**CreateTech**

**Software Development Life Cycle Methodologies**

Task 1 The 12SDLC Methodologies

1. Waterfall model - This model is basically a traditional and linear approach to software development. It consists of sequential phases, where each phase must be completed before moving on to the next. The phases usually include requirements gathering, design, implementation, testing, deployment, and maintenance. Once a phase is completed is difficult to go back and make changes, making it less flexible for accommodating changing requirements.

2. Agile – In the context of prototyping, Agile refers to an interactive and incremental software development approach that emphasizes flexibility, collaboration, and customer feedback. Agile methodologies focus on delivering working software in small, frequent increments, which allows for continuous improvements and adjustments based on user feedback. When combining Agile with prototyping, the development process involves creating and refining prototypes in short cycles, typically called sprints or iterations. Each iteration aims to build on the previous one, incorporating user feedback and evolving the prototype until it meets the desired requirements and functionality.

The key principle of Agile that apply to prototyping include:

A. Customer Collaboration: Active involvement of stakeholders and end-users throughout the development process to ensure that the prototype aligns with their needs and expectations.

B. Incremental Development: Prototyping is done small, manageable chunks, allowing for faster validation and improvements.

C. Responding to Change: Agile methods embrace change, so if new insights or requirements emerge during the prototyping process, they can be incorporated without disrupting the entire development cycle.

D. Self-organizing Teams: Cross functional teams collaborate and take collective responsibility for the prototype’s success, promoting better communication and efficiency.

3. Scrum: Scrum is an agile development methodology used in the development of Software based on an iterative and incremental processes. Scrum is adaptable, fast, flexible and effective agile framework that is designed to deliver value to the customer throughout the development of the project.

The primary objective of Scrum is to satisfy the customer’s need through an environment of transparency in communication, collective responsibility and continuous progress. The development starts from a general idea of ​​what needs to be built, elaborating a list of characteristics ordered by priority (product backlog) that the owner of the product wants to obtain.

4. Extreme Programming

Extreme Programming (XP) is a Software Development Methodology intended to improve software quality and responsiveness to change customer requirements. As a type of agile software development, it advocates frequent releases in short development cycles, which is intended to improve productivity and introduce checkpoints at which new customer requirements can be adopted. It is called extreme because it increases efficiency and makes sure that the application of principles and values in an extremely effective manner.

5. RAD

RAD (Rapid Application Development) is a form of agile software development methodology, that prioritizes fast product development. RAD uses frequent iterations and constant feedback that ultimately enables your organization to develop systems faster while maintaining quality and reducing costs. Its end goal is to deliver working software products to market faster as the demand pf new application is increasing rapidly. In practice, RAD puts more emphasis adaptive process instead of planning. The RAD model has 4 development phases are assessing the requirements, prototyping, testing and gathering feedback, and lastly presenting the product.

6. Spiral Model

Spiral Model is a risk-driven software development process model. It is a combination of waterfall model and iterative model. It helps to adopt software development elements of multiple process models for the software project based on unique risk patterns ensuring efficient development process. Each phase of this model in software engineering commences with a design goal and ends with the client reviewing the progress. The process includes planning, risk analysis, engineering and evaluation. Spiral Model is commonly put in use in software engineering when the project is large, when risk and costs evaluation is important, when changes may require at any time, when creation of prototype is applicable and when requirements are unclear and complex.

7. Prototype methodology

A prototype is an elementary visualisation of a product to test its functionalities and the concept. It arose a way to build a product that will ensure the highest degree of satiation from a consumer of the product, and it revolves around the different needs that may arise and the desired goal through four methods of prototyping which include Feasibility prototypes, Low-fidelity user prototypes, High -fidelity use prototypes and Live-data prototypes.

8. Joint Application Development

Joint Application Development is the process used to design computer-based system/solutions by collecting requirements side-side as per organisational needs while developing new information systems for a company through the approach of assessing the needs of the end-user to ensure the highest degree of satisfaction when using the product.

9. Feature driven developments.

Feature driven developments is an agile framework that is organised around the progression of features of the prototype through software development, it usually includes a team of around 50 or less people with and relies on chief developers which makes this methodology a top-down decision making approach following a 5 step development process which includes developing an overall model, drafting a features list, plan by feature, design by features and ultimately build the feature. Feature driven development is a methodology that is most used by large corporation running large software projects.

10. DSDM (Dynamic Systems Development Method): This is an Agile project delivery framework that focuses on delivering high-quality solutions on time and within budget. It emphasizes continuous user involvement, frequent iterations, and collaboration between stakeholders. DSDM encourages flexibility and adaptation to changing requirements throughout the development process.

11. Lean: Lean Software Development is inspired by lean manufacturing principles. It aims to maximise the value delivered to customers while minimizing waste. It focuses on eliminating unnecessary activities, streamlining processes, and creating efficient workflows. Lean advocates for delivering a Minimum Viable Product (MVP) and iteratively improving it based on customer feedback.

12. Rational Unified Process (RUP): RUP is an iterative software development framework that is process-centric and focuses on defining roles, activities and artifacts. It provides guidelines for tailoring the development process into four phases: Inception, Elaboration, Construction, and Transition, and it emphasizes the importance of documentation and continuous evaluation.

Task 2 Different Prototype approaches:

Throwaway Prototyping

Evolutionary Prototyping

Incremental Prototyping

Extreme Prototyping

**Throwaway Prototyping**

Throwaway prototyping is planning approach that companies implement to ensure that the final product is as close to the desired goal as it can be. This approach allows companies / individuals to view the requirements of the project through a physical or digital understanding. This is generally constructed in order to be a draft. The draft is created for feedback purposes and is discarded once the uses are complete.

**Evolutionary Prototyping**

The Evolutionary prototyping model is defined as stage-based process on creating the outcome of the product. This process uses both the incremental and extreme prototyping fundamentals to branch the respective models of each version into stages so that the prototype improves throughout the stage process. The basics of how this works is that a version of a prototype is created by the development team and evaluated by management then released to the client basis in order to receive feedback of which that is used to “go back to the drawing-board” and add or dissolve any functionality that is required to be.

**Incremental Prototyping**

Like Evolutionary Prototyping, Incremental prototyping is a staged based approach. Incremental differs from the previously mentioned prototype method though in the aspect that instead of multiple versions of a single prototype, the product is segmented into a collection of different prototypes then evaluated accordingly. This allows the development team to accumulate feedback on a wider range of functionality in different sectors of the product’s planning/testing stage then in the end, when multiple candidate prototypes are present, the team merges the most suitable traits to create the final product.

**Extreme Prototyping**

The last Prototyping approach method is the Extreme prototyping approach. The approach is divided into three subphases: the basic phase where the software is showcased in HTML format, the data layer where the prototype is simulated in a prototype service environment and the final phase is one where the functionality and services are integrated into the actual prototype. This is generally used in the web development domain, an approach where the User Interface is the focal point at first before converging this with functionality.

Task 3: THE THREE LEVELS OF SOFTWARE **PROTOTYPE**

LOW FIDELITY PROTOTYPE

Low fidelity is the most basic type of software prototype available. This will only incorporate the key elements of the product, such as very simple visuals such as shapes, and important content like titles. There is no functionality to these prototypes, they are a series of static screens that may be scrolled through to mimic a user’s journey and actions. In fact, a low fidelity prototype might not even involve software. It could be a paper prototype.

## **MID-FIDELITY PROTOTYPE**

A mid-fidelity prototype represents the middle ground between a low and high-fidelity prototype. This means that it offers some of the key content and functionality that will be used in the product, but it will not contain all. It might mean photos and graphics are missing, or some of the high-level interactions such as animation are not present. It will most likely be a series of wireframes that are linked together to give some levels of interactivity.

## **HIGH-FIDELITY PROTOTYPE**

A high-fidelity (hi-fi) prototype is created to work as closely to the finished product as possible. It has a high level of detail, including all the visual aspects that will appear on a site, all links, and most content. In many ways, this type of prototype could be mistaken for the final product.

Task 4 PROTOTYPING SOFTWARE EXAMPLES