Proceedings of the 8th International Conference on Applied Informatics
Eger, Hungary, January 27–30, 2010. Vol. 2. pp. 339–342.

The Evergreen SSADM

Attila Kuki

University of Debrecen
e-mail: kuki@inf.unideb.hu

Abstract

This paper gives an overview of the traditional system engineering methodology SSADM. There will be presented a comparison with other methods and tools, eg. RUP and UML. Some environments are highlighted where SSADM could efficiently be used nowdays, as well.

The paper includes a presentation about CASE tools which simplify the work with SSADM graphs.

Keywords: System Engineering , methodologies, SSADM

1. History of SSADM

This paper is devoted to the structured system engineering methodology SSADM. The word SSADM stands for Structured System Analysis and Design Method, and was developed in England in the 80s. For the first let us overview the brief history of development of SSADM.

- 1980 - Central Computer and Telecommunications Agency (CCTA) evaluates the method
- 1983 - SSADM made mandatory for all new information system developments
- 1984 - Version 2 of SSADM released
- 1986 - Version 3 of SSADM released, adopted by NCC.
- 1988 - SSADM Certificate of Proficiency launched, SSADM promoted as 'open' standard
- 1990 - SSADM 4 launched
- 1993 - SSADM V4 Standard and Tools Conformance Scheme Launched
- 1995 - SSADM V4+ announced, V4.2 launched

The SSADM is a life cycle based so called 'Waterfall' method, which assumes that

- All requirements are identified at the start
- All the system is analysed
- All the system is designed
- All the system is written

- All the system is tested
- All the system is handed over to the client
- So there is only one run through the life cycle

This 'Waterfall' approach has a lot of advantages, eg. known requirements are all documented, analysis can cover all required areas, design can be optimised as all known requirements are included, coding and testing can be carried out efficiently as all areas to be covered are known, project management is easier as tasks required are known - timetables and staffing can be planned and controlled.

Of course, this model has some disadvantages, too: The client might consider of important requirements later in the project. The client's requirements might be changed during the project. A step in the project might take longer than expected. The client might not like what the developers have produced. The system handed over to the client might produce wrong things.

The general concept of a 'Waterfall' model can be seen on the following figure:

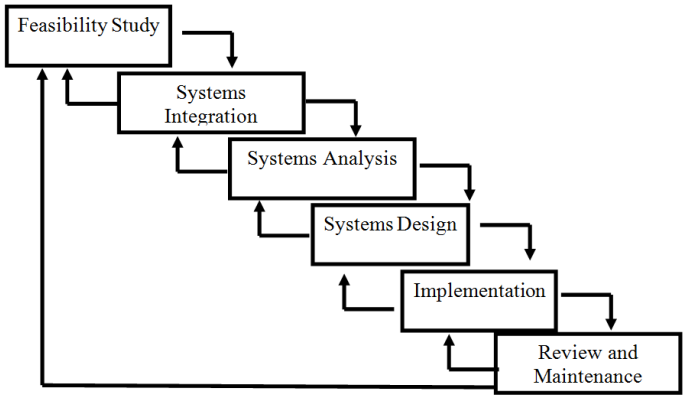

Figure 1

In history of SSADM there was a change in philosophy of covering the life cycle. The first versions of SSADM cover the whole life cycle, while version 4 covers only up to physical design. The main reason of this was the following: the SSADM as a standard system engineering methodology is not adequate to give general instructions for environment-dependent stages, eg. implementation, maintenance.

2. Other methods

The application environments and its information systems become more complicated and for such systems the SSADM does not proofed efficient enough. New methods should have been developed. One characteristic example is the Rational Unified Process, which was created by the Rational Software Corporation, a division of IBM since 2003.

Beside the RUP is a single concrete prescriptive process, it is an adaptable system development framework, as well. The following list contains the key aspects

of RUP:

- Risk-driven process
  - Risk management integrated into the development process
  - Iterations are planned based on high priority risks
- Use-case driven development
  - Use cases express requirements on the system's functionality and model the business as context for the system
  - Use cases are defined for the intended system and are used as the basis of the entire development process
- Architecture-centric design
  - Architecture is the primary artifact to conceptualize, construct, manage, and evolve the system
  - Consists of multiple, coordinated views (or models) of the architecture

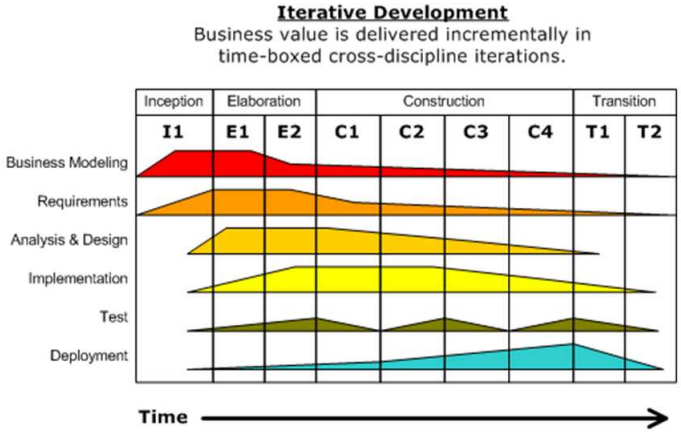

Figure 2: The Rational Unified Process

Beside the RUP there exist a lot of other iterative methodologies, eg. incremental methods, spiral methods. These methods contain numerous iterations of the system life cycle and they serve one of the most important application fields of SSADM: It can be efficiently used to evaluate these life cycle iterations.

3. SSADM and UML

At present days still exist numerous projects for which different system engineering methods can be applied. In case of relatively small projects the use of SSADM could be still more efficient and adequate than other 'complex' methodologies. The designer team has to collect the characteristics of the application system or the information system to be developed, and based on these facts a decision should be made on the way of building the product.

In the following several point of views will be presented, which describe the behavior of SSADM and UML.

- Data Management
  - Logical Data Models - SSADM
  - Class Diagrams - UML
- Event Management
  - Entity Life History - SSADM
  - Behavior, Interaction Diagram - UML
- Process Management
  - Data Flow Diagram - SSADM
  - The Activity Diagram - UML
- Resources
  - Requirement Catalogue - SSADM
  - Modeled by using the stereotype feature - UML
- Interfaces
  - Dialogue Design - SSADM
  - Modeled in class and component diagrams - UML
- Quality Management
  - Requirements Catalogue - SSADM
  - In analysis explorative prototypes and in design
- Business Issues
  - Data Flow Diagrams, Entity Life History Diagrams - SSADM
  - Activity diagrams describe and model business process - UML
- User involvement
  - Gathering information about system, reviewing products of each stage - SSADM
  - Gathering information about system in use case models, CRC and technical dictionary, review/check prototypes - UML

At present days some efforts were made to investigate the possibility of evolving a system developed by SSADM to fit the UML and XML. Evolving SSADM to apply Unified Modeling Language and XML could be very important. Large systems developed with SSADM in the 80's can be reengineered by modern tools. This importance is expressed in UML 2.0. This version supports SSADM while the previous ones not.

References

[1] Hongji Yang Software Evolution with UML and XML, Idea Group Publishing, (2005)

[2] Nick Lambrou, Matthew Walkley, Philip L. Weaver Practical Business Systems Development Using Ssadm: A Complete Tutorial Guide, Financial Times Management; (2002)

[3] Per Kroll, Philippe Kruchten The Rational Unified Process Made Easy: A Practitioner's Guide to the RUP, Addison-Wesley Professional (2003)