

# A Comparison of Various Software Development Methodologies: Feasibility and Methods of Integration

<https://doi.org/10.3991/ijes.v4i4.6547>

M. Samir Abou El-Seoud

The British University in Egypt - BUE  
Faculty of Informatics and Computer Science, Cairo, Egypt

**Abstract**— System development methodologies which have been used in the academic and commercial environments during last two decades have advantages and disadvantages. Researchers had tried to identify objectives, scope ...etc. of the methodologies by following different approaches. Each approach has its Limitation, specific interest, coverage ...etc. In this paper, we tried to perform a comparative study of those methodologies which are popular and commonly used in banking and commercial environment. We tried in our study to determine objectives, scope, tools and other features of the methodologies. We also, tried to determine how and to what extent the methodologies incorporate the facilities such as project management, cost benefit analysis, documentation ...etc. One of the most important aspects of our study was how to integrate the methodologies and develop a global methodology which covers the complete span of the software development life cycle? A prototype system which integrates the selected methodologies has been developed. The developed system helps analysts and designers how to choose suitable tools or to obtain guidelines on what to do in a particular situation. The prototype system has been tested during the development of a software for an ATM “Auto Teller Machine” by selecting and applying SASD methodology during software development. This resulted in the development of high quality and well documented software system.

**Index Terms**— ATM, SASAD, JSD, SSADM, SDLC

## I. INTRODUCTION

In the past several people have tried to analyze and compare the software development methodologies e.g. SASD, JSD, SSADM ...etc. with different objectives. In order to do the studies on software development methodologies the following approaches can be adopted:

1. Developing a list of model questions and applying it on selected methodologies to identify and study their characteristics and features. Several different criteria were selected to study these methodologies in order to determine their suitability. One of these characteristics was to study the Ada compatibility, i.e. how Ada might be used by these methodologies and how successfully these methodologies can solve problems related to Ada environments [1]. During 1983, Wassermann and his group conducted a similar survey of 24 software development methodologies [2]. One of the purposes of this study was to determine which methodologies could be integrated with the Ada language and related Ada programming support environment [3]. A similar approach to study the methodologies was also adopted by IFIP, WG 81 during July 1983 Olle et al. [4]. The comparisons and analysis of these methodologies were done against a number of features and on the basis of the information about their goals, applicability, origin, experience, etc. As a result of the comparison and analysis, they found that the information about the methodologies cannot be guaranteed and there are some weak points for each methodology. A similar study based on feature analysis was also made by Livari and his group [5]. Features like scope and content, support and usability, assumption, etc. were studied.
2. Using the methodologies to solve a model problem related to a particular environment and then comparing them to see how successfully the features and facilities of these methodologies can be applied to solve that problem. This approach was adopted by a large research group in the early eighties [1]. This group created two model problems for partial development of the systems using the selected methodologies. The first one was an aircraft monitoring system and the second was to develop a given database by using VDM, (Software development ideals given by Jones) [6]. The output systems for the above examples could not be implemented due to lack of access to suitable compilers [7].
3. Selecting a real life problem and studying how best this problem can be solved by a specific set of methodologies.
4. Developing a set of questions and apply them to the individual methodologies which are popular in any specific environment e.g. banking and commercial environment to determine their objectives, scope, characteristics, features, etc. Further use the phases of the life cycle model [8] as standard activities to examine, determine and evaluate the extent to which each methodology covers the phases of the life cycle i.e. the span of systems development tasks. This approach is a modification of the first approach and was also used in this study.
5. Defining what is a good methodology for a particular environment, such as banking and commercial environment; and comparing the methodologies which are commonly used in banking and commercial environment against the good one. This is second approach which was used in our study.

The Object Modeling Technique (OMT) and SA/SD methodologies both incorporate similar modeling components. Both methodologies support three orthogonal views of a system- the object, dynamic, and functional models. The OMT and SA/SD methodologies differ in the relative emphasis that they place on the various modeling components. OMT designs are dominated by the object model. The real-world paradigm of objects and relationships provides the context for understanding dynamic and functional behavior. In contrast, SA/SD stresses functional decomposition. A system viewed primarily as providing one or more functions to the end user [9].

SA/SD organizes a system around procedures. In contrast, an object-oriented design technique (such as OMT) organizes a system around real-word objects, or conceptual objects that exist in the user's view of the world. Most changes in requirements are changes in function rather than in the objects, so change can be disastrous to procedure-based design. By contrast, changes in function are readily accommodated in an object-oriented design by adding or changing operations, leaving the basic object structure unchanged. SA/SD is useful for problems where functions are more important and complex than data. SA/SD assumes that this often occurs [9].

#### A. Scope of the Study

The scope of our study is to do a comparative study of the methodologies which are popular and commonly used in banking and commercial environment.

#### B. Objectives of the research paper

Our study of the system development methodologies was designed to achieve the following objectives:

1. To determine the objective, scope, tools, output and other major features of the methodologies.
2. To determine the strong and weak points of the methodologies.
3. To determine how and to what extent the methodologies incorporate the facilities such as project management, cost benefit analysis, risk analysis, documentation, enhancement, maintenance, prototyping, quality plan of a system,...etc.
4. To develop a conceptual methodology i.e. a framework which can be used as a standard methodology or a yardstick for the purpose of comparison of feature of different methodologies.

In order to do the study the following three methodologies were selected:

1. Structured Analysis and Structured Design methodology (SASD).
2. Jackson System design methodology (JSD).
3. Structured system Analysis and Design methodology (SSADM).

The criteria for the selection were that these methodologies are the ones which are being introduced in the commercial, banking and government areas of Gulf Cooperative Countries (GCC). Further these methodologies are being taught in most universities of USA, EEC, GCC countries, and in fact one of these methodologies i.e. SSADM is mandatory to be followed by all the government projects in EEC countries.

## II. RESULTS OF THE PROJECT

Out study came out with several important results which can be summarized in the following:

There are several problems which are common in each of the selected methodologies. The major ones are listed in the following:

#### A. Problems with SASD methodology

**SASD** methodology does not describe how to do the activities such as project management, cost estimation, cost benefit analysis, controlling the time, project strategy, validation & verification, testing and how to apply quality metrics and quality assurance.

#### B. Problems with JSD methodology

The major deficiency in **JSD** is that it neglects certain important phases/activities of software development. For example, **JSD** methodology does not cover all the aspects of analysis, it excludes some analysis tasks e.g. project proposal, analysis of environment, requirement analysis, feasibility study, cost benefit analysis...etc.

**JSD** can be applied efficiently only with problems whose subject matter has a strong time ordering, namely sequential processes and it cannot be applied efficiently on real time systems e.g. air craft monitoring system and scientific systems.

#### C. Problems with SSADM methodology

**SSADM** does not include stages/phases for investigation and implementation. Also **SSADM** excludes some software development activities e.g. controlling time, project strategy ...etc.

There are several problems which are common in almost all these methodologies. The major ones are in the following:

During our study it was observed that all these methodologies did not specify clearly and explicitly their objectives, scope, phases, definitions of terminologies, used. Scope of these methodologies is limited to some extent. In other words, we can say that they are appropriate for specific problems, activities and environments.

**SASD**, **SSADM** and **JSD** methodologies do not have any tool or mechanism by which they can verify that the outputs are correct or that nothing is missing. However, they have some mechanism to check the output such as cross-referencing in **SASD**; comparisons of the deliverables of one step with the deliverables of other steps in **SSADM**, but these are not checked in real situation.

Guidelines on how to use these methodologies are explicitly and clearly defined and can help the developer during the development process.

**SASD**, **JSD** and **SSADM** methodologies do not have any tool or mechanism which can be used to handle the administrative problems which may arise during the software development life cycle. However, in the case of **SASD** and **JSD**, one may say that they have some approaches which may deal with such situations. For example, if the project proposal is clearly specified in **SASD** methodology it can be used as a yardstick where there is conflict between the analysts/designers or management/user. In **JSD** methodology the system specifications which are developed at the later stage by the analysts/designers can be used for checking in case of conflict

with user management; but this is not a perfect solution. This is because **JSD** specifications are written just to solve the technical aspects, not the administrative ones.

One of the results was that the selected methodologies do not cover the whole span of the life cycle. In such cases according to our definition of methodology, the above mentioned methodologies, namely **SASD**, **SSADM** and **JSD** are considered a subset of our definition of model methodology. For example, **JSD** and **SASD** ignore the tasks concerned with the Project Proposal, Analysis and Transition Phases. **SASD** also does not cover the cost benefit analysis whereas **SSADM** does not cover the Transition Phase. Also, we come to know that none of the methodologies have covered the strategy phase defined in the life cycle.

An important result of this research work showed that there is not methodology which is commonly used and appropriate to all different types of problems and environment. In our study it was observed that it is very difficult and complex to study and understand various information systems development methodologies. This is because of their wide range and the large number of tools/techniques, overlapping terminologies, phases/steps to understand. We therefore, developed explanatory notes and step by step information on how and when to use a particular phase, tool/technique of a particular methodology. This was a difficult job since it involved understanding and experience of using a tool/technique or a method. Also, we want to mention that we covered some very important topics in our research which were not covered in any of the previous studies. These topics are testing , quality assurance, project management, cost analysis, cost benefit analysis and risk analysis.

These results are also summarized in, tables I and II.

### III. WATERFALL MODEL

David Whitgift points out that in the earliest days of software development, code was written and then debugged [12]. There was no formal design or analysis. This code and debug approach rapidly became less than optimal as complex software systems were required. Since the approach to developing complex hardware systems was well understood, it provided a model for developing software.

This model, known as the *waterfall*, is an approach to development that emphasizes completing a phase of the development before proceeding to the next phase. In conjunction with certain phase completions, a baseline is established that "freezes" the products of the development at that point. If a need is identified to change these products, a formal change process is followed to make the change. The graphic representation of these phases in software development resembles the downward flow of a waterfall.

Many software development methodologies have evolved from attempts to optimize the waterfall model for software. For example, software prototyping helps provide the complete understanding of the requirements that is typical of hardware production--which understanding is critical to the waterfall model.

### IV. NEED FOR A GLOBAL METHODOLOGY

It is not possible for an organization to reply on a single methodology which is available, particularly if the organi-

TABLE I.  
CERTAIN IMPORTANT ACTIVITIES COVERED BY THE METHODOLOGIES

Activities	Methodologies		
	JSD	SASD	SSADM
Project MANAGEMENT	No	No	YES
Verification & VALIDATION	YES	No	YES
Enhancement	Yes	Yes	Yes
Maintenance	Yes	Yes	No
Quality Plan/Metrics	Yes	Yes	No
Prototyping	No	No	No

TABLE II.  
EVALUATION OF THE TECHNICAL FEATURES OF THE METHODOLOGY

Criteria	Methodologies		
	SD	SASD	SSADM
Supports Prototy P I Nc	No	No	No
Involvement Of User/Management	No	No	Yes
Assists In Project Manag Em Ent	No	No	Yes
Communication Facilities Between Analyst/Designer & User/Management	Not Any Specif One	Not Any Specific One	Not Specific One
Quality Of Documentation Facilities	Good	Good	Good
Applies Of Quality Plan Or Quality Metrics	No	Not Significantly	No
Suitable To The Class Of Problem	Inherently Sequential or Time Dimension	Commercial & Industrial	Wide Class of Problems
Uniqueness	Develops Model of the Real World and Later Super-Imposes Functions	Concept of Cou-pling & Cohesion.... etc.	Data Driven: Data Flows, Data Structures and Events in Parallel When Developing a System

zation has tasks of multi-nature. For example, the software system of an Air-Lines company may perform tasks of different nature, i.e. Simulation, Booking and Reservation, Payroll, Real time system to control traffic ...etc. A rigid approach in the software development process cannot be successful. This is because tools/techniques and methods suitable for one type of problem in an organization cannot be suitable for other types within the same organization. The search for perfect methodology which can do all different nature of tasks is a dream or an illusion.

We therefore, felt that it is necessary to have a methodology of global nature i.e. a methodology which should cover all the aspects of software development life cycle model. This methodology should be able to be used in all different environments and problem areas. This methodology must have a provision to incorporate early changes in business policies or user requirements during the phase of system development, if there are any, i.e. it should be able to handle effectively the changes made. The methodology should be able to handle software related questions and matters relating to people and business functions. This methodology should also be able to handle the issues of formal correctness, reliability, project management and human factors. The methodology must be able to deal

with the problems related to the relationship between an organization, analyst, designer and its environment. The new methodology should have human aspects of systems development activities which have not been considered by any of the methodologies.

The Global methodology should be people and as well as process/data oriented. By people oriented methodology we mean that it should be able to handle the socio-technical problems which may be created between users, management and analysts, designers because of conflict between administrative and personal interests. It should be able to identify the fundamental causes of such problems and give possible socio-technical solutions.

This methodology will be a blend of data analysis and structured analysis approaches based on analysis of human activities and behavior of user, management, data analysis/process analysis and structured analysis and object oriented methodologies a new concept in software development. The combined methodology must be able to handle different types of problems and offer their solutions which may satisfy the user, management.

On the basis of the observation and results obtained during the analysis of comparison of the methodologies it was felt that efforts should be made to determine the feasibility and possible ways to integrate the three mentioned methodologies i.e. **SASD**, **JSD** and **SSADM**. The output of integration may provide a flexible integrated methodology which will be of multi-purpose having various techniques and tools suitable to different problems; we come out with a plan for a flexible integrated methodology.

This prototype of the global methodology is suitable for developing information system of various natures. We hope it will help information systems developers by guiding them to which methodology is suitable for their problem and further how to use it. In short, different methodologies and their tools may be used for different situations, sometimes the tools of more than one methodology may be applied during the software development project. However, the selection of a methodology or tools may not be an easy task. It will depend on the level of experience and skill of the developers and also the number of manpower available.

In our project a prototype has been developed which integrates **JSD**, **SSADM** and **SASD** methodologies, as we mentioned earlier, given clear Input/output and showing the phases of the life cycle model. It also guides what to do at a particular phase or subphase the life cycle

This prototype of the global methodology has a framework which links all different methodologies. The analyst, designer may select any step or tool of a methodology which is a part of the integrated methodology after using the step or tool he may go back to any phase of the software development life cycle.

Using integrated methodology may be difficult may be difficult because of a large number of tools/techniques, different terminologies which are being used and overlapping the phases of the selected methodologies. This may need to have highly qualified and well experienced analyst, designer who has a good command in this selected methodology.

The output of the integrated methodology will be a system which will satisfy all the social requirements of humans, concerned with the related system by relying to a great extent on information technology.

The flexible approach to information systems development may be proved to be more useful than a best or good methodology which still does.

The prototype of the global integrated methodology has a framework which links all different methodologies. The analyst, designer may select any select any step or tool of a methodology which is a part of the integrated methodology after using the steps not exist.

## V. DEVELOPMENT OF ATM SOFTWARE

After a long discussion and careful thought we decided to use the **SASD** methodology, to solve a model problem in banking and commercial environment, because it is the most applicable methodology in this environment.

As an application, we developed **ATM** software for local bank by using **SASD** and development life cycle model (**SDLC**) which is described in (Shaikh, 1987); during the system development of software for **ATM**. Although this methodology is not the best one to be used. We did not use the **SSADM**, which is the better methodology for such types of problem and is also mandatory to be followed in all government projects of EEC countries. The main two reasons for not using **SSADM** are:

**SSADM** is comparatively a new methodology, there is not enough literature available which can explain the working details of **SSADM**. The only possible source was the introductory information provided by the developers **SSADM**. These people provide the detailed information and training on commercial bases.

**SSADM** is still under continues modifications i.e. in progress. It's old name was **LSDM** which has been changed after eliminating the drawbacks and shortcomings which were in other methodologies such as **SASD**, **JSD**, ..etc. There are several life cycle models available in the literatures and some models are the integral part of the methodologies. These methodologies generally cover certain aspects of the software development, i.e. do not cover the complete span of software development tasks. It was therefore decide to use **SDLC** during the development of software for **ATM**. This life cycle model covers all aspects of system development and it is suitable for all environments. This life cycle model starts from Project Proposal, i.e. users 'wants' not from 'requirements' {needs}. This model can be used by all different systems development approaches. The phases of the life cycle model are flexible which can be modified according to the environment and type of problem and has property of recursion.

The **ATM** project covered all the aspects of system development life cycle (**SDLC**). The project was professionally undertaken using the de-facto standard methodology, the Structured Analysis and Structured Design Methodology (**SASD**). **ORACLE** was chosen as the platform for the application development. The whole project was very well coordinated among the Sultan Qaboos University and a local bank in Oman during all the phases of development. This resulted in the development of a software system which is of high quality and very satisfactory to the users. All parts of the projects are systematically documented. The documentation is complete, self explanatory and very useful for any future augmentation or modification. It is highly expectable that the developed software will be in operation.

## VI. CONCLUSION

In this paper, determining objectives, scope, tools and other features of the system development methodologies had been studied. One of the most important aspects of our study was how to integrate the methodologies and develop a global methodology which covers the complete span of the software development life cycle? A prototype system which integrates the selected methodologies has been developed. The prototype system has been tested during the development of software for an ATM "Auto Teller Machine" by selecting and applying SASD methodology during software development. This resulted in the development of high quality and well documented software system.

## REFERENCES

- [1] Denvir et al. "Report on the study of an Ada Based System Development Methodology" Dept. of Industry UK, Sept 1981
- [2] Wasser A.I., Freeman P., "Ada Methodologies Concepts and Requirement", Software Engineering Notes Vol 8, Number 1, Jan. 1983, pp 33-50.
- [3] Procella M., Freeman P., Wassermann A.I, "Ada Methodology Questionnaires Summary", Software Engineering Notes Vol 8, Number 1, Jan. 1983, pp 51-98.
- [4] Olle, T. W., Sol, H.G., et al., "Information systems Design Methodologies" A Comparative Review, Proceeding of the IFIP WG. 8.1, The Netherlands, North Holland Publishing Company, 10-14 May 1982.
- [5] Iivari J., Kerola P., "A Sociocybernetic Frame Work for the feature Analysis of Information Systems Design Methodologies", Information Systems Design Methodologies, Proceeding of IFIP WG. 8.1, Working Conference on Feature Analysis of Information System Design Methodologies, York, UK, Elsevier Science publisher B.V., pp 87-139.
- [6] Iivari J., Kerola P., "A Sociocybernetic Frame Work for the feature Analysis of Information Systems Design Methodologies", Information Systems Design Methodologies, Proceeding of IFIP WG. 8.1, Working Conference on Feature Analysis of Information System Design Methodologies, York, UK, Elsevier Science publisher B.V., pp 87-139.
- [7] Jones C., "Software Development: A Rigorous Approach". Prentice Hall, 1980.
- [8] Olle, T. W., Sol, H.G., et al., "Information systems Design Methodologies", A feature Analysis, Proceeding of IFIA WG. 8.1, Working Conference on Feature Analysis of Information System Design Methodologies, July 1983.
- [9] Iivari J., Kerola P., "A Sociocybernetic Frame Work for the feature Analysis of Information Systems Design Methodologies", Information Systems Design Methodologies, Proceeding of IFIP WG. 8.1, Working Conference on Feature Analysis of Information System Design Methodologies, York, UK, Elsevier Science publisher B.V., pp 87-139.
- [10] Shaikh M.U., "The Summary of the Life Cycle Model", "Analysis and Comparison of Software Development Methodologies", Ph.D. thesis, University of Liverpool, U.K., 1987, page 103-105.
- [11] James Rumbaugh, et. al., "Object-Oriented Modeling and Design", Prentice-Hall International Inc., 1991.
- [12] Whitgift, David, Methods and Tools for Software Configuration Management, 1991.

## AUTHOR

**M. Samir Abou El-Seoud** is with the British University in Egypt (BUE), Faculty of Informatics and Computer Science, El-Sherouk City, Misr-Ismalia Road, Cairo, EGYPT. E-mail: samir.elseoud@bue.edu.eg.

Submitted 27 October 2016. Published as resubmitted by the author 03 December 2016.