Assignments

Name **:**  Aayushi Sethi

Class **:** Bca IV sem

Subject **:**  Software Engineering

College **:** MKHS Gujarati Girls College Indore

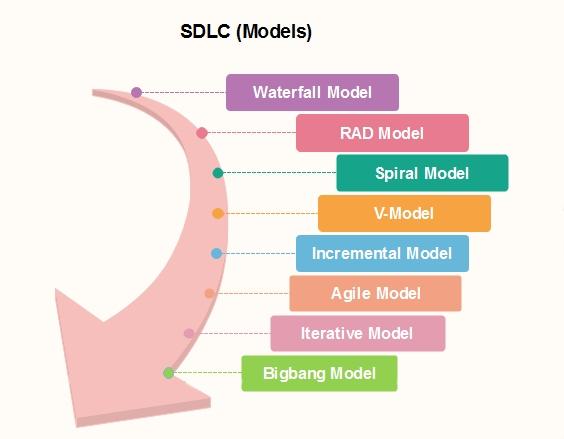
**INDEX**

|  |  |  |
| --- | --- | --- |
| **S.No.** | **Topic** | **Page No.** |
| 1. | SDLC Models | 2 |
| 2. | The Product | 6 |
| 3. | Software Design | 9 |
| 4. | Software Testing | 13 |

Assignment 1

**System Development Life Cycle**

* It is also known as Software Development Process.
* System development life cycle is a process used to design, develop, & test high quality software.
* It consists of a detailed plan describing how to develop, maintain, replace and alter or enhance specific software.
* The life cycle defines methodology for improving the quality of software and overall development process.



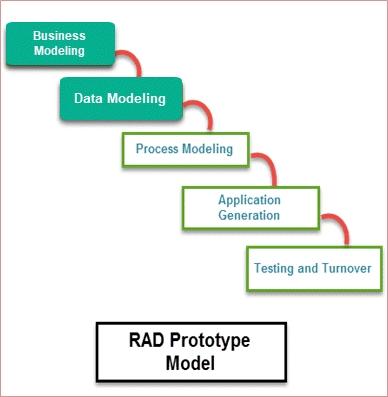
**RAD Model**

* It is also known as Rapid Application Development.
* It is based on prototyping and iterative development with no specific planning involved.
* It targets at developing software in a short span of time.
* The most important aspect of this model to be successful is to make sure that the prototype developed are reusable.

**When To Use RAD** **Methodology**

* When a system needs to be produced in a short span of time.
* When the requirements are known.
* When the user will be involved all through the life cycle.
* When technical risk is less.
* When there is a necessity to create a system in 2-3 month of time.
* When a budget is high enough to afford designers for modelling along with the cost of automated tools for code generation.

**Design Phases Of RAD MODEL**



**Different Phases Of RAD Model**

* **Business Modelling :** On basis of flow of information and distribution between various business channels, the product is designed.
* **Data Modeling :**  The information collected from business modelling is refund into a set of data objects that are significant for the business.
* **Process Modelling :**  The data object that is declared in the data modeling phase is transformed to achieve the information flow necessary to implement a business function.

**Different Phases Of RAD Model**

* **Application Generation :** Automated tools are used for the construction of the software, to convert process and data models into prototypes.
* **Testing and Turnover :** As prototypes are individually tested during every iterations, the overall testing time is reduced in RAD.

**Advantages**

* RAD model is flexible & adaptable to changes.
* It’s used when you have to reduce the overall project risk.
* Increases reusability of components.
* Due to prototyping in nature, there is possiblity of lesser defects.
* Each phase in RAD delivers highest priority functionality to client.
* With less people, productivity can be increased in short time.

**Disadvantages**

* Requires highly skilled developers / designers.
* Management complexity is more.
* High dependency on modelling skills.
* Requires user involvement throughout the life cycle.
* Can’t be used for smaller projects.
* Not all applications is compatible with RAD.
* If developers are not committed to delivering software on time, RAD projects can fail.

**Applications Of RAD**

* RAD should be used only when a system can be modularized to be delivered in an incremental manner.
* It should be used if there is a high availability of designers for modelling.
* It should be used only the budget permits use of automated code generating tools.

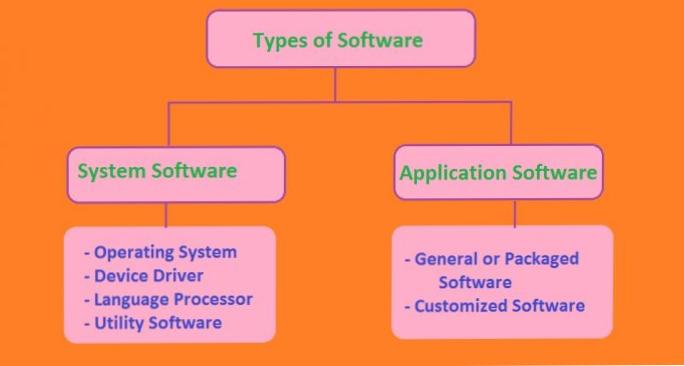
**Assignment 2**

**WHAT IS SOFTWARE ?**

* A software is a part of computer system that consists of encoded information or computer instructions.
* It is collection of which we give to computer to perform specific task.
* A software is considered to be collection of executable programming code, associated libraries and documentation. Software is made for a specific requirement is called software product.

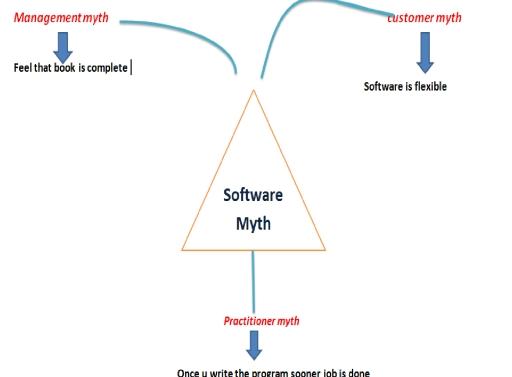
**TYPES OF SOFTWARE**

1. ***Application Software* :** It perform specific function in order to satisfy user needs. Eg. Accounting software, Database Management System, etc.
2. ***System software* :** It is responsible to provide hardware resources to the application software. It also manages and control all the system resources.
3. ***Platform software* :** Platform software is not readily available in the system. It is required when we want to use special hardware such as printer, scanner, etc.

****

**SOFTWARE MYTH’S**

* Software myths are wrong belief or misinformation about software and the process used to build it.
* All people who come into contact with software may suffer from various associated with developing and using software.
* Myths have number of attributes that causes serious problems on software.

****

**TYPES OF SOFTWARE MYTH’S**

***Customer Myth’s :*** Software myth’s believed by customer leads to false expectations and ultimately, dissatisfaction with the developer.

* Customer believe that general statement of objective is sufficient to begin writing program.
* Requirement changes are easy to accommodate because software is flexible.
* Customer always think that software development is an easy process.

***Management Myth’s* :** This provides everything that our programmers and managers need to know.

* Standard procedures for building software, so develpers have everything they need to know.
* They think they have latest computers.
* We can add more programs to catch up.
* A good manager can manage any project.

***Developers Myth’s* :** The job is done when the code is delivered thus, the project success depends solely on the quality of the delivered program.

* If I miss something now i can fix it later.
* Once the program is written and running, my job is done.
* Until is program is running their is no way of accessing its quality.
* The only deliverable in a software project is a working program.

**Assignment 3**

**Software Designing**

* Software designing is a process of problem solving and planning for a software solution. After the purpose and specifications of software are determined, software developers will design or employ designers to develop a plan for a solution. It includes low level components and algorithm implementation issues as well as the architectural view.

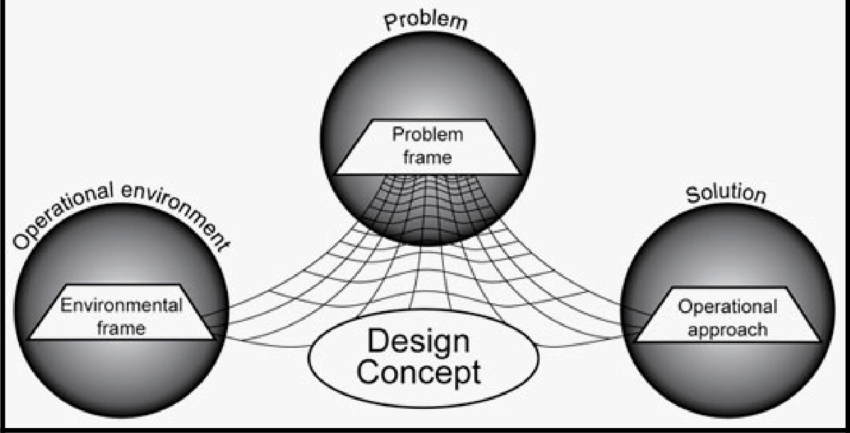
**Process of Design**

1. **Logical Design :**  It is shown by DFD flow charts, which defines the boundaries for a candidate system. It describes the input, output. Databases and procedures ( dataflow ) all in a format that meets the user requirements. It specify user needs at a level of detail and determines the information flow into and out of the system and required data source.

**2) Physical Design :** It defines the specification that tell the Exactly what the candidate system do ? That is the Programmer produces the working system by writing the Necessary program or modifies the software, packages that Accept the input from user, perform the necessary Calculations produces the reports on a hard copy or soft Copy and maintains an updated data base at all times.

**Design Methodologies**

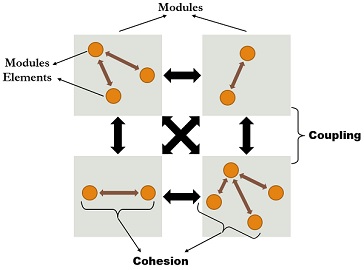
1. Improve productivity of analyst / programmer.
2. Improve documentation, maintenance and enhancement.
3. Cut down / minimize the cost over.
4. Improve communication among the user, analyst, designer and programmer.
5. Standardize the approach to analysis and design.
6. Simplify design by segmentation.



**Structured Design**

* It is basically concerned with the solution design. It is based on ‘divide and conquer’ strategy where a problem is broken into several small problems & each small problem is solved.
* The small pieces of problem are solved by means of solution modul. Structured design emphasis that these modules will organized in order to achieve precise solution.
* These modules are arranged in hierarchy. They communicate with each other. A good structured design always follow some rules for communication among multiple modules.

**Coupling And Cohesion**

* **Cohesion :** Cohesion is a measure that defines the degree of intra dependability within elements of the module. The greater the cohesion, the better the program design.
* ** Coupling :** Coupling is measure that defines the degree of inter dependability among modules of the program. It tells at what level the modules interfere and interact with each other. The lower the coupling, better the program.

**Data Flow Diagram ( DFD )**

* A data flow diagram is a graphical representation of the ‘flow’ of data through an information system. DFD’s can be also used for the visualization of data processing.
* DFD doesn’t mention anything about how data flows through the system.
* There is a quiet difference between DFD and flowchart, the flowchart shows the flow of control in program module. DFD shows flow of data in the system at various levels.
* DFD doesn’t contain any control and branch element.

**Symbols Used in DFD**

* **Square :** It defines source or destination of system data.
* **Arrow :** It defines data flows that is it a pipe lines to

which information flows.

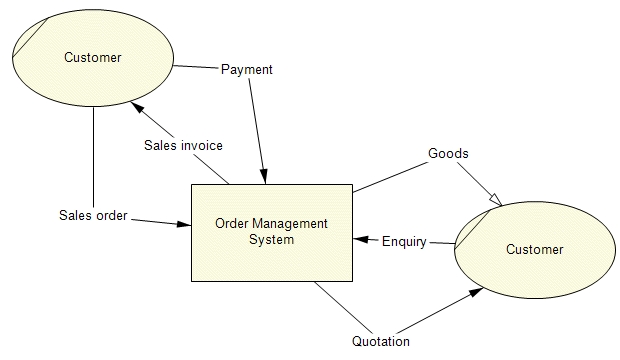
* **Circle :**  It represent a process that transforms incoming

Data flow into outgoing data flow.

* **Open Rectangle :** It represent data storage, files**.**

**Steps for making DFD**

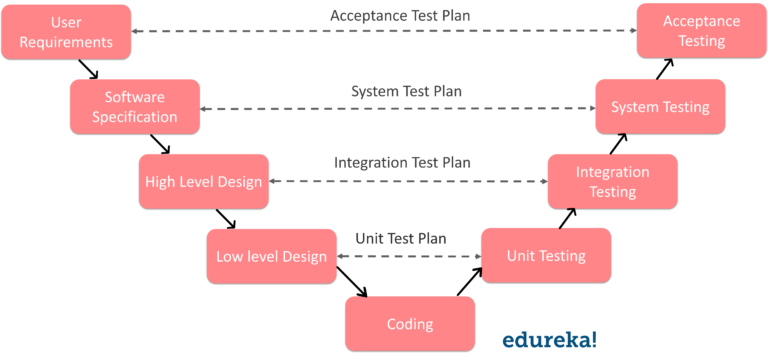
* **Step 1 :** All process in the DFD should be name & number. Each name should be relevent & representative of the process.
* **Step 2 :** The direction of the data flow is from top to bottom and from left to right.
* **Step 3 :** When the process is divided into lower details (subprocess) they are numbers as sub divided.
* **Step 4 :** The names of data stores, sources and destination are written in capital letters. Process & data flow names have the first letter of each word is capitalize.

****

**Assignment 4**

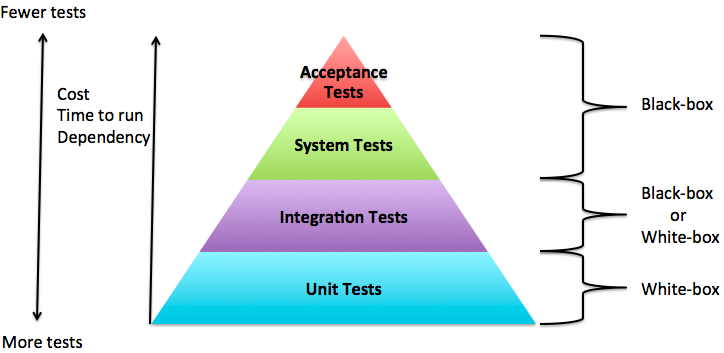
**SOFTWARE TESTING**

* Software testing is a process of evaluation a software item to detect difference between given input and the expected output.
* Also to access the features of a software item. Testing access the quality of the product.
* Software testing is the process that is a process that should be done during the development process, in the other words, It is a varification and validation process.

****

**VARIOUS SOFTWARE TESTING METHODS**

* Acceptance Testing
* System Testing
* Smoke Testing
* Integration Testing
* Unit Testing
* Gray Box Testing
* WHite Box Tesing
* Black Box Testing
* Validation Testing

****

**UNIT TESTING**

* Unit testing is performed by developers before the setup is handed over to the testing team to formally execute the test case.
* It is performed by the respective developers on the individual units of source code assigned area. The test developers use test data that is different from the test data of the quality assurance team.
* The goal of unit testing is isolate each part of the program and show that individual parts are correct in terms of requirements and functionality.

**INTERGRATION TESTING**

* It is defined as the testing of combined parts of an application to determined if they function correctly.
* It can be done in two ways -

Bo**ttom up integration :** This testing with unit testing followed by testing of progressively higher level combinations of units called modules or builds.

**Top down integration** : In this testing the highest level modules are

tested first & progressively, lower level modules are tested thereafter**.**

**SYSTEM TESTING**

* System testing test the system as a whole, once all the components are integrated, the application as a whole is tested rigorously to see that it meets the specified quality standards. this type of testing is performed by a specialized testing team.

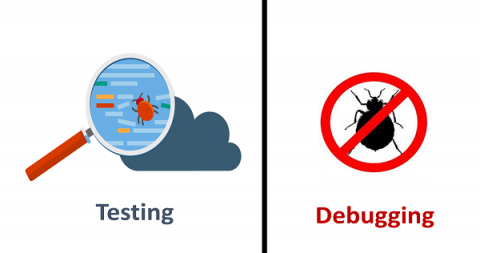
**WHITE BOX TESTING**

* White box testing is the detailed investigation of internal logic & structure of the code.
* It is also known as open box testing. In order to perform white box testing on an application, a tester needs to known the internal working of the code.
* The tester needs to have a look inside the source code and find out which unit/ chunk of the code is behaving inappropriately.

**BLACK BOX TESTING**

* The technique of testing without having any knowledge of the interior working of the application is called black box testing.
* The tester is oblivious to the system architecture and does not have access to the source code.
* While performing a black box test, a tester will interact with the system's user interface by providing inputs & examine outputs are worked upon.

|  |  |
| --- | --- |
| **White Box Testing** | **Black Box Tesing** |
| The code remains visble to the testers. | The code remains hidden from testers. |
| It can be called Structural Testing | It can be called as Functional Testing. |
| It can be called Clear Box Testing. | It can be called Closed Testing. |
| It is low level testing. | It is high level testing. |
| Generally done by developers. | Generally done by independent testers. |
| Detailed design document is required. | Specification document is required. |
| Programming knowledge & implementations are required. | Programming knowledge & implementations are not required. |
| Testing tools depend on programming  language. | Testing tools are independent of programming language. |

****

**SQA v/s Testing**



|  |  |
| --- | --- |
| **Software Quality Assurance** | **Software Testing** |
| SQA is about engineering process that ensures quality. | It is to test a product for problems before the product goes live. |
| It involves activities related to the implementation of processes, procedures, standards. | Involves activities concerning verification of product. |
| Process focused. | Product focused. |
| Preventative technique. | Corrective technique. |
| Proactive measure. | Reactive measure. |
| Scope of SQA applied to all products that ell be created by organizing. | Scope of Software testing applies to a particular product being tested. |