

System analysis and design



Process modeling is a technique for documenting and organizing the structure as well as the flow of data through a system's processes and the procedures, logic and policies to be implemented by a system's process. This highlights the fact that system development life cycle is a process model that is adopted and followed during the creation of a computer application (Giordano, 2011). Process modeling aims to graphically highlight the processes which capture, distribute, store, and manipulate data during SDLC. Process modeling is an important part of SDLC in the sense that it helps the project team to study the problem domain better, to understand the intended system's desired behavior and functionality, to understand the systems components and their relationships, and understand the system's interaction with end users and other systems. Second, process modeling plays a vital role in SDLC in the sense that it facilitates the communication between the various stakeholders of the system, including end-users, customers, system analysts, domain experts and software developers considering the fact that these stakeholders have different expectations from the system (Roebuck, 2011). This means that process modeling is one way through which the project team can contract the design-reality gap.

A data flow diagram is a common process modeling tool that is used in SDLC. A data flow diagram is a picture representation of system processes and data that pass among them. There are four symbols namely process, data flow, data store and external entities that form a data flow diagram. A process is a symbol that depicts an activity that does something. Every process has a verb phrase or name, and it often has at least one input as well as an output. A data flow is an object or piece of data and has a description

and name. A data flow either starts or ends at a process. On the other hand, a data store is a computer file or manual and it has a name and a number. A data flow has at least one input and an output data flow.

System Development Life Cycle

By definition, System Development Life Cycle is a process or a conceptual model that describes the stages involved in developing or changing computer information systems from an initial feasibility study to the final maintenance phase of a completed computer application (Giordano, 2011).

Project planning, feasibility study

This is the first stage of the system development life cycle. During this stage, the management together with the project team develops a high-level view of the intended system in order to determine its objectives and goals (Roebuck, 2011). It is at this stage that the project team establishes the current and future needs in the market, so as to understand what to create.

System analysis, requirement definition

Once the management has approved the project, the project team breaks down the project objectives and goals into defined functions and operations. It is at this stage that all the information regarding end-user needs is analyzed. Design

The architecture of the system is developed at this stage. The project team establishes the features and operations of the system in detail. This includes, creating process and workflow diagrams, screen layout, pseudo codes, business rules and other documentations. Roebuck (2011) argues <https://assignbuster.com/system-analysis-and-design-analysis-paper-samples-3/>

that the design is intended to give a skeletal structure of what the end product will operate and look like. Testing

This stage of the SDLC is intended to ensure that the end-product works well and to the maximum in various environments. In order to ensure that the system works well, the project team tests the program. At this stage, all pieces are brought together into a special testing environment where bugs, errors, and interoperability are checked.

Implementation

Once the system has been tested and is deemed to be good and working, it is released to the end-user. Similar to the testing stage, the project team often experiences many challenges, such as fixing bugs during the implementation stage. The system can be implemented in two ways. First, it can be implemented so that it works hand-in-hand with the old system. Secondly, it can be implemented at once where the old system is phased out, whereas the new system is implemented. Maintenance

For a new system to work well and maximally effective, it must be maintained constantly. The developers, in conjunction with the users, are expected to work reactively on the new system by making changes, corrections, additions and performing continuous upgrades. It has been established that for the end-users to enjoy the benefits of a new system, it must be maintained throughout its lifespan.