Journal of
Information Technology & Software Engineering

Editorial Open Access

System Analysis and Design

Ramakrishnan S*

Department of Information Technology, Dr. Mahalingam College of Engineering and Technology, Pollachi, India

Introduction

Systems Analysis and Design (SAD) is a broad term for describing methodologies for developing high quality Information System which combines Information Technology, people and Data to support business requirement. The SAD technique is not only limited to IT systems and can be used to create just about anything, from a family house to the international space station. But there is no silver bullet in simplifying the development of computer systems. This principle is still true today. In other words, there is no single, simple technique that developers can use to ensure successful Information Technology (IT) projects. However, there are development methodologies that can be followed which will greatly assist an IT professional in developing and enhancing systems. [1] A methodology is essentially a procedure to get something done. A development methodology can be thought of as a roadmap. While a roadmap for a traveler will provide the details from driving from point A to point B, a development methodology will provide the IT professional with guidelines for taking a system from conception through implementation and beyond [2]. Before going into the details of System Analysis and Design we will shortly discuss the history of it.

Brief History

The growth of Software Engineering methodologies started during the year 1950 to 1960. The new technologies and practices which were developed after 1970 -1990 were primarily focused on solving the software issues like software crisis [3]. The major elements used were software tools, formal methods, well defined processes that uses the methodologies like OOP, CASE tools and Structured Programming approaches.

The researchers argued that the software crisis was due to the lack of discipline of programmers and Some believed that if formal engineering methodologies would be applied to software development, then production of software would become as predictable an industry as other branches of engineering and they advocated proving all programs correct using models such as the Capability Maturity Model [4].

In 1986, No Silver Bullet article was published by Fred Brooks which described that no individual technology or practice would ever make a 10 fold improvement in productivity of software within 10 years. So they realized the need for developing the software in a structured manner. However, it could also be said that there are, in fact, a range of silver bullets today, including spreadsheet calculators, lightweight methodologies, in-site search engines, customized browsers, integrated design-test coding-editors, database report generators and each issue in software is related with only a small portion of the entire problem which makes the software engineering approaches too complex for finding complete solution to all problems [5,6].

The Need for Systems Analysis and Design

The software projects have a poor track record and the U.S. National Institute of Standards and Technology stated that in 2003, software bugs cost domestic companies $59.5 billion (0.6% of GNP). A 2002 Standish Group study (CHAOS) found that only 34% of IT projects were completed on budget and 17% were complete failures. In 2005, the FBI abandoned their Virtual Case File (VCF) project after spending over $100,000,000 on code that will never be used. Even so, the need for IT projects keeps increasing. In 2000, there were 300,000 new IT projects, and in 2001, over 500,000 new IT projects were initiated. [7] So, understanding SAD is the first step in developing a successful project. However, the proper application of SAD principles is the key to making a project successful. In order to properly apply these principles, an analyst must be able to lead, communicate and sell the SAD principles to the project team.

Methodologies of Sad

A SAD methodology can also be referred to as a Systems Development Life Cycle (SDLC) that includes the development process as well as the ongoing maintenance process. The classic SAD methodology is the waterfall model which was originally conceived for the software development; hence the focus is on programming. The key phases of the waterfall model are the analysis and design phases. It is obvious that there will always be an implementation phase and an operations phase. The analysis phase focuses on understanding the needs of the organization. The design phase focuses on designing the physical aspects of a system to support the needs of the organization [8].

Many projects fail because the implementation of the system is pursued without understanding the reason for the system in the first place. This concept is equivalent to purchasing lumber for a new home without even having discussed the required size or style of the structure. While no one would start a house project by purchasing lumber without some planning, many IT projects have been initiated by purchasing software or hardware without fully considering the needs of the organization. Obviously there are many ways to go about developing an IT system. Likewise, there are many SAD methodologies. For example, the spiral methodology or Extreme Programming techniques can also be employed in the place of waterfall model. The spiral model iterates several times through the phases and includes some additional items. This model builds on the waterfall model concepts. The spiral model is intended for large software development projects, thus it would be overkill to use this methodology for a small project. Extreme programming is considered an agile development methodology tailored for small and medium sized projects. It streamlines the development process, but does not eliminate the fundamental steps of planning, analysis and design [9].

The SSADM

The advancement in SAD resulted the SSADM (Structured

***Corresponding author:** Ramakrishnan S, Professor and Head, Department of Information Technology, Dr.Mahalingam College of Engineering & Technology, Pollachi-642003, India, Tel: 914259-236030, 914259-236040; Fax: 914259-236060; E-mail: ram_f77@yahoo.com

Received December 27, 2012; **Accepted** December 28, 2012; **Published** December 31, 2012

Citation: Ramakrishnan S (2012) System Analysis and Design. J Inform Tech Soft Engg S8: e001. doi:10.4172/2165-7866.S8-e001

Copyright: © 2012 Ramakrishnan S. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Systems Analysis & Design Method) which is a widely-used computer application development method in the United Kingdom, where its use is often specified as a requirement for government computing projects. It is increasingly being adopted by the public sector in Europe. It 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. The major objectives of this methodology includes the improvement of the project management & control, making more effective use of experienced and inexperienced development staff, developing better quality systems, making projects resilient to the loss of staff, enabling projects to be supported by computer-based tools such as computer-aided software engineering systems and establish a framework for good communications between participants in a project [10].

In detail, SSADM also follows waterfall view of systems development, in which there are a series of steps, each of which leads to the next step. This method also includes feasibility Study, investigation of the current environment, business systems options, definition of requirements, technical system options, logical design and physical design. For each stage, SSADM sets out a series of techniques and procedures, and conventions for recording and communicating information in textual and diagrammatic form. SSADM is a very comprehensive model, and the main characteristic of the method is that projects may use only those elements of SSADM appropriate to the project. SSADM is also supported by a number of CASE tool providers who makes the automated software development [10].

Summary

The area of SAD has the contribution of different developmental teams which includes the system analysts who are analyzing how users interact with technology and businesses function by examining the inputting and processing of data and the outputting of information for improving organizational processes. Many improvements involve better support of users work tasks and business functions through the use of computerized information systems. So the system analyst must play the role as a consultant, a supportive expert and an agent for change.

The Business Analysts focuses on the business issues of the system. These include identifying the business value that the system will create, developing ideas and suggestions for how the business processes can be improved, and designing the new processes and policies in connection with the systems analyst. This individual will be equipped with the business experience and good professional training for analyzing business trends.

The infrastructure analyst concentrates on the technical issues surrounding how the system will interact with the organization's technical infrastructure. They are also ensuring that the new information system conforms to organizational standards and identifying infrastructure changes needed to support the system. These individuals will have experience in networking, database administration, and various hardware and software products [11].

Another important role in SAD is the change management analyst who works on the people and management issues surrounding the system installation. The roles of this person include ensuring that adequate documentation and support are available to users, providing user training on the new system, and developing strategies to overcome resistance to change. This individual likely will have significant training and experience in organizational behavior in general and change management in particular. He or she represents the interests of the project sponsor and users for whom the system is being designed. The change management analyst works most actively during the implementation phase but begins laying the groundwork for change during the analysis and design phases.

References

1. Wasson CS (2005) System Analysis, Design, and Development: Concepts, Principles, and Practices. Wiley Series in Systems Engineering and Management.
2. Langer AM (2007) Analysis and Design of Information Systems. Software engineering.
3. Sydow A, Tzafestas SG (2000) Systems Analysis Modelling Simulation - Special issue on systems modelling, analysis and design. Systems Analysis Modelling Simulation, USA.
4. Wang S, Wang H (2012) Information Systems Analysis and Design. Universal Publishers, USA.
5. Ahern DM, Clouse A, Turner R (2003) CMMI distilled: a practical introduction to integrated process improvement. The SEI Series in Software Engineering, Addison-Wesley.
6. Parsons J, Saunders C (2004) Cognitive Heuristics in Software Engineering applying and Extending Anchoring and Adjustment to Artifact Reuse. IEEE Transactions on Software Engineering 30: 873-888.
7. Prowell SJ, Trammell CJ, Linger RC, Poore JH (1999) Cleanroom Software Engineering: Technology and Process. Addison-Wesley.
8. Dobing B, Parsons J (2000) Understanding the Role of Use Cases in UML: A Review and Research Agenda. Journal of Database Management 11: 28-36.
9. Hungerford BC, Hevner AR, Collins RW (2004) Reviewing Software Diagrams: A Cognitive Study. IEEE Transactions on Software Engineering 30: 82-96.
10. Tilley S, Parveen T (2012) Software Testing in the Cloud: Perspectives on an Emerging Discipline. Idea Group,U.S.A.
11. Hay D (2012) UML and Data Modeling: A Reconciliation. Technics Publications.