**Assignment 3**

**Research and compare SDLC models suitable for engineering projects present findings on waterfall, agile, spiral and v model approaches, emphasizing their advantages, disadvantages, and applicability, in different engineering contexts.**

**Waterfall Model**

Software Development life cycle (SDLC) is a spiritual model used in project management that defines the stages include in an information system development project, from an initial feasibility study to the maintenance of the completed application.

There are different software development life cycle models specify and design, which are followed during the software development phase. These models are also called "**Software Development Process Models**." Each process model follows a series of phase unique to its type to ensure success in the step of software development.

Winston Royce introduced the Waterfall Model in 1970.This model has five phases: Requirements analysis and specification, design, implementation, and unit testing, integration and system testing, and operation and maintenance. The steps always follow in this order and do not overlap. The developer must complete every phase before the next phase begins. This model is named "**Waterfall Model**", because its diagrammatic representation resembles a cascade of waterfalls.

**1. Requirements analysis and specification phase:** The aim of this phase is to understand the exact requirements of the customer and to document them properly. Both the customer and the software developer work together so as to document all the functions, performance, and interfacing requirement of the software. It describes the "what" of the system to be produced and not "how. “In this phase, a large document called **Software Requirement Specification (SRS)** document is created which contained a detailed description of what the system will do in the common language

**2. Design Phase:** This phase aims to transform the requirements gathered in the SRS into a suitable form which permits further coding in a programming language. It defines the overall software architecture together with high level and detailed design. All this work is documented as a Software Design Document (SDD).

**3. Implementation and unit testing:** During this phase, design is implemented. If the SDD is complete, the implementation or coding phase proceeds smoothly, because all the information needed by software developers is contained in the SDD.

During testing, the code is thoroughly examined and modified. Small modules are tested in isolation initially. After that these modules are tested by writing some overhead code to check the interaction between these modules and the flow of intermediate output.

**4. Integration and System Testing:** This phase is highly crucial as the quality of the end product is determined by the effectiveness of the testing carried out. The better output will lead to satisfied customers, lower maintenance costs, and accurate results. Unit testing determines the efficiency of individual modules. However, in this phase, the modules are tested for their interactions with each other and with the system.

**5. Operation and maintenance phase:** Maintenance is the task performed by every user once the software has been delivered to the customer, installed, and operational.

**Advantages:**

* Sequential Approach: It follows a step-by-step progression, making it easier to understand and manage.
* Clear Documentation: Each phase requires detailed documentation, aiding in project understanding and maintenance.
* Easy to Manage: Progress is measurable at each stage, making it easier to manage and track.
* Stable Requirements: Well-defined requirements upfront reduce the risk of scope creep.

**Disadvantages:**

* **Rigid Structure:** Lack of flexibility, making it challenging to accommodate changes once a phase is completed.
* **Long Delivery Time:** The linear progression can result in longer delivery times, especially for large-scale projects.
* **Client Involvement:** Limited client involvement until the later stages, which can lead to misunderstandings or unmet expectations.
* **Testing at the End:** Testing is usually done towards the end, which can lead to late identification of defects.

**Applicability:**

Waterfall is suitable for projects with stable and well-defined requirements, where changes are unlikely and the technology is well understood. It's commonly used in traditional engineering projects like construction, manufacturing, and infrastructure development.

**Agile Model**

The meaning of Agile is swift or versatile. **“Agile process model**" refers to a software development approach based on iterative development. Agile methods break tasks into smaller iterations, or parts do not directly involve long term planning. The project scope and requirements are laid down at the beginning of the development process. Plans regarding the number of iterations, the duration and the scope of each iteration are clearly defined in advance.

Each iteration is considered as a short time "frame" in the Agile process model, which typically lasts from one to four weeks. The division of the entire project into smaller parts helps to minimize the project risk and to reduce the overall project delivery time requirements. Each iteration involves a team working through a full software development life cycle including planning, requirements analysis, design, coding, and testing before a working product is demonstrated to the client.

1. **Requirements gathering:** In this phase, you must define the requirements. You should explain business opportunities and plan the time and effort needed to build the project. Based on this information, you can evaluate technical and economic feasibility.
2. **Design the requirements:** When you have identified the project, work with stakeholders to define requirements. You can use the user flow diagram or the high-level UML diagram to show the work of new features and show how it will apply to your existing system.
3. **Construction/ iteration:** When the team defines the requirements, the work begins. Designers and developers start working on their project, which aims to deploy a working product. The product will undergo various stages of improvement, so it includes simple, minimal functionality.
4. **Testing:** In this phase, the Quality Assurance team examines the product's performance and looks for the bug.
5. **Deployment:** In this phase, the team issues a product for the user's work environment.
6. **Feedback:** After releasing the product, the last step is feedback. In this, the team receives feedback about the product and works through the feedback.

**Advantages:**

* **Flexibility:** Emphasizes adaptive planning and welcomes changes throughout the development process.
* **Customer Collaboration:** Regular interactions with the customer ensure alignment with their needs and expectations.
* **Early Delivery:** Incremental development allows for early delivery of functional components.
* **Continuous Improvement:** Iterative cycles promote continuous improvement and learning within the team.

**Disadvantages:**

* **Complexity:** Managing multiple iterations simultaneously can be complex, requiring skilled project management.
* **Documentation Challenges:** Agile prioritizes working software over comprehensive documentation, which can be a challenge for some projects.
* **Resource Intensive:** Requires active involvement from the customer and the development team, which can be resource-intensive.

**Applicability:**

Agile is suitable for projects with evolving requirements, where quick adaptation to changes is crucial. It's commonly used in software development, but it can also be applied to engineering projects like product development, where requirements may evolve based on user feedback or market changes.

**Spiral Model**

The spiral model, initially proposed by Boehm, is an evolutionary software process model that couples the iterative feature of prototyping with the controlled and systematic aspects of the linear sequential model. It implements the potential for rapid development of new versions of the software. Using the spiral model, the software is developed in a series of incremental releases. During the early iterations, the additional release may be a paper model or prototype. During later iterations, more and more complete versions of the engineered system are produced.

**Each cycle in the spiral is divided into four parts:**

**Objective setting:** Each cycle in the spiral starts with the identification of purpose for that cycle, the various alternatives that are possible for achieving the targets, and the constraints that exists.

**Risk Assessment and reduction:** The next phase in the cycle is to calculate these various alternatives based on the goals and constraints. The focus of evaluation in this stage is located on the risk perception for the project.

**Development and validation:** The next phase is to develop strategies that resolve uncertainties and risks. This process may include activities such as benchmarking, simulation, and prototyping.

**Planning:** Finally, the next step is planned. The project is reviewed, and a choice made whether to continue with a further period of the spiral. If it is determined to keep, plans are drawn up for the next step of the project.

The development phase depends on the remaining risks. For example, if performance or user-interface risks are treated more essential than the program development risks, the next phase may be an evolutionary development that includes developing a more detailed prototype for solving the risks.

The **risk-driven** feature of the spiral model allows it to accommodate any mixture of a specification-oriented, prototype-oriented, simulation-oriented, or another type of approach. An essential element of the model is that each period of the spiral is completed by a review that includes all the products developed during that cycle, including plans for the next cycle. The spiral model works for development as well as enhancement projects.

**Advantages:**

* Risk Management: Emphasizes risk analysis and mitigation throughout the development life cycle.
* Flexibility: Allows for iterations and changes based on feedback and evolving requirements.
* Prototyping: Supports prototyping and early validation of critical components.
* Progressive Development: Enables progressive refinement and improvement with each iteration.

**Disadvantages:**

* **Complexity:** Requires experienced project management and technical expertise to manage the iterative cycles effectively.
* **Resource Intensive:** Involves more effort and resources due to the iterative nature and risk analysis activities.
* **Documentation Challenges:** Like Agile, may face challenges in comprehensive documentation, especially in early iterations.

**Applicability:**

The Spiral model is suitable for projects with high-risk factors or complex requirements that may evolve over time. It's commonly used in engineering projects like aerospace, defense systems, and large-scale infrastructure projects where risk management and iterative development are critical.

**V-Model**

V-Model also referred to as the Verification and Validation Model. In this, each phase of SDLC must complete before the next phase starts. It follows a sequential design process same as the waterfall model. Testing of the device is planned in parallel with a corresponding stage of development.

**Verification:** It involves a static analysis method (review) done without executing code. It is the process of evaluation of the product development process to find whether specified requirements meet.

**Validation:** It involves dynamic analysis method (functional, non-functional), testing is done by executing code. Validation is the process to classify the software after the completion of the development process to determine whether the software meets the customer expectations and requirements.

So, V-Model contains Verification phases on one side of the Validation phases on the other side. Verification and Validation process is joined by coding phase in V-shape. Thus, it is known as V-Model.

**There are the various phases of Verification Phase of V-model:**

1. **Business requirement analysis:** This is the first step where product requirements understood from the customer's side. This phase contains detailed communication to understand customer's expectations and exact requirements.
2. **System Design:** In this stage system engineers analyze and interpret the business of the proposed system by studying the user requirements document.
3. **Architecture Design:** The baseline in selecting the architecture is that it should understand all which typically consists of the list of modules, brief functionality of each module, their interface relationships, dependencies, database tables, architecture diagrams, technology detail, etc. The integration testing model is carried out in a particular phase.
4. **Module Design:** In the module design phase, the system breaks down into small modules. The detailed design of the modules is specified, which is known as Low-Level Design
5. **Coding Phase:** After designing, the coding phase is started. Based on the requirements, a suitable programming language is decided. There are some guidelines and standards for coding. Before checking in the repository, the final build is optimized for better performance, and the code goes through many code reviews to check the performance.

**There are the various phases of Validation Phase of V-model:**

1. **Unit Testing:** In the V-Model, Unit Test Plans (UTPs) are developed during the module design phase. These UTPs are executed to eliminate errors at code level or unit level. A unit is the smallest entity which can independently exist, e.g., a program module. Unit testing verifies that the smallest entity can function correctly when isolated from the rest of the codes/ units.
2. **Integration Testing:** Integration Test Plans are developed during the Architectural Design Phase. These tests verify that groups created and tested independently can coexist and communicate among themselves.
3. **System Testing:** System Tests Plans are developed during System Design Phase. Unlike Unit and Integration Test Plans, System Tests Plans are composed by the client?s business team. System Test ensures that expectations from an application developer are met.
4. **Acceptance Testing:** Acceptance testing is related to the business requirement analysis part. It includes testing the software product in user atmosphere. Acceptance tests reveal the compatibility problems with the different systems, which is available within the user atmosphere. It conjointly discovers the non-functional problems like load and performance defects within the real user atmosphere.

**Advantages:**

* **Emphasizes Testing:** Testing activities are integrated into each phase, ensuring early defect detection.
* **Clear Documentation:** Requires detailed documentation for each phase, aiding in traceability and project understanding.
* **Structured Approach:** Follows a structured and sequential approach, combining elements of both Waterfall and iterative models.

**Disadvantages:**

**Rigidity:** Similar to Waterfall, can be rigid and challenging to accommodate changes once a phase is completed.

**Limited Flexibility:** Less flexible compared to Agile or Spiral models, especially in terms of accommodating evolving requirements.

**Complexity:** Managing testing activities in parallel with development phases can add complexity to the project.

**Applicability:**

The V-Model is suitable for projects with well-defined requirements and a focus on comprehensive testing throughout the development life cycle. It's commonly used in safety-critical systems like medical devices, automotive engineering, and industrial control systems.

In summary, each SDLC model has its strengths and weaknesses, and their applicability depends on the project's specific requirements, risk factors, and stakeholder preferences. Waterfall suits well-defined, stable projects; Agile adapts to evolving requirements; Spiral manages high-risk projects; and V-Model emphasizes comprehensive testing in structured environments.