CHAPTER II

THEORY AND LITURATURE EVALUATION

2.1. Information System

Information is the outcome or processed data which can be read and understood using common language, as stated in the book titled "Introduction to Information System" published by George M Marakas and James A O'Brien. Data are raw material which has not been processed for people to see. In addition an Information System is an organized combination of people, hardware, software, communication networks and the data resources that collects, transforms and disseminates information in an organization.

2.2. Understanding System

A System is a Set of Interrelated components, with clearly defined boundary, working together to achieve common set of objectives. Every system is delineated by its spatial and temporal boundaries, surrounded and influenced by its environment, described by its structure and purpose and expressed in its functioning. A system accepts input and process them produce reliable output for the user to use.

The basic functions of a system are:

1. **Input** : Capturing and assembling elements that enter the system to be processed
2. **Process** : Transformation process that converts input into output

3. **Output** : Transferring transformed elements to their ultimate destination

2.2.1. Characteristics of a System

A system has the characteristic of having components, limited environmental resources and it mainly focus on inputting data, processing the data and outputting meaningful information to meet the target or goal of the system.

1. Mini System With Components

A system consists of a number of interacting components, which work together to form a complete unit. System components or system elements can be either a subsystem or the parts of the system.

2. System Restrictions.

Boundary is an area that limits a system between a system with other systems or with External environment. All system are designed to perform certain task according to their restrictions, they cannot perform beyond their restrictions.

3. System External environment.

The External environment of a system is anything beyond the limits of the system that affect the system's operation. Outside the system environment can be beneficial as well as favorable external detrimental. Environment is the energy of the system and thus must be maintained and managed. Whereas adverse external environment must be retained and controlled, otherwise it will interfere with the viability of the system.

4. System Liaison.

Liaison is a medium that combines the subsystem with other subsystems. Through this link the possibility of resources flow from one subsystem to another

subsystem is effective. Output from one Subsystem will be the input for the other subsystems through liaison. By connecting one Subsystem to another can continue until integration with other subsystems to form a unity.

5. System Input.

The energy source that drives a system is input source, input can by in any form of Electronic data, graphical data, picture data, sound wave data, mathematical data in numbers, and binary in '0's' and "1's" , as mentioned by D. Gotterbarn in the booked title "Software engineering ninth edition"

6. System Output.

Energy system output is processed and classified into useful output. Output can be an input for another subsystem.

7. System Processing.

All system has a processing centre, this processing system is design to accept input process them and output perfect information. The system processing centre of a system is very essential for any given system.

8. System Target.

A system has goals and objectives; if the system does not have a target then the system would not exist. A system is successful when it is focusing on the target or goal. Target is very influential when there is an input and outputs are produced.

2.2.2. Classification of System

The system is a form of integration between the components of a system with other components. A system has different target or goal, if a system has many subsystems; they all have different targets or goals that they must achieve.

Therefore, the system can be classified into several viewpoints. Some example of a system that is abstract, natural systems, the nature of which is deterministic and systems that are open and closed. The more detailed and detailed explanation will be presented below:

1. **Abstract system** is a system in the form of thoughts or ideas that do not appear physically.

2. **Natural systems** are systems that occur due to natural processes, they are not created by humans (determined and subject to the will of the creator of nature).

3. **Specific system** (deterministic system) and indeterminate system (probabilistic system). A Particular system which operates with a predictable behavior. Particular system is relatively stable in the long term. Indeterminate system is a system with conditions of the future and it is unpredictable because it contains probability.

4. **The Open and Closed systems**

Closed system is a system that is not related and not affected by the external environment. This system works automatically without any intervening from the outside. An open system is a system that is associated with and affected by the external environment. This system Receives input and produces output to the outside environment or other subsystems. According to Roger S. Pressman. PHD, in the book titled Software Engineering Ninth Edition (2012:166).

2.3. Understanding Data

Data are raw material which has not been processed for people to see, Data can be in many forms:

1. **Electronic data**

Electronic data are soft copy of data that are in a form of code or graph where people cannot understand it, it need processing to be understood as information. According to Roges S. Pressman. PHD, (2012:115).

2. **Graphical Data**

Graphical Data can also be seen as data saved in a form of graphic; it can only be understood by user through processing.

3. **Picture Data**

Some data are stored as picture; they also need to be processed to be viewed by others.

4. **Sound wave data**

Most data are stored as Sound wave, it can be processed to convey and understand the meaning of it.

5. **Mathematical Data in Numbers**

Mathematical data in numbers is a common type of data, it must also be processed for users to understand them.

6. **Binary digits in 0's and 1's.**

Binary digits also store up data, to understand those data, they must be processed and displayed in common language.

The above mentioned data can be processed to come up with information readable for users.

2.4. Understand Information

Information is the outcome or processed data by any given system which can be read and understood using common language, information is very essential to use and make effective decision for any given organizations. According to Barbara C (2002:432) in a book titled "Information systems management in practice", 6th edition, "Information contains meaning which is very important, the recipient can understand that information and can further use it to draw conclusion or even take a step in making beneficial decisions". Viewing the above definition of information, it can be said that information obtained from a system is very important.

2.4.1. The basic concept of information

For information derived from the processing of data must have a different quality. According Barbara C (2007: 80) the quality of information consists of three things: accurate, relevant and timely.

1. Accurate

Accurate means that information should be free of errors, not biased or misleading. Accurate also means that information must clearly reflect the intent. Information must be accurate to be resourceful to the receiver of the information to make informed decision. Some things that can affect the accuracy of information include:

a. Completeness Information

The resulting or final information consists of a single unit of information that is comprehensive and covers various matters related thereof. Because of the

information, generated partial course will affect the decisions or determine the action as a whole, so that will affect the ability to control or solve a problem with either.

b. Truth (correctness) Information

The information generated by the data processing, must be strictly in accordance with existing calculations in the process.

c. Security Information

An information should be safe, in the sense that only accessed by those who are interested only in accordance with the nature and purpose of the information.

2. Relevant

Means that information has benefits for the wearer. The relevance of the information for each person different from one another, then the information can be useful say if it really useful and in need of the wearer.

3. Just in time

Some of the information that comes at the receiver should not be too late. Obsolete information will not have value anymore because the information constitutes grounding in taking decisions.

2.5. Understanding Information System

Information is the outcome or processed data which can be read and understood using common language, while Data are raw material which has not been processed for people to see. A System is a Set of Interrelated components, with clearly defined boundary, working together to achieve a common set of objectives. Every system is delineated by its spatial and temporal boundaries, surrounded and influenced by its environment, described by its structure and purpose and expressed

in its functioning. An information system (IS) is typically considered to be a set of interrelated elements or components that collect (input), manipulate (processes), and disseminate (output) data and information and provide a feedback mechanism to meet an objective. In addition an Information System is an organized combination of people, hardware, software, communication networks and the data resources that collects, transforms and disseminates information in an organization. According to Roges S. Pressman. PHD, (2012:125).

2.5.1. Component of Information System.

According to Mathew beard in the book titled "Information Systems: Development Methodology and Evolution", components of information system are building blocks that can be divided into six blocks, namely:

1. Block input

Block input is the data entered into the information system, which can be either basic document that can be processed into certain information.

2. Block model

This block consists of a combination of procedures, logic and mathematical models which will process the input data to generate required information.

3. Block output

Is information that generates a set of data that will be stored in the form of print data reports.

4. Block technology

Block technology is major support system ongoing information. Which has several components, namely the tools to enter data (input device), a tool for storing

and accessing data (storage device), a tool to generate and send the output (output device) and tools to make a total help control system (control device). Consists of information technology

Of three (3) main parts, namely technician (human ware or brainier), software, and hardware.

5. Block the database

The database is a collection of data related to each other, are stored in the computer hardware and software used to manipulate it. The data needs to be stored and the need in the organization in such a way that the resulting quality information.

6. Block of control

Some controls need to be designed and implemented to ensure that the things that can damage the system can be prevented if already occurred mistakes in the use of system damage.

2.5.1. Basic component of Information System

Developing of Systems takes a long time of planning and analysis before it is literally developed. Development of an effective system is done through carefully following the regulations set by Major Software development organizations like Institute of Electrical and Electronic Engineers (IEEE) software. The basic components of an information system are;

1. **Hardware**

These are the devices like the monitor, processor, printer and keyboard, all of which work together to accept, process, show data and information.

2. **Software**

Software's are the programs that allow the hardware to process the data.

3. **Data**

Data are facts that are used by programs to produce useful information. Like programs, data are generally stored in machine-readable form on disk or tape until the computer needs them.

4. **Network**

Is a connecting system that allows diverse or different computers to distribute resources with each other.

5. **People**

Every Information System needs people if it is to be useful, people are the users of the information system probably the component that most influence the success or failure of information systems.

The components of the information system mentioned above are all inter related. Relationship can be stated as:

- Hardware used processor to produce useful information
- People input data into the computer using hardware to be processed by the processor
- People follow procedure to make information system function
- Carry out instructions to turn data into useful information
- People use procedure to make information software function properly

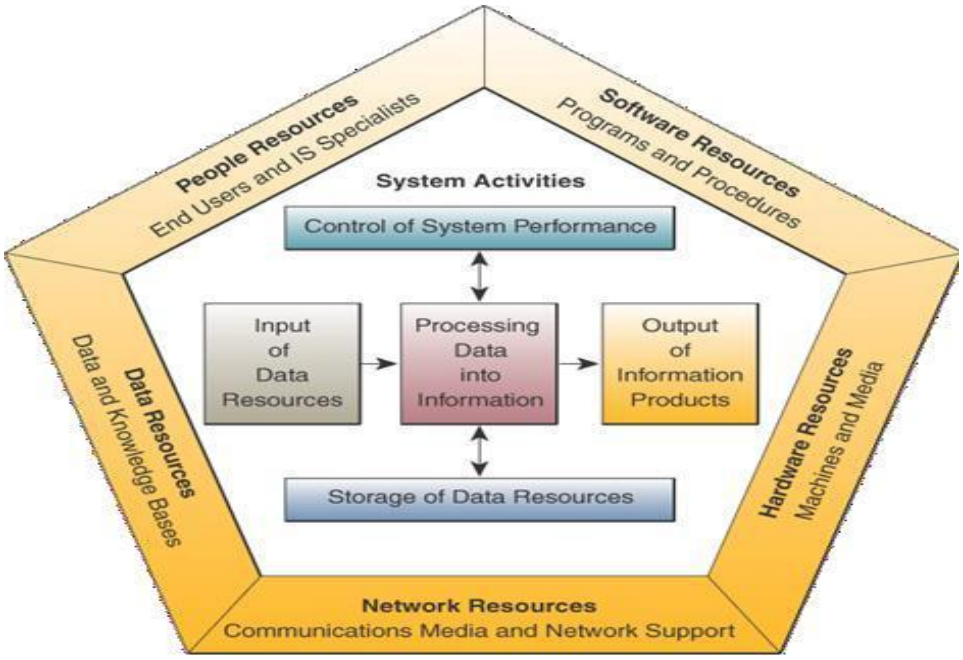


Figure 2.1 The structural lay out of the components of Information System. (James A. O'Brien, George M. Marakas, "Management information systems", 8th edition, McGraw Hill, 2008.)

The five components of an information system work together to produce an effective complete system. Systems work corporately to accept a data input and process it to produce information which is useful for users. As the diagram stated above, the entire five components act as a strong building kept intact to create a conducive environment with controls of procedure to effectively input data resource and process data and output information as a product.

The inter-relationship between the above diagrams can be further explained by the following diagram;

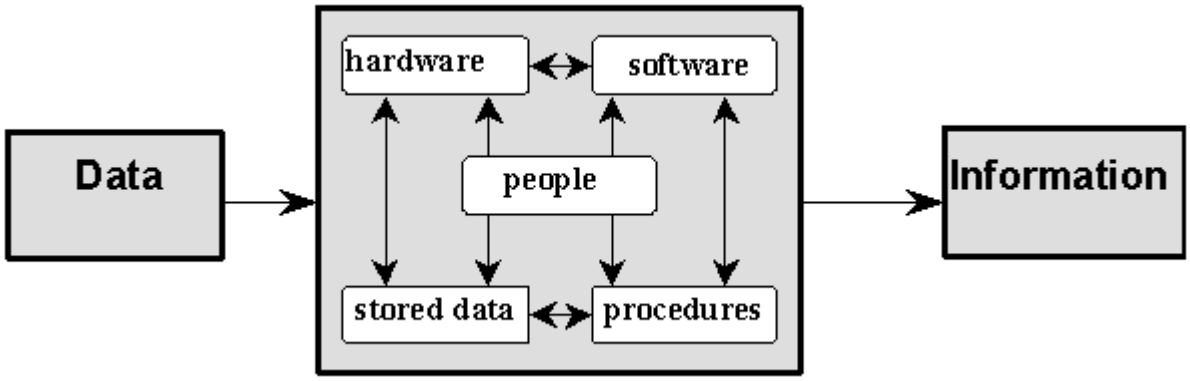


Figure 2.2 Data Processing diagram of information system involving the 5 Information System Components. (James A. O'Brien, George M. Marakas, "Management information systems", 8th edition, McGraw Hill, 2008.)

2.6. Scheduling

The notions of Scheduling are as follows:

2.6.1. Understanding Scheduling

Scheduling is a collection of policies and mechanisms in the operating system relating to the sequence of work done by the computer system. Scheduling is charge of deciding:

- The process should be run

- When and for how long the process was running.

Main Target Optimization Process Scheduling work to meet particular criteria.

2.6.2. Criteria Scheduling

According to (Tan: 92 and MIL: 92) the criteria for measuring the performance of scheduling and optimization:

1. Fair (fairness)

2. Efficiency

3. Response time

4. Turnaround Time

5. Throughput

1. **Fair (fairness**)

Fair are the same processes are required to get the same amount of time in processing and no one who did not get the service process so experienced start vision processors. Scheduling target should ensure each process gets the service of fair processing.

2. Efficiency

Efficiency or processor utilization is calculated by comparison (ratio) is busy processing time. Scheduling target is to keep the processor busy that remain in a state of achieving maximum efficiency. Busy is not idle processors, including the time spent to execute user programs and the operating system.

3. Response Time

Different response times for:

- Interactive Systems

- Real-time Systems

- Response time in interactive systems

Response time in interactive system is defined as the time spent from the time of the last character of the command or transaction entered until the first result appears on the screen (terminal). This response time is called the terminal response time.

- Response time to the real time system

In real-time systems, response time is defined as the time from the time of the incident (internal or external) until the first instruction of the service routine is executed, called event response time. Target scheduling is to minimize response time.

4. Turnaround time

Turnaround time is the time spent on the current program or job started to get into the system until the process is completed by the system. The time in question is the time spent in the system, expressed as the amount of execution time (service time job) and waiting time, namely: Turnaround Time = time + waiting time execution, scheduling Goal is to minimize the turnaround time.

5. Throughput

Throughput is the amount of work that can be completed in one unit of time. How to express the throughput is the number of jobs that users can executed in one unit / interval. Scheduling target is to maximize the number of jobs that are processed at a time interval. Higher throughput figures, more work done system. These criteria can also join together and so it is not possible conflicting optimization criteria simultaneously. Example: To give a small response time requires scheduling processes are often switched between it. This method increases the system overhead and reducing throughput. Wisdom scheduling scheme involves a compromise between the needs of the conflicting. This compromise relies nature and use of computer systems.

2.7. Voting System of National Capital District

The notions of the National Capital District voting system are as follows:

2.7.1. Definition of voting System

National Capital District is one of the 22 provinces of the nation of Papua New Guinea; it is the central province where the capital city of Papua New Guinea, Port Moresby is located. The National Capital District is looked up at by other province as the one model province of the nation of Papua New Guinea.

Voting System is the Procedural System put in place by the Electoral Commission of Papua New Guinea for the citizens to follow through during the national election. The Electoral Commission of any nation is responsible to conduct any given election and manage citizen's data, therefore, The Electoral Commission deals with Citizen Data and management of National Election. The voting system of National Capital District is the step by step procedures follow by all citizens of National Capital during a national election to vote for their District Candidate and Regional/ Provincial Candidate.

2.8. Understanding Database

According to Marcia Harton,B.Boehm PhD, (2005: 44), database is defined as the set of groups of data (files) that are interconnected in such a way that is organized so that later can be recovered quickly and easily stored in electronic storage media. According to the above understanding can be concluded that the database is a group of centralized data organization. The database is generally analogous to a filing cabinet or cupboard set of documents.

2.9. Software Support

Below is the description of the software:

2.9.1. Eclipse

According to the Marcia Harton (2013: 16), Eclipse is an IDE (Integrated Development Environment) to develop the software and can be run on all platforms (platform independent).

Here is the nature of Eclipse:

1. **Multi-platform**

Eclipse Target operating system is Microsoft Windows, Linux, Solaris, AIX, HP-UX and Mac OS X.

2. **Multi-language**

Eclipse was developed with the Java programming language, but Eclipse-based application development support other programming languages such as: C / C ++, Cobol, Python, Perl, PHP, and so forth.

3. **Multi-role.**

Besides as an IDE for application development. Eclipse can be used for activities in the software development cycle such as documentation, software testing, web development, and so forth.

At this time, the Eclipse IDE is one of the favorites for a free and open source. Open source means that everyone should see this software programming code. In addition, the advantages of Eclipse that makes it popular is its ability to be developed by the user to create a component called plugins.

Eclipse was originally developed by IBM to replace the software development of the IBM Visual Age for Java 4.0. Eclipse product was launched by IBM on November 5, 2001. IBM is investing US $ 40 million for development. Since 5 November 2001, the consortium Eclipse Foundation took over further development.

2.9.2. Eclipse Architecture

According to the Marcia Harton (2013: 17), since version 3.0, Eclipse is essentially a kernel. What can be used in the Eclipse is actually a function of the plug-ins that are already installed. This is called the base of the Eclipse Rich Client Platform (RCP).

1. The following are the components that make up the RCP:

Core platform

2. OSGi
3. SWT (Standard Widget Toolkit)
4. Jface
5. Eclipse Workbench

By default Eclipse is always equipped with JDT (Java Development Tools), plug-ins that make compatible Eclipse to develop Java programs, and PDE (Plug-in Development Environment) to develop new plug-ins. Eclipse plug-ins as well as its implementation in the Java programming language.

Eclipse is an IDE concept is

1. Open
2. Easily expandable for anything, and

3. Not for something specific.

Eclipse is not only to develop Java programs, but also for a variety of purposes. Any extensions simply by installing plug-ins are required. If you want to develop a program of C / C ++ then there has been a plug-in CDT (C / C ++ Development Tools) that can be installed in Eclipse to be a tool for developing Eclipse C / C ++.

There is nothing that is not possible by the Eclipse UML2 plug-ins available to create UML diagrams. By using PDE every person can make a plug-in accordance with his wishes. One site that offers free plug-ins that are Eclipse downloads by project.

2.9.3. Java Programming

Java is a programming language that can run on a variety of computers including mobile phones. This language was originally created by James Gosling while still joined Sun Microsystems is now a part of Oracle and released in 1995. These languages are adopting syntactic contained in C and C ++ but with syntax simpler object model and support top-down routines minimal.

Java based applications are generally compiled into p-code (byte code) and can run on different Java Virtual Machine (JVM). Java is a programming language that is general / non-specific (general purpose), and is specifically designed to utilize minimal implementation dependencies as possible. Because the functionality that allows Java applications can run on several different operating system platforms. Currently Java is a programming language used most popular

and widely used in the development of various types of software applications or web-based applications.

According to Gotterbarn (2006) Characteristics of Java, among others:

- Simple
- Object Oriented
- Distributed
- Interpreted
- Robust
- Secure
- Neutral Architecture
- Portable
- Performance
- Multithreaded
- Dynamic

Various Java 2 Software Developers Kit (J2SDK), among others:

- J2SE (Java 2 Standard Edition)
- J2EE (Java 2 Enterprise Edition)
- J2ME (Java 2 Micro Edition)

Java has a platform that is:

- Java Virtual Machine (Java VM)
- Java Application Programming Interface (Java API)

Java applications, among others:

- Stand alone

- Applets
- Windows-based Applications (GUI)
- Java Servlet
- Java Server Pages (JSP)
- Java Beans and Enterprise Java Beans
- Java Micro Edition

2.9.4. Java Architecture

In architecture, Java has not changed at all since the beginning of the language was released. Java compiler (called by java or Java Compiler) will transform the code in the Java language into a byte code. What's byte code? Byte code compilation result is a set of commands that can then be executed by a computer abstract machine, called the JVM (Java Virtual Machine). JVM is also often called as an interpreter, because it is always translate the codes stored in the byte code by way line by line.

2.9.5. Unified Modeling Language (UML)

According to Doug Rosenberg (2005). "Unified Modeling Language (UML) is a tool of analysis and design of object-based software". Unified Modeling Language (UML) is a standard modeling language consisting of clusters diagram, developed to help developers and software systems in order to accomplish tasks such as: Specifications, Visualization, Architectural Design, Construction, Simulation and testing and documentation ".

Based on some of the opinions expressed above can be concluded that the "Unified Modeling Language (UML) is a language that is based on the graphic or picture to visualize, specify, build and documentation of a software development system based OO (Object Oriented)". UML is a collection of the best techniques that have proven successful in modeling large and complex systems. UML is not only used in process modeling software, but almost in all the fields that require modeling. The main part of the UML is View, Diagram, a model element, and general mechanism.

1. **View**

   View used to see the system being modeled from several different aspects. View does not see the graph, but is an abstraction that contains a number of diagrams. Several types of View in UML, among others: Use Case View, Logical View, Component View, concurrency View, and Deployment View.

2. **Use Case View**

   Use Case View describe the functionality of the system is supposed to do according to the desired external actors. Actors that interact with the system can be either a user or system is described in other. View Use Case Diagrams and sometimes with Activity Diagrams. This view is used primarily for the customer, the designer, developers, and system testers.w

3. **Logical View**

   Logical View describes how the functionality of the system, the static structure (class, object, and relationship) and dynamic collaboration that occurs when an object sends a message to another object in a particular function. This

view is described in the Class Diagram for the static structure and the State, Sequence, Collaboration, and Activity Diagrams for dynamic models. This view is used for the designer and developers.

4. **Component View**

Component View describes the implementation and dependency modules. Components which are other types of code modules are shown with the structure and dependence also resource allocation component and other administrative information. This view is described in the Component View and used for the developers.

5. **Concurrency View**

Concurrency View divides the system into the process and processor. This view is described in the dynamic diagram (State, Sequence, Collaboration, and Activity Diagrams) and Diagrams implementation (Component and Deployment diagrams) and used for the developers, integrator, and testers.

6. **Deployment View**

Deployment View describes physical systems such as computers and devices (nodes) and how it relates to this other. View described in Deployment Diagrams and used for the developers, integrator, and testers.

7. **Diagram**

Diagram shaped graph showing symbol of model elements are arranged to illustrate certain parts or aspects of the system. A diagram is a part of a

particular View and when described usually allocated for a particular View. The type of diagram, among others:

1. Use Case Diagram

According to the US Marcia Harton, Software Engineering ninth Edition, 2012: 134, Use case or use case diagram is a model for the behavior of information systems that will be made. Use case describes an interaction between one or more actors with information systems that will be made. Roughly speaking, the use case is used to determine what function is in an information system and who is entitled to use those functions. Terms naming the use case are defined name as simple as possible and understandable.

2. Activity Diagrams

According to the US Marcia Harton, Software Engineering ninth Edition, 2012: 136, Diagram activity or activity diagram illustrates the workflow or the activity of a system or business process or menu in the software. Noteworthy here is that the activity diagram illustrates the activity of the system; it is not what actors do, so activities can be carried out by the system.

3. Sequence Diagram

According to the US Marcia Harton, Software Engineering ninth Edition, 2012: 137, Sequence diagram illustrates the behavior of objects in use case by describing the life time of the object and the message that is sent and received between objects. Therefore, to describe the sequence diagram objects must be known to be involved in a use case along with the methods of the class owned

instant(computer operational) become the object. Creating a sequence diagram is also needed to look at the existing scenario in the use case.

4. Collaboration Diagram

Collaboration diagrams are another form of sequence diagrams, sequence diagrams which are organized by time, while collaboration diagrams are organized according to the room / space.

5. Class Diagram

According to the US Marcia Harton, Software Engineering ninth Edition, 2012: 138, grade or class diagram is a diagram depicting the structure of the system in terms of defining the classes that will be made to build the system. Classes have what are called attributes and methods or operations. Attributes are variables owned by a class, operations or methods are functions that are owned by the class.

The class diagram is made so that the programmer or programmer make appropriate design classes in the class diagram that is between design and software documentation in sync.

6. Component Diagram

According to the US Marcia Harton, Software Engineering ninth Edition, 2012: 140, Diagram components or component diagram is made to show the organization and dependence among a collection of components in a system. Diagram component focuses on components and systems needed in the system.

7. Deployment Diagram

According to the US Marcia Harton, Software Engineering ninth Edition, 2012: 141, Diagram deployment or deployment diagram showing the configuration of the components in the process of execution of the application. Deployment diagram can also be used to model such things, additional systems (embedded systems) that portray the device design, nodes, and hardware.

The purpose of the use of UML as follows:

A. Provide modeling language that is free from various discuss programming and engineering processes.

B. Brings together the best practices contained in the modeling.

C. Provide ready-made model, expressive visual modeling language support to develop and exchange models easily and commonly understood.

D. UML can also function as a blueprint for a very complete and detailed. With this blueprint will be known bias detailed information about the coding program or even to read and interpret the program back into shape diagram (reserve engineering).

While the software that supports the creation of UML diagram, namely:

a) Star UML

Star UML is an open source project for developing Star fast, flexible, extensible, featureful, and freely-available UML / MDA platform running on a platform Win32.Tujuan of Star UML project is to build a software modeling tool and also platform that is interesting is the replacement tool commercial UML Rational Rose, Together and so on.

b) Acceleo

Acceleo is a code generator that transforms the model into code. Acceleo is easy to use and provide "off the shelf" template generator and editor for Eclipse.

c) **Argo UML** Argo UML is a UML modeling tool Star open reputable source and includes support for all standard UML 1.4 diagrams. It runs on any Java platform and is available in ten languages.

2.9.6. SQLite

The database systems used by Android are SQLite. SQLite is a SQL-based database, but it has many limitations, one of which is the provision of data types is limited and does not support the concept of normalization. Database created can only be accessed by the application itself. And for what we use SQLite? Commonly used to store data temporarily. While normally used for permanent data web service.

SQLite is an embedded SQL database engine. Unlike most other SQL databases, SQLite does not have a separate server process. SQLite reads and writes directly to ordinary disk files. A complete SQL database with multiple tables, indexes, triggers, and views, contained in a single disk file. Database file format is cross-platform - you can freely copy a database between 32-bit and 64-bit or between big-endian and little-endian architectures. These features make SQLite a popular choice as an Application File Format. Think of SQLite not as a replacement for Oracle but as a replacement for fopen ()

2.10. System Development Life Cycle (SDLC)

The **systems development life cycle** (**SDLC**), also referred to as the **a**pplication development life-cycle, is a term used in systems engineering, information systems and software engineering to describe a process for planning, creating, testing, and deploying an information system. The systems development life-cycle concept applies to a range of hardware and software configurations, as a system can be composed of hardware only, software only, or a combination of both.

A systems development life cycle is composed of a number of clearly defined and distinct work phases which are used by systems engineers and systems developers to plan for, design, build, test, and deliver information systems. Like anything that is manufactured on an assembly line, an SDLC aims to produce high quality systems that meet or exceed customer expectations, based on customer requirements, by delivering systems which move through each clearly defined phase, within scheduled time-frames and cost estimates. Computer systems are complex and often (especially with the recent rise of service-oriented architecture) link multiple traditional systems potentially supplied by different software vendors. To manage this level of complexity, a number of SDLC models or methodologies have been created, such as: Waterfall, Fountain, Spiral, Build and Fix, Agile Software Development, Rapid Prototype, Incremental and Synchronize and Stabilize.

2.10.1. Major Phase of System Development Life Cycle

The System Development life cycle is the description of the complete process of developing a system. The system development life cycle framework provides a sequence of activities for system designers and developers to follow. It consists of a set of steps or phases in which each phase of the SDLC uses the results of the previous one.

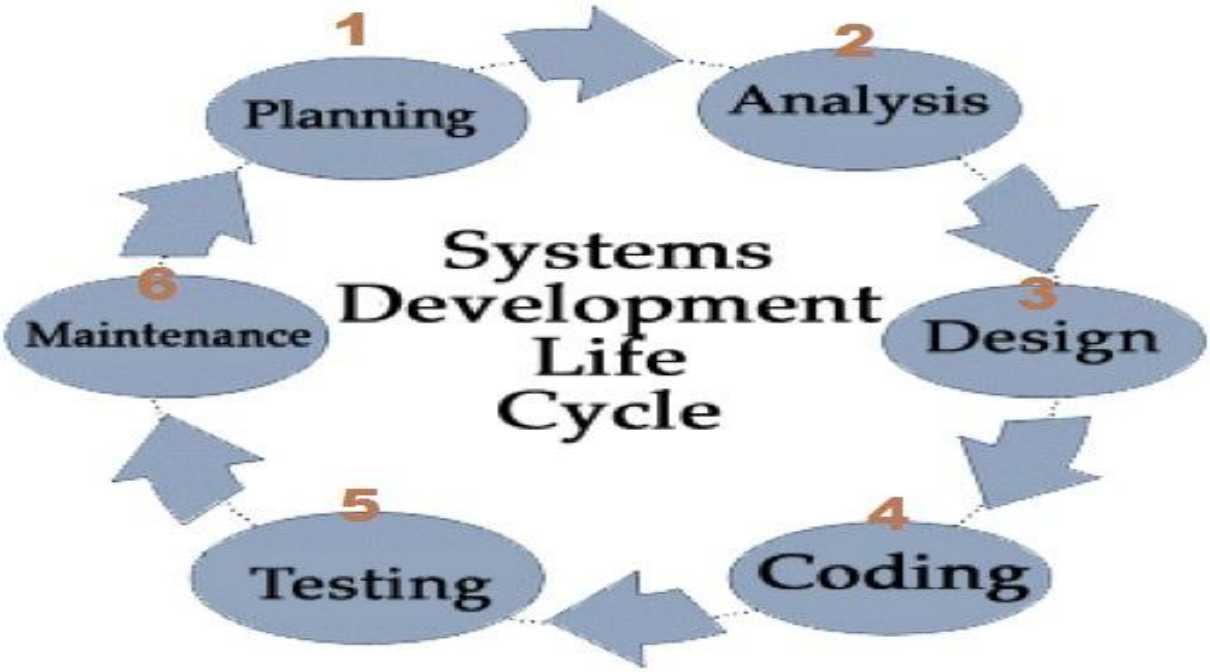


Figure 2.3.: The 6 main phase of System Development Life Cycle (SDLC) (James A. O'Brien, George M. Marakas, "Management information systems", 8th edition, McGraw Hill, 2008.)

2.10.2. Phases of System Development Life Cycle

The system development life cycle framework provides a sequence of activities for system designers and developers to follow. It consists of a set of steps or phases in which each phase of the SDLC uses the results of the previous one.

The SDLC adheres to important phases that are essential for developers, such as ***planning, analysis***, ***design*** and ***implementation***, and are explained in the section below. It includes evaluation of present system, information gathering, and feasibility study and request approval. These stages can be characterized and divided up in different ways, including the following:

1. **Preliminary analysis**: The objective of phase 1 is to conduct a preliminary analysis, propose alternative solutions, describe costs and benefits and submit a preliminary plan with recommendations.

   Conduct the preliminary analysis: in this step, you need to find out the organization's objectives and the nature and scope of the problem under study. Even if a problem refers only to a small segment of the organization itself then you need to find out what the objectives of the organization itself are. Then you need to see how the problem being studied fits in with them.

   Propose alternative solutions: In digging into the organization's objectives and specific problems, you may have already covered some solutions. Alternate proposals may come from interviewing employees, clients, suppliers, and/or consultants. You can also study what competitors are doing. With this data, you will have three choices: leave the system as is, improve it, or develop a new system.

Describe the costs and benefits.

2. **Systems analysis, requirements definition**: Defines project goals into defined functions and operation of the intended application. Analyzes end-user information needs.
3. **Systems design**: Describes desired features and operations in detail, including screen layouts, Business rules, process diagrams, pseudo code and other documentation.
4. **Development**: The real code of the system developed is written here completely in this stage of system development.
5. **Integration and testing**: Brings all the pieces together into a special testing environment, then checks for errors, bugs and interoperability.

   Acceptance, installation, deployment: The final stage of initial development, where the software is put into production and runs actual business.
6. **Maintenance**: During the maintenance stage of the SDLC, the system is assessed to ensure it does not become obsolete. This is also where changes are made to initial software. It involves continuous evaluation of the system in terms of its performance.

   Evaluation: Some companies do not view this as an official stage of the SDLC, but is it an important part of the life cycle. Evaluation step is an extension of the Maintenance stage, and may be referred to in some circles as Post-implementation Review. This is where the system that was developed, as well as the entire process, is evaluated. Some of the questions that need to be answered include: does the newly implemented system meet the initial business requirements and objectives?

Is the system reliable and fault-tolerant? Does the system function according to the approved functional requirements? In addition to evaluating the software that was released, it is important to assess the effectiveness of the development process. If there are any aspects of the entire process, or certain stages, that management is not satisfied with, this is the time to improve. Evaluation and assessment is a difficult issue. However, the company must reflect on the process and address weaknesses.

Disposal: In this phase, plans are developed for discarding system information, hardware and software in making the transition to a new system. The purpose here is to properly move, archive, discard or destroy information, hardware and software that is being replaced, in a matter that prevents any possibility of unauthorized disclosure of sensitive data. The disposal activities ensure proper migration to a new system. Particular emphasis is given to proper preservation and archival of data processed by the previous system. All of this should be done in accordance with the organization's security requirements.