

Comparison of SSADM and XP Using NIMSAD Framework

Part of my study in

Information System Methodologies

Wael Almadhoun

Table of Contents

Introduction.....	5
SSADM.....	6
SSADM Objectives:.....	6
Stages of SSADM:.....	7
Stage 0: Feasibility study:.....	7
Stage 1: Investigation of the current environment:.....	7
Stage 2: Business systems options:.....	7
Stage 3: Definition of requirements:.....	7
Stage 4: Technical system options:.....	7
Stage 5: Logical design:.....	7
Stage 5: Physical design:.....	8
XP.....	8
XP Lifecycle:.....	9
Planning:.....	9
Designing:.....	9
Coding:.....	9
Testing:.....	9
Listening:.....	9
NIMSAD.....	10
Problem Situation:.....	10
Intended Problem Solver:.....	10
Problem Solving Process (the methodology):.....	11

Comparison of SSADM and XP Methodologies	12
Problem Situation:.....	12
Intended Problem Solver:.....	12
Problem Solving Process:.....	13
1. Phase 1 - Problem formulation:	13
Stage 1 – Understanding the Situation of Concern.....	13
Stage 2- Performing the Diagnosis	13
Stage 3 – Defining the Prognoses Outline	14
Stage 4 – Defining the Problems	14
Stage 5 – Deriving Notional Systems	14
2. Phase 2 - Solution Design:	14
Stage 6 – Performing Conceptual / Logical Design	14
Stage 7 – Performing Physical Design	15
3. Phase 3 - Design implementation:	15
Stage 8 – Implementing the Designs	15
Evaluation	16
Conclusion	17
References	18

Table Of Figures

Figure 1 System Development Life Cycle – Waterfall (Community.mis.temple.edu)	6
Figure 2 Extreme Programming Project (Extremeprogramming.org)	8
Figure 3 The methodology user (slides by Jenny Coady).....	10
Figure 4 Evaluation (slides by Jenny Coady)	11

Introduction

In this report eventually the author aims to compare two different information system methodologies. I believe that moving with the sequence of the previous essay and presentation we have done – group assessment- in IS Methodologies course, I would obviously talk about SSADM (Structure System Analysis and Design Methodologies) and XP (Extreme Programming).

SSADM is a methodology applied in information systems to solve problems using a staging approach as waterfall and XP is a methodology anticipated to improve the software quality, customer satisfaction and more responsive to changes. The discussion about advantages, disadvantages and limitations of each methodology, my opinion based on past experience and the benefits of using each method will be explained.

For this comparison NIMSAM (Normative Information Model-based Systems Analysis and Design) framework is used to discuss each methodology strength and weaknesses area by going through the problem situation, who is the problem solver and what is the problem solving process.

Initially the author thoughts comparing methodologies would be pure academic reason, even why organizations want to compare methodologies? as they are comfortable with what they do and most of them have internal defined processes and expertise. But thinking from project prospective, people and technology constraint, you would eventually need to understand different methodologies scope, strength and weaknesses. The author believes doing so should help organizations make proper choice of the right information system method or a combination of both or some, which I usually practice on my career.

In the following section we will go through the outline of each method from the previous group assignment which the author has done during this course to explain the development process, the requirement capturing to help us understand how each method is dealt with problem situations and problem solving process.

SSADM

SSADM (Structured Systems Analysis and Design Method) it's a systematic approach for analysis and design of information systems based on the waterfall method. Is formally specified in British Standard BS7738. (Informatik.uni-bremen.de, 2002)

SSADM Objectives:

To improve project management. To effectively utilize the team expertise and development staff. Aims to develop better quality systems. Follows best practices and industry standard i.e. prince for project management, Enables good communications between participants, as it is a key success factor in any project. (Rouse, n.d.)

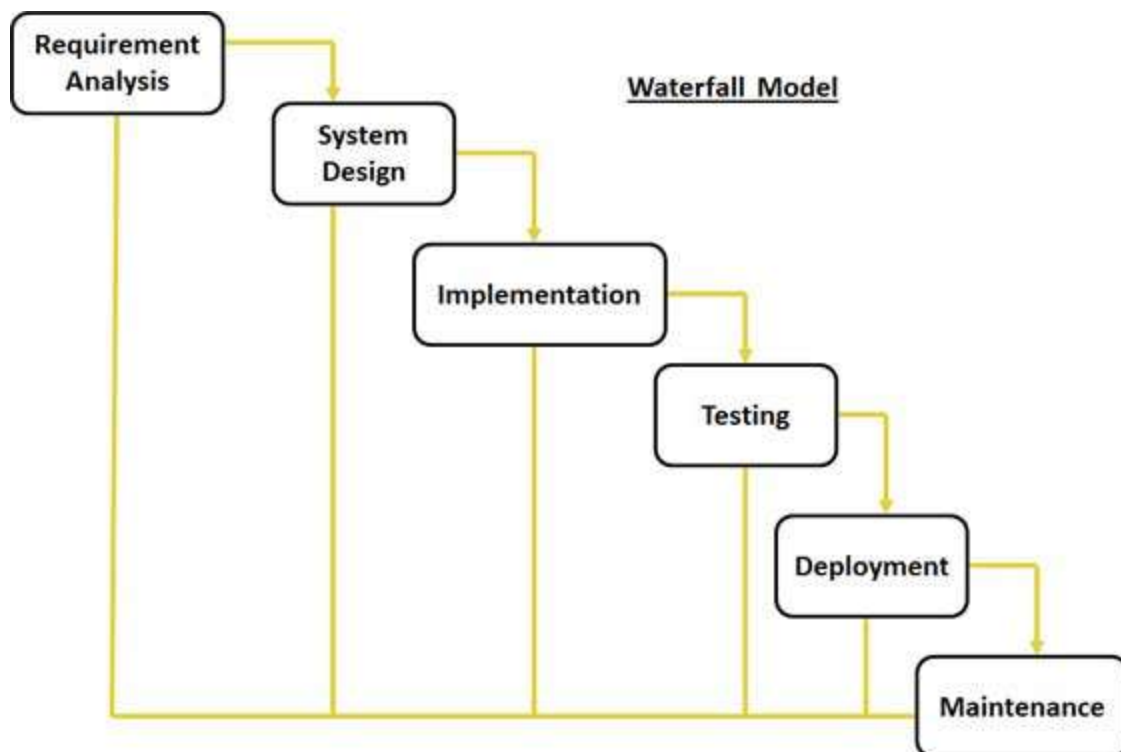


Figure 1 System Development Life Cycle – Waterfall (Community.mis.temple.edu)

Stages of SSADM:

Stage 0: Feasibility study:

The system analyst analyses the business requirements mentioned by the client and prepares the feasibility study before going ahead with the project which entails whether the proposed scheme is viable for the client and suitable to solve their problem. The feasibility study, should tackle the technical, financial, organizational and ethical aspects of the project.

Stage 1: Investigation of the current environment:

The analyst could use the data flow diagram (DFD) to investigating the current system. "It is the process of identifying, modeling and documenting how data moves around an information system"(Beal, n.d.).

Stage 2: Business systems options:

Once the current system is evaluated, the analyst should generate a list of system options, which will meet the user requirements along with financial and risk assessment.

Stage 3: Definition of requirements:

Upon selection of the appropriate business system option, the analyst should prepare a documentation to describe the new system and its specifications in details. In this stage, the analyst defines the systems processes and functions, user job specifications, and system objectives. The analyst should also consider the hardware, software and network requirements for the system. (McKay, n.d.).

Stage 4: Technical system options:

Number of options for the implementation of new system are generated towards a physical implementation of the new system. The options is narrowed down to minimum number of options usually two or three to present to the user from which the final option is chosen.

Stage 5: Logical design:

This stage is where the logical dialogue of the new system is defined, such as the logical processes behind each user defined task. The analyst designs the menu structures and user dialogues. The analyst also defines update and inquiry processes.

Stage 5: Physical design:

The logical and technical specifications are applied to real hardware and software. The functions, their description and how they are implemented is specified. The physical data structure is optimized to meet the performance and size requirements specified by the user.

XP

XP (Extreme Programming) methodology was introduced to control the quality of systems by taking the stages into extreme level. It's agile software development, XP methods depends on breaking the system into small parts to be delivered in iterations. This means, the team will have more control on the client requirements.

The main focus of XP is coding, development and software testing, which will be performed in each iteration to assure that the system meets the quality measures and the client needs.

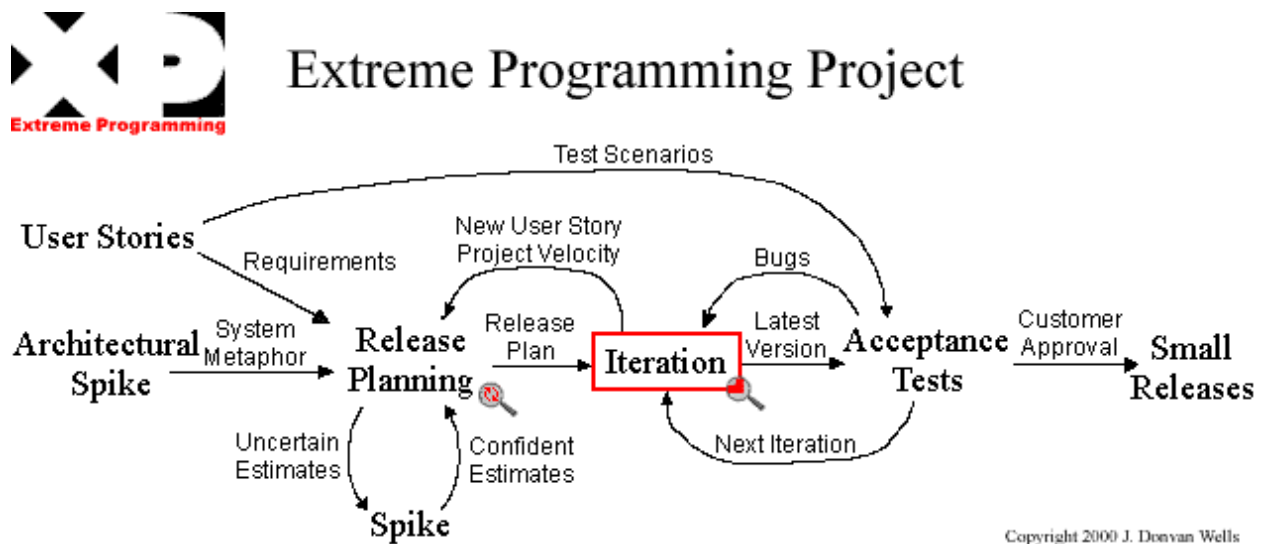


Figure 2. Extreme Programming Project (Extremeprogramming.org)

XP Lifecycle:

Planning:

The development team meets with the client to identify the user requirements and capture the system functional requirements which are converted into iterations. A combination of iterations give the customer an accumulative final practical product.

Designing:

Using systems metaphor or standards on names, class names and methods. Using Software Class Responsibilities (CRC) cards that allow for a departure from the traditional mindset and make possible object oriented technology. Creating spike solutions or simple programs that explore potential solutions for a specific problem, ignoring all other concerns, to mitigate risk” (Smak, 2012) (Group Assessment 4)

Coding:

This phase is the core part of the XP stages. XP prioritizes this part, the technical team and developers will be working towards the client needs and to make sure they are given a valuable product.

Testing:

This phase is important and goes side by side with the development phase (at the end of the iteration). The Users Acceptance Testing, which is based on the client demands after the completion of development, a demo will be presented to the client.

Listening:

The client feedback is the crucial part, on each iteration and during the development phase. If there's a new requirement, the team will make sure that the client feedback taken in consideration. This might be the basis of a new design. Then the whole process repeats itself on iterations.

NIMSAD

The author will use NIMSAD to comparing the above mentioned methodologies, so we should see a brief about NIMSAD framework and its elements. NIMSAD is a framework that is used to evaluate and compare methodologies. Methodology is defined as an explicit way of structuring thinking and action, involving both critical and creative thinking (Jayaratna, 1994, p. xi). NIMSAD framework has three elements to compare methodologies. The last step is to evaluate:

1. Problem Situation
2. Intended Problem Solver
3. Problem Solving Process

Problem Situation:

Problem situation involves a client. The framework defines distinction between action world and thinking world, with problem situation existing in the action world and problem solving taking place in both. (slides by Jenny Coady)

Intended Problem Solver:

The problem solver could be internal or external to the organisation and could be a consultant, systems analyst, etc. The focus is on the role, rather than the person. (slides by Jenny Coady)

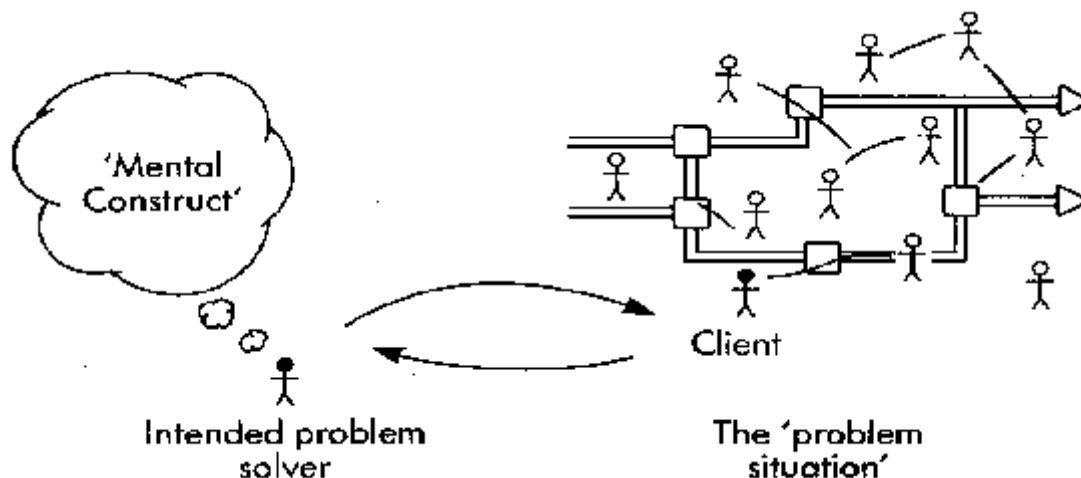


Figure 3 The methodology user (slides by Jenny Coady)

Problem Solving Process (the methodology):

Problem solving process is the main part of all the methodologies. The methodology in this stage can be seen as a structured process to assist in transformation from the current situation to the desired situation. NIMSAD defines 3 essential phases of this process, further broken down into 8 detailed stages that can be seen as applicable to any problem-solving process. (slides by Jenny Coady)

Evaluation is the most important part. The main idea of NIMSAD is to evaluate methodologies based on the previously mentioned elements. The author will use the practical experience in both selected methodology to evaluate them.

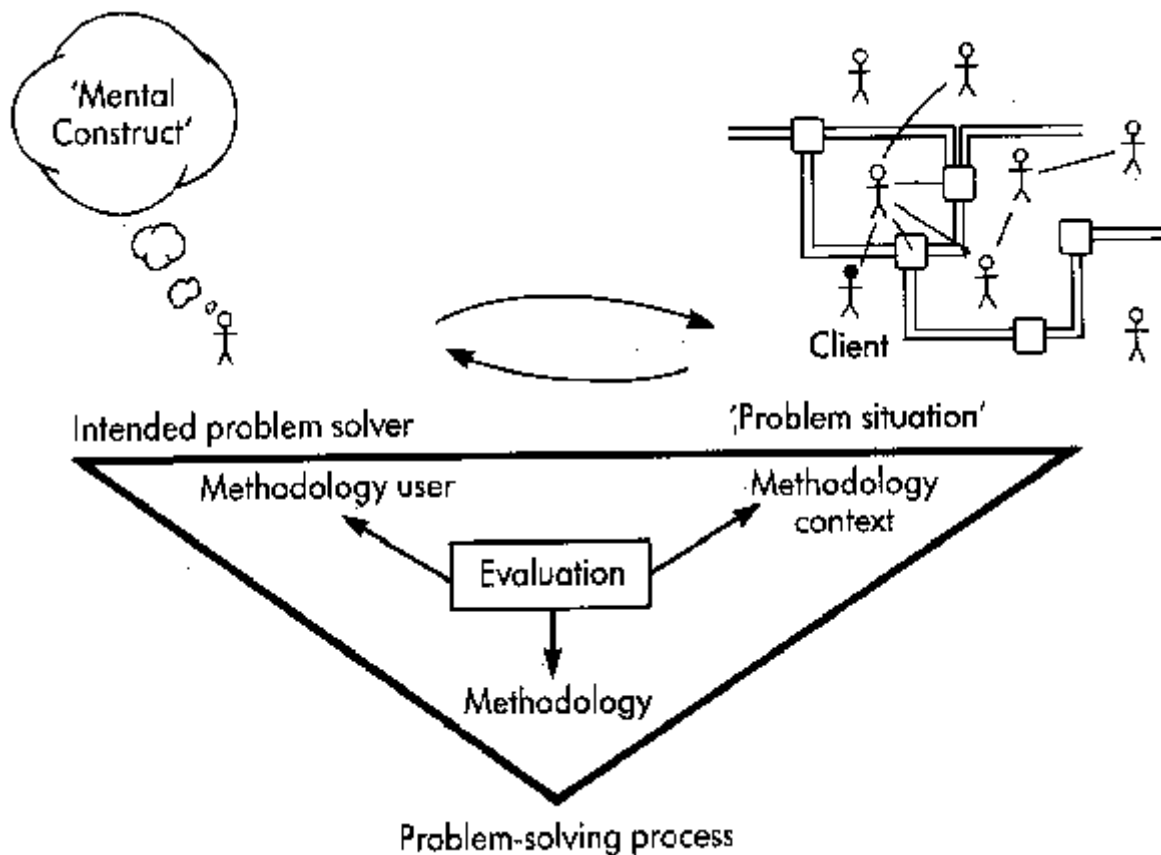


Figure 4 Evaluation (slides by Jenny Coady)

Comparison of SSADM and XP Methodologies

In this section we will go through the comparison of both SSADM and XP methodologies by using NIMSAD framework stages.

Problem Situation:

Both methodologies starts with the planning stage. The team starts with meeting the users, by mentioning users eventually you should consider all stakeholders whom affected by the problem situation. In some situations organisations, government and society representative might be required to contribute in this stage meetings. Early involvement is a key success factor in identifying the problem situation.

In SSADM the technical analyst meet with the client with intention to investigate the problem and gather the system requirements which will be handled by the desired system. The data flow diagram used to descript the situation. Usually analyst used different tools to identify the new system requirements and to analysis the problem situation starts with brainstorming sessions, questionnaires or standard templates. The planning is an important stage in SSADM and to the next stage, so more time and efforts involved here.

As mentioned earlier both SSADM and XP starts with planning stage but not like SSADM, XP methodology starts with high level planning or just enough to start requirement gathering as this stage intended to be iterative. The XP nature of doing things in iteration make the involvement of client / user often in each iteration. The analyst is a technical person, also developers are involved in the planning stage. The approach in XP is flexible in terms of users requirement during the development life cycle the stakeholders will have more visibility on the problem situation and the outcome of the new system.

Intended Problem Solver:

In Both SSADM and XP the developers, analysts and project manager are the problem solver due to the technical nature of the desired system after requirement gathering stage. Only developer can understand the outcome of the system analysis and problem description from the previous stage. End users might not have the technical ability to understand the situation and eventually they need help from technical person if they intend to have efficient contribution.

The XP methodology may have advantage in this situation as users can have more visibility during the development stage and the opportunities given in each iteration.

Problem Solving Process:

1. Phase 1 - Problem formulation:

Stage 1 – Understanding the Situation of Concern

In SSADM the project team set the boundaries for based on the understanding of the problem situation. The data flow diagram is a helpful tool to describe the problem situation and construct the boundaries, but in the DFD it is difficult to identify the hidden boundaries specially in this early stage.

Like SSDM in XP the project team are involved in this stage, but not like SSADM in dealing with the problem hidden boundaries. The team has multiple opportunities to change the problem situations and the expand or reduce the boundaries in future iterations.

Stage 2- Performing the Diagnosis

In SSADM methodology the data flow diagram provide image on how problem solved in each data flow state. It will say how data processed, also will have logical diagram and logical data flow model. The available documentation in this early stage will only indicate the regular used pattern.

As XP has just enough requirement to start problem diagnosis, the early stage is not efficient to provide fully comprehensive problem diagnoses. The understanding of the problem situation will be advanced in later iterations so forth the diagnoses shall be performed in multi stages.

It might seem weak for XP to perform the problem diagnoses, but from my experience it doesn't matter how much time the team spends in all previously mentioned stages. Due to the technical nature of both methodologies users will have advantage in XP methodology to change their understanding to the problem situation, therefore the problem diagnoses can be performed efficiently in later stages and during the development lifecycle.

Stage 3 – Defining the Prognoses Outline

In both SSADM and XP the prognoses outline stage is not important as the desired new system is designed based on client requirement. Although the XP has advantage here as the client can change their requirement during the project execution in further iterations based on the understanding of the new system functions which is usually tested by the users in every iteration.

Stage 4 – Defining the Problems

In SSADM the problem is defined as early in the feasibility study, although the problem is defined but it is not clear as they don't have prognoses. They are going to develop the system based on the defined problem.

In XP the client requirement is a living document which might be changed in advanced iteration, so problems can be further defined or redefined.

Stage 5 – Deriving Notional Systems

The client requirements gathering is a lengthy process in SSADM, so this is an advantage in this stage. Developers will start work based on the gathered user requirements to investigate if they have enough requirement to start system development or if it is impossible to do so. This means further investigation is required.

In XP the developers will have the initial client requirement to start work in the system development. The approach of XP depend on gathering more requirements and further investigate them in every iteration. As a result if each test requirement might be changed or modified.

2. Phase 2 - Solution Design:

Stage 6 – Performing Conceptual / Logical Design

Data flow diagram, logical diagram, and ER entity relationship diagrams are commonly used on most of the methodologies to explain the process stages, the role of individuals. Data flow shows the information required on each step and the flow of information. ER represents the relation

between the information, the constraints and the characteristics of the same. This stage like the logical / conceptual design analysis.

SSADM has advantages over XP on this stage, because the methodology will use the data flow diagram to describe the process. In SSADM the logical design is documented as a design base for the coming stages and for the developers to make the physical design. In XP methodology the team will make just enough logical design to start coding, then eventually the design will grow, change and be mature along with the project progress.

Stage 7 – Performing Physical Design

This stage means coding for developer, user interface for designers, hardware infrastructure for system engineers and in summary connectivity / security / performance for others. In SSADM this stage the developer will focus on the technical side of delivering the solutions. Other areas as hardware and user experience will be investigated.

There is some drawback in SSADM in this stage, because the team focuses on the technical side of the design but not on other aspects of the solution. In this particular stage the XP has the disadvantages in more context, because XP team focuses on the coding more than other parts. But I can see some benefits of XP methodology by involving the test users in each iteration not only to test the code but also to evaluate the functional outcomes.

3. Phase 3 - Design implementation:

Stage 8 – Implementing the Designs

I can see the implementation stage is usually going along with the design stage. In SSADM the user will be trained and involved on this late stage to test the solution. This stage covers system integration with existing systems and current business process.

In XP methodology the listening approach will give the implementation stage different nature. Users usually involved early to test the solution and to provide feedback, which ultimately considered as part of the next solution design iteration.

Evaluation

Both methodologies have advantages and disadvantages on each stage, but the common conclusion for both methodologies that they requires a high technical skilled people. The project team and people whom developing shall be experienced to deliver the right solution. SSADM and XP are good methodologies to deal with system situations, but can't be utilized on wider area of problem solving.

In SSADM and XP the involvement of people is limited to technical expertise, business analysts and user whom state's the requirements. Some organizations hire external consultant to do the information gathering, user functional requirements and users requirements. Obviously business process reengineering is required, because implementing new system will change the way people work and interact.

The client requirement is identified, the system requirements are defined since the early stages, so evaluating the system outcome basically done on predefined bases. But in SSADM the objectives are not clear to the development team. Client will be able to see the result at the end of each stage and will be able to provide the feedback. The concern here is the need to rework is a long journey and the project team has to go through - the waterfall approach- all the SSADM stages to reproduce.

Nothing can be compared with seeing or trying, the author recommends how the XP methodology involves the client / users to see and test the system functionality on each iteration. This is more powerful in a sense of getting real effective feedback. The XP methodology is more responsive to changes, by controlling and reevaluating the small releases .

Conclusion

This report covers SSADM and XP methodologies, obviously both are information system domain related. Comparing two methodologies from the same nature in dealing with situation was a challenging task spatially to identify the tiny difference in each stage of the NIMSAD framework. Usually, the comparison between methodologies to solve structured / unstructured problems which intended to deliver a system, or intended to solve a wider problem.

As outcome of the report; I have learned the advantages and disadvantages of both SSADM and XP methodologies, hoping the readers also have benefited from the previous comparison helping them make decision on which methodology to use based on the system requirement and which project approach to use.

In SSADM structured approach are used for designing and modelling. SSADM will have feasibility study, analysis stage, logical and physical design. XP will have analysis, coding, testing, and listening which depends on users feedback in accumulative way. The SSADM is a waterfall model which handles the Software Development Life Cycle in the implementation project. Documentation on each stage of SSADM is more focused than in XP. The XP focuses more on coding and testing rather than documenting.

In SSADM and XP the technical skills in the project team is a success factor. XP do care about the working environment and the 40 working hours per week as a factor to keep the team motivated, focused and not exhausted.

The author believes that the SSADM is a costly methodology for implementing large projects, also the need to write more documentation and maintaining them make the control over stage expensive and requires more manpower. The author recommends XP methodology for moderate projects, specially if the client worried about the accuracy of their requirements.

The bottom line, there are several factors can contribute on the selection decision between different methodologies other than the points covered in this report. This includes but not limited to, the budget constraints, the organization size, policy and procedures, political situations, the clarity of the problem, and the people.

References

- McKay, V. (n.d.). *What is SSADM? | Analysis and Design / FAQ*. [online] Selectbs.com. Available at: <http://www.selectbs.com/analysis-and-design/what-is-ssadm>.
- Rouse, M. (n.d.). *What is SSADM (Structured Systems Analysis & Design Method) ? - Definition from WhatIs.com*. [online] Search Software Quality. Available at: <http://searchsoftwarequality.techtarget.com/definition/SSADM>.
- Informatik.uni-bremen.de, (2002). *2.8 The Complex Method SSADM*. [online] Available at: <http://www.informatik.uni-bremen.de/gdpa/methods/m-ssadm.htm>.
- Beal, V. (n.d.). *What is SSADM? Webopedia*. [online] Webopedia. Available at: <http://www.webopedia.com/TERM/S/SSADM.html>.
- Jayaratna, Nimal. *Understanding And Evaluating Methodologies*. London: McGraw-Hill, 1994. Print.
- Smak, M. (2012). *Extreme programming*. [Online] Slideshare.com Available at: <http://www.slideshare.net/MrSMAk/extreme-programming-12047889>.
- Withrow, S. (2001). *Extreme Programming: Do these 12 practices make perfect?* - TechRepublic. [Online] TechRepublic. Available at: <http://www.techrepublic.com/article/extreme-programming-do-these-12-practices-make-perfect/>
- Community.mis.temple.edu,. 'SDLC Vs. The Chaotic Evolution Of Technology (System Development Life Cycle) | MIS2101 – Summer 1 2015MIS2101 - Summer 1 2015'. N.p., 2015. Web. 21 Nov. 2015. [Accessed on 20 November 2015].
- UKEssays,. 'Comparision Of Ssadm And Ethics Using Nimsad Information Technology Essay'. N.p., 2015. Web. 21 Nov. 2015. [Accessed on 20 November 2015].
- Extremprogramming.org,. 'Extreme Programming: A Gentle Introduction.'. N.p., 2015. Web. 21 Nov. 2015. [Accessed on 21 November 2015].