

## **STRUCTURED SYSTEMS ANALYSIS & DESIGN METHOD (SSADM)**

SSADM is a widely-used computer application development method in the UK, where its use is often specified as a requirement for government computing projects. SSADM is in the public domain, and is formally specified in British Standard BS7738.

SSADM divides an application development project into modules, stages, steps, and tasks, and provides a framework for describing projects in a fashion suited to managing the project.

SSADM's objectives are to:

- Improve project management & 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

SSADM covers those aspects of the life-cycle of a system from the feasibility study stage to the production of a physical design; it is generally used in conjunction with other methods, such as PRINCE, which is concerned with the broader aspects of project management.

*(PRjects IN a Controlled Environment (PRINCE2) is a process-based project management methodology that aims at improving the organization, management, and control of the project phases from beginning to end. The approach in PRINCE2 with agile project management is outcome-based which means that more emphasis lies on the end result rather than planning out the project meticulously.)*

In detail, SSADM sets out a cascade or waterfall view of systems development, in which there are a series of steps, each of which leads to the next step.

### **SSADM Techniques**

The three most important techniques that are used in SSADM are as follows:

#### *Logical data modeling*

The process of identifying, modeling and documenting the data requirements of the system being designed. The result is a data model containing entities (things about which a business needs to record information), attributes (facts about the entities) and relationships (associations between the entities).

### *Data Flow Modeling*

The process of identifying, modeling and documenting how data moves around an information system. Data Flow Modeling examines processes (activities that transform data from one form to another), data stores (the holding areas for data), external entities (what sends data into a system or receives data from a system), and data flows (routes by which data can flow).

### *Entity Event Modeling*

A two-stranded process: Entity Behavior Modeling, identifying, modeling and documenting the events that affect each entity and the sequence (or life history) in which these events occur, and Event Modeling, designing for each event the process to coordinate entity life histories.

### **Stages of SSADM**

The SSADM method involves the application of a sequence of analysis, documentation and design tasks concerned with the following:

#### *Stage 0 – Feasibility study*

In order to determine whether or not a given project is feasible, there must be some form of investigation into the goals and implications of the project. For very small-scale projects this may not be necessary at all as the scope of the project is easily understood.

In larger projects, the feasibility may be done but in an informal sense, either because there is no time for a formal study or because the project is a “must-have” and will have to be done one way or the other.

When a feasibility study is carried out, there are four main areas of consideration:

- Technical – is the project technically possible?
- Financial – can the business afford to carry out the project?
- Organizational – will the new system be compatible with existing practices?
- Ethical – is the impact of the new system socially acceptable?

To answer these questions, the feasibility study is effectively a condensed version of a fully blown systems analysis and design. The requirements and users are analyzed to some extent, some business options are drawn up and even some details of the technical implementation.

The product of this stage is a formal feasibility study document. SSADM specifies the sections that the study should contain including any preliminary models that have been constructed and also details of rejected options and the reasons for their rejection.

### *Stage 1 – Investigation of the current environment*

The developers of SSADM understood that in almost all cases there is some form of current system even if it is entirely composed of people and paper. Through a combination of interviewing employees, circulating questionnaires, observations and existing documentation, the analyst comes to full understanding of the system as it is at the start of the project. This serves many purposes.

### *Stage 2 – Business system options*

Having investigated the current system, the analyst must decide on the overall design of the new system. To do this, he or she, using the outputs of the previous stage, develops a set of business system options. These are different ways in which the new system could be produced varying from doing nothing to throwing out the old system entirely and building an entirely new one. The analyst may hold a brainstorming session so that as many and various ideas as possible are generated.

The ideas are then collected to options which are presented to the user. The options consider the following:

- the degree of automation
- the boundary between the system and the users
- the distribution of the system, for example, is it centralized to one office or spread out across several?
- cost/benefit
- impact of the new system

Where necessary, the option will be documented with a logical data structure and a level 1 data-flow diagram.

The users and analyst together choose a single business option. This may be one of the ones already defined or may be a synthesis of different aspects of the existing options. The output of

this stage is the single selected business option together with all the outputs of the feasibility stage.

### *Stage 3 – Requirement's specification*

This is probably the most complex stage in SSADM. Using the requirements developed in stage 1 and working within the framework of the selected business option, the analyst must develop a full logical specification of what the new system must do.

The specification must be free from error, ambiguity and inconsistency. By logical, we mean that the specification does not say how the system will be implemented but rather describes what the system will do.

To produce the logical specification, the analyst builds the required logical models for both the data-flow diagrams (DFDs) and the Logical Data Model (LDM), consisting of the Logical Data Structure (referred to in other methods as entity relationship diagrams) and full descriptions of the data and its relationships. These are used to produce function definitions of every function which the users will require of the system, Entity Life-Histories (ELHs) which describe all events through the life of an entity, and Effect Correspondence Diagrams (ECDs) which describe how each event interacts with all relevant entities. These are continually matched against the requirements and where necessary, the requirements are added to and completed.

The product of this stage is a complete requirements specification document which is made up of:

- the updated data catalogue
- the updated requirements catalogue
- the processing specification which in turn is made up of
  - user role/function matrix
  - function definitions
  - required logical data model
  - entity life-histories
  - effect correspondence diagrams

### *Stage 4 – Technical system options*

This stage is the first towards a physical implementation of the new system. Like the Business System Options, in this stage a large number of options for the implementation of the new system are generated. This is narrowed down to two or three to present to the user from which the final option is chosen or synthesized.

However, the considerations are quite different being:

- the hardware architectures
- the software to use
- the cost of the implementation
- the staffing required
- the physical limitations such as a space occupied by the system
- the distribution including any networks which that may require
- the overall format of the human computer interface

All of these aspects must also conform to any constraints imposed by the business such as available money and standardization of hardware and software.

The output of this stage is a chosen technical system option.

### *Stage 5 – Logical design*

Though the previous level specifies details of the implementation, the outputs of this stage are implementation-independent and concentrate on the requirements for the human computer interface. The logical design specifies the main methods of interaction in terms of menu structures and command structures.

One area of activity is the definition of the user dialogues. These are the main interfaces with which the users will interact with the system. Other activities are concerned with analyzing both the effects of events in updating the system and the need to make inquiries about the data on the system. Both of these use the events, function descriptions and effect correspondence diagrams produced in stage 3 to determine precisely how to update and read data in a consistent and secure way.

The product of this stage is the logical design which is made up of:

- Data catalogue
- Required logical data structure

- Logical process model – includes dialogues and model for the update and inquiry processes
- Stress & Bending moment.

### *Stage 6 – Physical design*

This is the final stage where all the logical specifications of the system are converted to descriptions of the system in terms of real hardware and software. This is a very technical stage and a simple *overview* is presented here.

The logical data structure is converted into a physical architecture in terms of database structures. The exact structure of the functions and how they are implemented is specified. The physical data structure is optimized where necessary to meet size and performance requirements. The product is a complete Physical Design which could tell software engineers how to build the system in specific details of hardware and software and to the appropriate standards.