**Structured systems analysis and design method (SSADM)**

**Introduction:**

Structured systems analysis and design method was designed by Learmonth Burchett Management Systems (LBMS) in cooperation with Central Computer Telecommunications Agency (CCTA) in 1980.

SSADM is an unambiguous approach in the form of methodology. It is composed by system surveys, structured analysis, structured designs, hardware studies, implementations and maintenance. Basically, it uses symbols instead of long descriptions, creating a graphic model of the system.

**Implementation of the SSADM on software development:**

It was developed as a standard for developing British database projects in order to standardise all software projects that were being designed across all government departments. It is an open methodology based on the waterfall model. It has been used by many commercial businesses, consultants, educational establishments and CASE (Computer-aided software engineering) tool developers.

The implementation of this methodology consists of five elements:

1. System Acquisition – The actual purchase of goods and services
2. Programming – Writing of the instructions that will be read and executed by the computers
3. Testing – Tests on various coded pieces of a design and correction of the errors
4. Conversion – The gentle changeover from the old system to the new one
5. Documentation – Writing the way the system and its functions were designed

**Values:**

* Improve project management and control
* Make more effective use of experienced and inexperienced development staff
* Develop better quality systems
* Make projects resilient to the loss of staff
* Enable projects to be supported by computer-based tools such as computer-aided software engineering systems
* Establish a framework for good communications between participants in a project

**Principles:**

SSADN follows the waterfall life cycle model starting from the feasibility study to the physical design stage of development.

It divides the cycle of development of the project in steps that are followed in sequence.

It is also characterized by intensive user involvement in the requirements analysis stage.

It revolves around these main modules:

* Logical Data Modelling: This involves the process of identifying, modelling and documenting data as a part of system requirements gathering. The data are classified further into entities and relationships
* Data Flow Modelling: This involves tracking the data flow in an information system. It clearly analyses the processes, data stores, external entities and data movement
* Entity Behaviour Modelling: This involves identifying and documenting the events influencing each entity and the sequence in which these events happen

Its guiding principles are five and can be described as:

1. The development of software is based on a number of phases
2. All these phases are arranged in a sequence beforehand their start
3. The sequence of phases represents the passage through time of the development of software
4. While previous phases are revisited, other phases overlap once new information becomes available
5. As the phases are being followed, the software becomes more useful, but all more complex

**Practices:**

* Dividing a project into small modules with well-defined objectives
* Diagrammatic representation and other useful modelling techniques
* Outline really well beforehand what is it that is to be built
* Before even starting to write code, consider and determine how to map all the requirements to a software environment
* Build the software according to the designs
* In the end prove that the software delivers exactly what was required

**Tools:**

SSADM is characterized by having a huge range of tools that can be utilized to make lighter the work burden of the software developers.

Most of the tools that are used are used to give the staff a higher understanding and better perception of the whole system, making it easier to find possible mistakes made during the decision module. Also, the tools must be free of unnecessary details also simplifying the overview of the software system.

Some examples of these tools are:

* The use of symbols instead of narrative descriptions
* Data Flow Diagrams (DFD)
* Data Dictionaries (collection of descriptions of the data objects or items)
* Structured English (pseudocode)
* Decision Trees
* Decision Tables
* Context Diagrams (defines the boundary between part of the system and its environment)
* Entity Relationship Diagram (for example, UML)

**Activities:**

There are seven core stages on the SSADM, which include.

1. Determining feasibility
2. Investigating the current environment – current systems and problems/requirements
3. Determining business systems options
4. Defining requirements - specification
5. Determining technical system options
6. Creating the logical design
7. Creating the physical design

Each of these phases applies specific techniques and analysis sequences.

**Key pros and cons:**

SSADM is not completely different from all the other methods. It also has its flaws. Nonetheless, in this case, the number of advantages is clearly superior to the number of disadvantages.

One of the pros to the SSADM is the good documentation provided by the fact this method is well structured, supplying both users and programmers really useful information to understand the use of the software. The fact that SSADM is a structured method also creates another advantage which is the standardization of software, offering very little scope to individual approach. In order to provide an even better understanding of the system, this method uses the modularisation, which means it is divided in smaller modules. Finally, since this method is user oriented and has a logical design, the probability of rejection from the hardware or from the users itself after the system being implemented is really low.

However, as previously said in this topic, there are also some drawbacks to this method. The rigidity of the SSADM, although being one of the reasons why it became the standard, can also lead to some difficulties. Since it is built on the analysis of data, if some of it changes after the method has already been applied, the system design will highly likely be incorrect. Another problem, and probably the biggest one, with the SSADM is the fact that it takes a huge amount of time just to analyse the whole project, creating a big gap in time between the creation of the idea and the delivery of the software system itself. At last, but not least, the complexity of the method, may force companies to spend a considerable amount of money in the training of the employees for them to use it.

**Pros and cons in a nutshell:**

**Pros:**

1. Good Documentation
2. Standardization
3. Modularisation
4. User Oriented
5. Logical Design

**Cons:**

1. Sometimes can be way too rigid
2. Time-consuming
3. Possibly expensive

**References:**

<https://www.techopedia.com/definition/3983/structured-systems-analysis-and-design-method-ssadm>

<https://searchsoftwarequality.techtarget.com/definition/SSADM>

<http://www.informatik.uni-bremen.de/gdpa/methods/m-ssadm.htm#TABULAR>

<https://www.researchgate.net/publication/266797414?channel=doi&linkId=543c6a720cf24ef33b762745&showFulltext=true>

<https://slideplayer.com/slide/4332856/>

<http://adataanalyst.com/it-strategy/structured-systems-analysis-and-design-method/>

<https://bizfluent.com/list-6781448-advantages-disadvantages-ssadm.html>