**Practical 1**

**Aim:- Detailed study of various Software Development Lifecycle (SDLC) Models.**

|  |  |
| --- | --- |
| 1. **Classical waterfall model** | |
| **Block Diagram** | https://i2.wp.com/s3.amazonaws.com/production-wordpress-assets/blog/wp-content/uploads/2016/12/08151155/waterfall-model.png?fit=604%2C270&ssl=1 |
| **Description** | 1. **Requirement Analysis & Definition:** This phase is focused on possible requirements of the system for the development are captured. Requirements are gathered subsequent to the end user consultation. 2. **System & Software Design:** Prior to beginning the actual coding, it is inevitable to understand what actions are to be taken and what they should like. The requirement specifications are studied in detail in this phase and the design of the system is prepared. The design specifications are the base for the implementation and unit testing model phase. 3. **Implementation & Unit Testing**: Subsequent to receiving the system design documents, the work is shared into various modules and the real coding is commenced. The system is developed into small coding units. These units are later integrated in the subsequent phase. Every unit is tested for its functionality. 4. **Integration & System Testing:** The modules that are divided into units are integrated into a complete system and tested for proper coordination among modules and system behaves as per the specifications. Once the testing is completed, the software product is delivered to the customer. 5. **Operations & Maintenance:** It is a never ending phase. Once the system is running in production environment, problems come up. The issues that are related to the system are solved only after deployment of the system. The problems arise from time to time and need to be solved; hence this phase is referred as maintenance. |
| **Advantages** | * This model is simple and easy to understand and use. * It is easy to manage due to the rigidity of the model – each phase has specific deliverables and a review process. * In this model phases are processed and completed one at a time. Phases do not overlap. * Waterfall model works well for smaller projects where requirements are very well understood. |
| **Disadvantages** | * Once an application is in the [testing](http://istqbexamcertification.com/what-is-a-software-testing/) stage, it is very difficult to go back and change something that was not well-thought out in the concept stage. * No working software is produced until late during the life cycle. * High amounts of risk and uncertainty. * Not a good model for complex and object-oriented projects. * Poor model for long and ongoing projects. * Not suitable for the projects where requirements are at medium to high risk of changing. |
| **When/where to use?** | * This model is used only when the requirements are very well known, clear and fixed. * Product definition is stable. * Technology is understood. * There are no ambiguous requirements * Ample resources with required expertise are available freely * The project is short. |
| 1. **Incremental Process Model** | |
| **Block Diagram** |  |
| **Description** | 1. **Requirement Analysis:** Requirement and specification of the software are collected 2. **Design:** Some high-end function are designed during this stage 3. **Code**: Coding of software is done during this stage 4. **Test:** Once the system is deployed, it goes through the testing phase |
| **Advantages** | * Generates working software quickly and early during the software life cycle. * This model is more flexible – less costly to change scope and requirements. * It is easier to test and debug during a smaller iteration. * In this model customer can respond to each built. * Lowers initial delivery cost. * Easier to manage risk because risky pieces are identified and handled during it’d iteration. |
| **Disadvantages** | * Needs good planning and design. * Needs a clear and complete definition of the whole system before it can be broken down and built incrementally. * Total cost is higher than [waterfall](http://istqbexamcertification.com/what-is-waterfall-model-advantages-disadvantages-and-when-to-use-it/). |
| **When to use Model?** | * This model can be used when the requirements of the complete system are clearly defined and understood. * Major requirements must be defined; however, some details can evolve with time. * There is a need to get a product to the market early. * A new technology is being used * Resources with needed skill set are not available * There are some high risk features and goals. |
| 1. **Prototype Model** | |
| **Block Diagram** |  |
| **Description** | 1. **Requirements gathering and analysis:** A prototyping model begins with requirements analysis and the requirements of the system are defined in detail. The user is interviewed in order to know the requirements of the system. 2. **Quick design:** When requirements are known, a preliminary design or quick design for the system is created. It is not a detailed design and includes only the important aspects of the system, which gives an idea of the system to the user. A quick design helps in developing the prototype. 3. **Build prototype:** Information gathered from quick design is modified to form the first prototype, which represents the working model of the required system. 4. **User evaluation:** The proposed system is presented to the user for thorough evaluation of the prototype to recognize its strengths and weaknesses such as what is to be added or removed. 5. **Refining prototype:** Once the user evaluates the prototype and if he is not satisfied, the current prototype is refined according to the requirements. 6. **Engineer product:**The final system is evaluated thoroughly followed by the routine maintenance on regular basis for preventing large-scale failures and minimizing downtime. |
| **Advantages** | * Users are actively involved in the development * Since in this methodology a working model of the system is provided, the users get a better understanding of the system being developed. * Errors can be detected much earlier. * Quicker user feedback is available leading to better solutions. * Missing functionality can be identified easily * Confusing or difficult functions can be identified Requirements validation, Quick implementation of, incomplete, but functional, application. |
| **Disadvantages** | * Leads to implementing and then repairing way of building systems. * Practically, this methodology may increase the complexity of the system as scope of the system may expand beyond original plans. * Incomplete application may cause application not to be used as the full system was designed Incomplete or inadequate problem analysis. |
| **When to use Model?** | * Prototype model should be used when the desired system needs to have a lot of interaction with the end users. * Typically, online systems, web interfaces have a very high amount of interaction with end users, are best suited for Prototype model. It might take a while for a system to be built that allows ease of use and needs minimal training for the end user. * Prototyping ensures that the end users constantly work with the system and provide a feedback which is incorporated in the prototype to result in a useable system. They are excellent for designing good human computer interface systems. |
| 1. **Spiral Model** | |
| **Block Diagram** |  |
| **Description** | 1. **Planning Phase:** Requirements are gathered during the planning phase. 2. **Risk Analysis:** In the **risk analysis phase**, a process is undertaken to identify risk and alternate solutions.  A prototype is produced at the end of the risk analysis phase. If any risk is found during the risk analysis then alternate solutions are suggested and implemented. 3. **Engineering Phase:** In this phase software is **developed**, along with testing at the end of the phase. Hence in this phase the development and testing is done. 4. E**valuation phase:** This phase allows the customer to evaluate the output of the project to date before the project continues to the next spiral. |
| **Advantages** | * High amount of risk analysis hence, avoidance of Risk is enhanced. * Good for large and mission-critical projects. * Strong approval and documentation control. * Additional Functionality can be added at a later date. * Software is produced early in the [software life cycle](http://istqbexamcertification.com/what-are-the-software-development-life-cycle-phases/). |
| **Disadvantages** | * Can be a costly model to use. * Risk analysis requires highly specific expertise. * Project’s success is highly dependent on the risk analysis phase. * Doesn’t work well for smaller projects. |
| **When to use Model?** | * When costs and risk evaluation is important * For medium to high-risk projects * Long-term project commitment unwise because of potential changes to economic priorities * Users are unsure of their needs * Requirements are complex * New product line * Significant changes are expected (research and exploration) |
| 1. **Rapid Application Process Model** | |
| **Block Diagram** |  |
| **Description** | 1. **Requirement Planning:** To collect the all requirements. 2. **User Description:** To join team of user and developers to give reviews, changes the requirement. 3. **Construction:** Design+Implementation+Testing 4. **Cutover:** To install the software. User training will be done in this phase. |
| **Advantages** | * Reduced development time. * Increases reusability of components * Quick initial reviews occur * Encourages customer feedback * Integration from very beginning solves a lot of integration issues. |
| **Disadvantages** | * Depends on strong team and individual performances for identifying business requirements. * Only system that can be modularized can be built using RAD * Requires highly skilled developers/designers. * High dependency on modeling skills |
| **When to use model?** | * RAD should be used when there is a need to create a system that can be modularized in 2-3 months of time. * It should be used if there’s high availability of designers for modeling and the budget is high enough to afford their cost along with the cost of automated code generating tools. |

**LAB Assignment:**

**1)**.**Write one example of software project that would be amenable to the classical waterfall model**.

Ans:- Bridges, airplanes, and large-scale engineering projects. Software projects that have a major dependence on hardware (the software depends on the underlying hardware) are examples. Drivers, embedded controllers, medical devices, etc.

**2) What** **is the difference between incremental process model and evolutionary process model?**

|  |  |  |
| --- | --- | --- |
| **#** | **Incremental Process Model** | **Evolutionary Process Model** |
| 1 | Work on the concept of developing software in small instalments where each instalment adds something new to its previous instalment. | It also Works on the concept of developing software in small instalments where each instalment adds something new to its previous instalment. |
| 2 | Needs a clear and complete definition of the whole system before it can be broken down and built incrementally. | Risk analysis is better. |
| 3 | Easier to manage risk because risky pieces are identified and handled during it’d iteration. | End of project may not be know which is a risk. |
| 4 | Works well for smaller projects. | Doesn’t work well for smaller projects. |
| 5 | Project’s success is dependent on clear and complete definition of the whole system before it can be broken down and built incrementally. | Project’s success is highly dependent on the risk analysis phase. |

**3) Which model is more suitable for your mini project? Why?**

Ans: For my mini project of Online Banking System Prototype Model because Prototype model should be used when the desired system needs to have a lot of interaction with the end users. 1) Typically, online systems, web interfaces have a very high amount of interaction with end users, are best suited for Prototype model. It might take a while for a system to be built that allows ease of use and needs minimal training for the end user. 2) Prototyping ensures that the end users constantly work with the system and provide a feedback which is incorporated in the prototype to result in a useable system. They are excellent for designing good human computer interface systems.

**4) State the parameters to choose suitable life cycle model for any system.**

Ans:- It is also essential to consider time and cost while choosing a process model as cost and/ or time constraints play an important role in software development. The basic characteristics required to select the process model are project type and associated risks, requirements of the project, and the users.

**5)** **Which life cycle model is/are widely used in software industry nowadays?**

Ans:- Now a days most Widely Used SDLC Model in Industries is Agile Model. In Agile Model there is a Story, Scrums. This model is used for short release like release in 15 -20 days. Requirement can be changes at any phase. No more Documentation in that.

**6). Give the comparison of all life cycle models with the options given.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Properties of Model** | **Water-Fall Model** | **Incremental Model** | **Spiral Model** | **Rad Model** |
| Planning in early stage  (Yes/No) | Yes | No | Yes | Yes |
| Returning to an earlier phase  (Yes/No) | No | Yes | Yes | No |
| Handle Large-Project  (Yes/No) | No | Yes | Yes | Yes |
| Requirement Specifications  Beginning/Intermediate/last) | Beginning | Beginning | Beginning | Beginning |
| Cost  (Cost-effective/expensive) | Cost-effective | Cost-effective | Cost-effective | Expensive |
| Flexibility to change  (Easy/difficult) | Difficult | Easy | Easy | Easy |
| Duration  (Short/long/very long) | Long | Short | Long | Very long |