chen and was further developed by Barker. These diagramming techniques are used to design relational databases.

An entity is a set of instances of real or abstract objects (people, events, states, ideas, objects, etc.) that have common attributes or characters. Any system object can be represented by only one entity, which must be uniquely identified. In this case, the name of the entity should reflect the type or class of the object, and not its specific instance (for example, PATIENT, not KIM N.I.).

An independent entity represents the independent data that is always present in the system and the relationship. Other entities may or may not exist.

A dependent entity represents data that is dependent on other entities in the system. Therefore, it must always have relationships with other entities.

An associated entity represents data that is associated with a relationship between two or more entities (see many-to-many relationships below).

A relationship in its most general form is a relationship between two or more entities. Verb groups are used to name relationships (has, can own, defines).

An unconstrained (mandatory) relationship is an unconditional relationship, that is, a relationship that always exists as long as there are related entities.

A constrained (optional) relationship is a conditional relationship between entities.

A substantially constrained relationship is used when the respective entities are interdependent in the system.

Relationships are used to identify the requirements by which entities are involved in relationships. Each relationship connects an entity and

a relationship and can only be directed from relationship to entity.

The value of a link characterizes its type and, as a rule, is selected from the following set:

{ "0 or 1", "0 or more", "1", "1 or more", "p: q" }

A pair of relationship values belonging to the same relationship determines the type of this relationship. The following types of relationships are used:

*1 * 1 (one-to-one);*

*1 * N (one-to-many);*

*N * M (many-to-many).*

Relationships of this type are used in the early design stages to clarify the situation. In the future, each of these relations must be converted into a combination of relations of types 1.

2 (possibly with the addition of auxiliary associative entities and the introduction of new relationships).

Demonstrating the relationship between the objects of a medical organization (polyclinic). According to this diagram, each doctor treats one or more patients. In addition, each patient may or may not be treated by one or more physicians. Each doctor conducts an outpatient appointment for which patients are registered (assigned). In the process of examining patients at an outpatient appointment, the doctor forms epicrisis that affect the diagnosis. Every diagnosis for a patient must have an author - a doctor who is responsible for making the decision.

Each entity has one or more attributes that uniquely identify each instance of the entity. Moreover, any attribute can be defined as key.

Entity drill-down is accomplished using attribute diagrams that expose the attributes associated with an entity. An attribute diagram consists of a drillable entity, associated attributes, and domains that describe the attribute value ranges. In the diagram, each attribute is represented as a relationship between the entity and the corresponding domain, which is a graphical representation of the set of possible attribute values. All attribute links have meaning at their end. The attribute name underscore is used to identify a key attribute.

An entity can be split and represented as two or more entity-categories, each of which has common attributes and / or relationships. The entity split into categories was called the general entity (at intermediate levels of decomposition, the same entity can be both a general entity and an entity-category).

To demonstrate the decomposition of an entity into categories, categorization diagrams are used. Such a diagram contains a general entity, two or more entity-categories, and a special node - a discriminator, which describes how entities are decomposed.

The ER-approach was further developed in the works of Barker, who proposed an original notation that made it possible to integrate the model description tools proposed by Chen at the top level. There is only

Impact Factor:

ISRA (India) = 4.971
ISI (Dubai, UAE) = 0.829
GIF (Australia) = 0.564
JIF = 1.500

SIS (USA) = 0.912
ПИИИ (Russia) = 0.126
ESJI (KZ) = 8.997
SJIF (Morocco) = 5.667

ICV (Poland) = 6.630
PIF (India) = 1.940
IBI (India) = 4.260
OAJI (USA) = 0.350

one type of diagram used in Barker notation - ERD. An entity on ERD is represented by a rectangle of any size, containing the entity name, a list of attribute names (possibly incomplete) and indicators of key attributes (the "#" sign in front of the attribute name).

All relationships are binary and are lines with two ends (connecting entities), for which a name, a degree of multiplicity (one or many objects participate

in a relationship) and a degree of obligation (a relationship between entities is required or optional) must be defined. For multiple links, the line is attached to the entity rectangle at three points, and for single links, at one point. For a mandatory link, a continuous line is drawn up to the middle of the link, for an optional one - a dotted line.

References:

1. Dixon, B. (2016). "Health Information Exchange: Navigating and Managing a Network of Health Information Systems", *Academic Press*, vol. 1, p.376.
2. Qarshiev, A.B., Nabieva, S.S., & Egamqulov, A.Sh. (2019). Medical information Systems. *International Scientific Journal Theoretical & Applied Science*, Issue 04, Vol. 72.
3. Bogoslovsky, V. V. (2009). "Prediction of productivity, adaptation capacity of species and hybrids of the silkworm for enzyme systems and proteins spectra the Text.: dissertation of candidate of biological Sciences," / V. V. Theological, Stavropol.
4. Feylamazov, S.A. (2016). "Information technologies in medicine: A manual for medical colleges". (p.163). Makhachkala: DBMK.
5. Sakiev, T., & Nabieva, S. (2018). Architecture of the medical information system. *International Scientific Journal Theoretical & Applied Science*. Section 4. Computer science, computer injeengineering and automation. Issue: 05 Volume: 61. Published: 14/05/2018, pp. 35-39
6. Sakiev, T., & Nabieva, S. (2017). Principles of computer design. *International scientific and practical journal "Theory and Practice of Modern Science"*, Issue No. 7 (25) (July, 2017).
7. Primova, H., Sakiev, T., & Nabieva, S. (2019). *Development of medical information systems*. XIII International scientific and technical conference "Dynamics of Systems, Mechanisms and Machines" November 2019, Omsk, Russia. (Scopus).
8. Karshiev, A., & Nabieva, S. (2019). Medical information systems. *International Scientific Journal Theoretical & Applied Science*. SECTION 4. Computer science, computer injeengineering and automation. Issue: 04 Volume: 72. Published: 30/04/2019, pp.505-508.
9. Sakiev, T., & Nabieva, S. (2017). Typical processes of AWP. *International scientific and practical journal "Theory and Practice of Modern Science"*, Issue No. 7 (25) (July, 2017).
10. Coplan, S. (2011). "Project Management for Healthcare", *Information Technology McGraw-Hill*, Education; vol. 1, p.288.