Abstract: This paper presents some basic approaches, strategies, methodologies used for developing information systems, in general, and decision support systems, in particular. The classical and iterative designing methods are widely used in information systems development. Each method is generally presented and there are highlighted the advantages and disadvantages of each method in decision systems development.

Keywords: decision support systems, development approaches, strategies, methodologies.

1. INTRODUCTION

A Decision Support Systems (DSS) are a class of interactive, flexible, and adaptable computer-based information systems, especially developed for supporting the solution of a non-structured or/and semi-structured problems for improved decision-making. It utilizes data, provides an easy-to-use interface, and allows for the decision maker’s own insights. In (Filip, [1]) a DSS is viewed as an anthropocentric, adaptive and evolving information system which is meant to implement the functions of a human support team that would otherwise be necessary to help the decision-maker to overcome his/her limits and constraints he/she may encounter when trying to solve complex and complicated decision problems that count.

Because of the DSS nature, usually it is strongly recommended to include in the DSS developing team, besides the specialists, the potential users.

The team members involved in the DSS development must collectively meet a series of requirements (Holsapple and Whinston [2] apud Filip [3]): to know the application field; to widely identify the requests related to the decisional task fulfilling and specific end users requirements; to have or gain access to adequate knowledge sources regarding the decisional problems approaching ways; to know one or more DSS designing methods; to know a widely set of informatics tools which can be used for application system development. The persons involved in the DSS development can meet to certain degrees the above-mentioned requirements.

2. APPROACHES

A DSS development involves several principles that must be followed:

- to globally approach the problem which must be solved;
- to use a unitary methodology for designing and developing the system;
- to apply the newest solutions and techniques in information systems designing and developing;
- to create the information system relatively independent of the organizational structure of the company where the system will be implemented;
- to involve the future system direct beneficiaries in the analysis, design and implement activities of the information system;
- to develop the designing activities in accordance with the low and with the user available resources;
- to forecast and control the potential software changes;
- to document the possible compromises inherent to the software developing;

In general, in the information systems development, there are three approaches: top-down approach, bottom-up approach and mixed approach).

2.1. Top-down approach

The top-down approach (Wirth [4]), based on the modularity principle, consists of the gradual decomposition of a complex system. This decomposition is implemented from the top to the bottom, to a fundamental modules level.

The decomposition follows the functional structure of the system. At the end of the process, the system functional tree is identified and the functional modules and the connections are defined on each hierarchical level, offering a description of each system component.

Through this approach, the system gains a modular hierarchical structure. Each component implements a particular functionality. The components are coordinated by the components placed on the immediate upper neighbor level. This approach is applicable to the complex wide range information systems.

When a system functional module is made, it can be set into service. Afterwards, the other future developed modules will be integrated into the system. The integration should not cause problems due to the unitary strategy system design.

The top-down approach involves a special effort in the analysis phase (a complex and very detailed analysis.
being necessary because of the informational processes complexity) and in the design and developing phases, which entails great financial efforts.

2.2. Bottom-up approach

The bottom-up approach, unlike the top-down one, is based on the aggregation principle. It consists of the identification from the bottom to the top of the system components and the gradual assembling of the modules and connections in order to obtain a single module corresponding to the system. The modules are defined on different hierarchical levels.

One advantage of the bottom-up approach is that it requests a shorter developing time and it is less expensive. However, the lack of a hardware and software unitary strategy and of a unitary designing solution generates the risk of a low degree of system modules integration.

2.3. Mixed approach

The mixed approach represents a combination of the two approaches presented above (top-down and bottom-up) in order to take advantage of the qualities of both approaches.

This main method feature is the top-down planning and the bottom-up executing (Masi [5]). This approach involves a definition of the information system components in accordance with the top-down strategy requests. The development and the integration of system components will be made according to the bottom-up strategy requests.

Usually, the DSS nature assumes a developing technique different from the classical one used for the transaction processing systems. The experience proved that the traditional designing and developing approaches are not appropriate for the DSS designing. The reason for that is the inexistence of a complete and comprehensive decision-making theory and the fast changes in the deciders’ environment conditions. The DSS users cannot define in advance the functions that the systems should accomplish, in order for the designer to implement it to the system (Sprague [6]).

3. STRATEGIES

There can be identified four developing strategies depending on the element which is the base of information system structure and the structural approaching: the functional decomposition strategy, data-flow strategy, data-structure oriented strategy and the object oriented strategy.

3.1. Functional decomposition strategy (functions-oriented)

This strategy implies developing the system modules in order to accomplish only one system function or sub-function. The interface with other modules should be as simple as possible. The strategy advantages are the simplicity, the relatively easy accomplishment of the users’ requests and the solutions generation on different levels of abstraction (system, sub-system, functions and sub-functions). The disadvantages of this strategy are related to the functions focused efforts (which leads to many redundant data accumulation), the inexistence of precise decomposition rules and the hardly relieving of the non-hierarchical interactions in the complex systems. Another possible disadvantage is that it may require skills, creativity and imagination in order to use the method well (Zeng [7]).

3.2. Data-flow strategy (processes-oriented)

This strategy represents the real world by data and processes flows. The processes-oriented strategy widely resembles the functional decomposition, involving the same advantages and disadvantages categories.

3.3. Data-structure oriented strategy

The data-structure oriented strategy was developed considering the fact that the data types used in the organization are changing less than the processing actions in the systems. Even if the data values change constantly, the data structure does not assume major changes if it was well designed at the beginning.

Two remarkable accomplishments in the field set the place of this new orientation in the system approaching: the data modeling with the entity-relation diagrams, by Peter P. Chen (Chen [8]) and the information engineering, by James Martin (Martin [9]).

3.4. Object-oriented strategy

The object-oriented strategy uses the object concept, considered as an entity which can be distinguished through other entities and that has a meaning in the application. The object associates the data and the processing actions in the frame of the same entity, the object interface being the only visible one.

The structural approaching specific to the object-oriented methods gets a pronounced conceptual character, diminishing the semantic distance between the system model and reality. The reduced coupling between the objects and the high cohesion obtained by encapsulating and polymorphism allows a better localization of the changes, which leads to a low level risk of unexpected effects.

The object-oriented information systems developing strategy presented above is most suitable for the DSS designing and developing (Filip [3]).

4. METHODOLOGIES

The information systems designing methodologies are many and can be grouped according to the method used.
Below we will be briefly present three designing methods for the following DSS technologies: Specific DSS, DSS generators, DSS tools (Sprague [6]). The methods are:

- The custom design method – consists in step by step system analyzing in a cyclical working process, with recursions and approaching the problems from their general form to the most detailed expression. As a feature of this method, the system designer is also its developer.

- Serial design method - an information system is constructed for a pilot organizational entity (model) which, by generalization, is implemented for other similar organisms or activities, by adaptation and integration. This method is focused on the use of programs packages which can be generalize, usually created by soft companies which are specialized in analysis work saving and in get into function time saving. Usually the system creators are different from the system designers.

- Automated design method - the system is developed with computer aided software tools, using automated design procedures.

4.1. The classical method (life-cycle)

This method, based on the system life cycle, is represented by a series of phases which must be progressively followed: problem identification and definition, diagnostic analysis in the feasibility study, system analysis, designing, programming, testing, installing, exploitation and maintaining (Filip [3]). Each of these phases treats the system globally and that leads to a long developing process. Figure 1 (De Kock, [10]) presents the activities necessary for the construction of a DSS using the classical method.

The primary advantage (Marakas, [11]) of this method is related to its structure and discipline. It is often used, especially in cases where there is a contractual relationship between the DSS developer and the end users. The main method disadvantage is its rigidity. The development is barely interactive and the place and the role of the future users are not clearly defined. The entire system elaboration is based on the designers which make an abstract work, starting from the actual state analysis.

4.2. The iterative method

The iterative design is a method with a permanent dialog between the designer and the user, the user being involved in the system developing and the designer in the system usage. A portion of the DSS system is quickly constructed, then tested, improved, and enlarged in systematic steps.

The elaboration process is structured in several cycles, one for each sub-system developing. The method described by Sprague and Carlson consists in the following steps:

- the designer and the user jointly define a sub-problem which will represent the start of the system developing. This sub-problem must be less important as dimension, clearly delimited, but important enough as utility for the decider;
- at the same time, the problem is analyzed and a prototype is easily elaborated. This prototype includes the main system functionalities;
- the subsystem is used and evaluated adding new representations, models and control structures after each expanding cycle.

Basically the first stages of the iterative design process are similar to the classical design until de first prototype is created. Starting with this point the prototype is constantly expanded until the end system is created. This process requires a significantly higher level of interaction between analyst and user (Marakas [11]).

The progressive system design has, among the advantages, the close cooperation between different categories of actors involved in the construction of the
decision support system. The iterative intercession allows a more active user involving into the DSS designing.

This method has also the advantage that it allows a constantly system evaluation not only an end evaluation as on the traditional designing. This user-oriented type of system, is flexible and so new versions of it can be created in order to solve different problems or to add new options.

In conclusion, the iterative method provides the following advantages (Turban and colleagues [12]):

- Both the user and the designer are involved;
- The user learns about the system during the DSS construction so he can understand better how to use it to its maximum potential;
- Prototyping bypasses information requirement;
- The time between two iterations is short;
- Usually it involves low costs.

Among the disadvantages, we can mention:

- Changing requirements;
- May not have thorough understanding of benefits and costs;
- Poorly tested;
- Dependencies, security, and safety may be ignored;
- High uncertainty;
- Problem may get lost;
- The quality may not be too high;
- It can involve higher costs due to multiple productions.

5. CONCLUSIONS

The development approach of DSS system should be strongly iterative. This will allow for the application to be changed and redesigned at various intervals. The initial problem is used to design the system which can subsequently be tested and revised to ensure the desired outcome is achieved.

Even if the sequential and structured nature of the life cycle design is one of its primary strengths, “in practice, a more iterative, bottom-up design approach might work better” (Marakas [11]).

The classical designing method is usually used due to its primary advantage related to the structure and discipline of the method. However, because of its low interactivity and not clearly defined user place and role, in DSS designing, the iterative method is preferred. The iterative method has the advantages of the flexibility, the user involvement, and the continuous evaluation.

REFERENCES

[1] Filip, F.G., Decision support and control for large-scale complex systems. Annual Reviews in Control, 2008;