

Ex.No. 1

Title – PROCESS MODEL SELECTION FOR “MEDICINE DELIVERY APP”

NAME -JAY HANSRAJ KHANIA

REGISTRATION NUMBER- 21BCE1394

[THIS ASSIGNMENT IS DONE WITH MY PROJECT PARTNER CHIRAYU BATRA 21BCE5756]

Date: 11-05-2023

Aim :

Analysis and Identification of suitable process model for “ MEDICINE DELIVERY APP”.

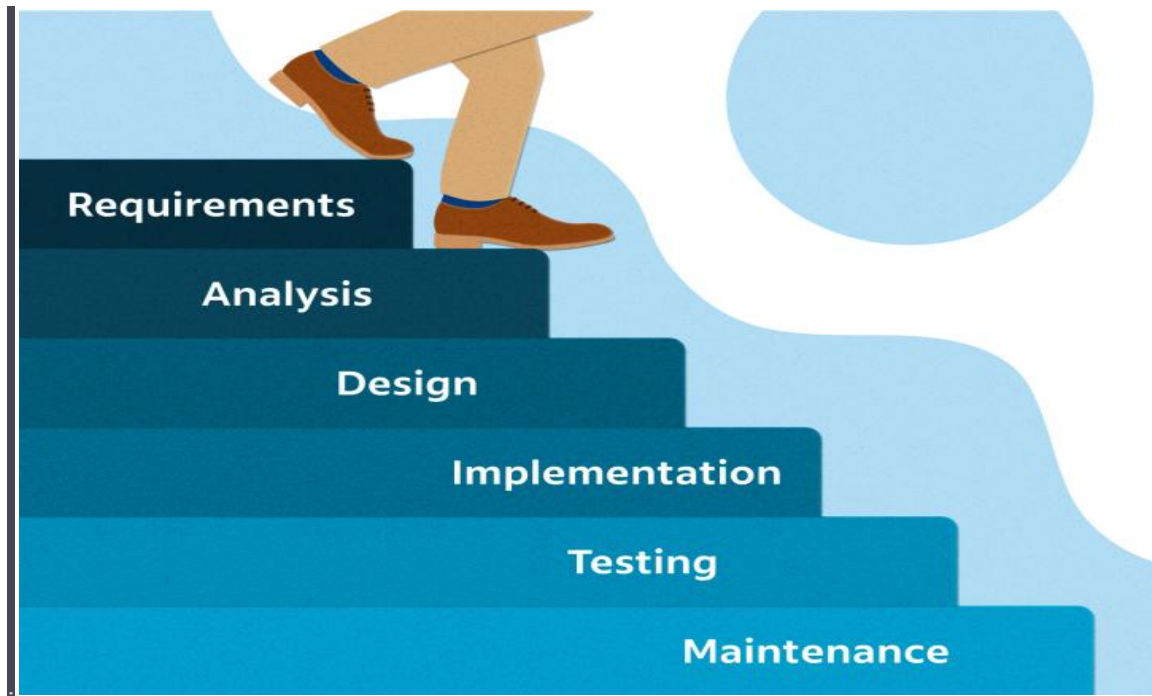
Description:

Modeling processes for a medicine delivery app involves designing a seamless and efficient flow of activities. The user begins by selecting medicines and providing delivery details. If a prescription is required, its authenticity is verified. The app checks inventory availability and offers alternatives if needed. Payment processing is securely handled, and the order is assigned to a nearby pharmacy or delivery partner. The pharmacy ensures accurate packaging, while the partner collects the order for delivery. Real-time tracking and notifications keep the user informed throughout the process. Upon delivery, the user confirms receipt and can report any issues. Feedback and ratings contribute to app improvement and customer satisfaction. The focus is on optimizing the delivery stages, ensuring accurate medication provision, reliability, and timely service.

Process Models:

1. WATERFALL MODELS

The waterfall model was the initial Software Development Life Cycle (SDLC) model widely adopted in Software Engineering to ensure project success. It involves dividing the software development process into distinct phases. In this approach, each phase's output serves as the input for the subsequent phase in a sequential manner. The diagram below provides a visual representation of the various stages in the waterfall model.



The sequential phases in Waterfall model are –

Requirement Gathering and analysis – All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification document.

System Design – During this phase, the requirement specifications gathered in the initial phase are carefully examined, and based on them, the system design is created. The system design plays a crucial role in identifying the necessary hardware and system requirements, as well as establishing the overall structure and framework of the system.

Integration and Testing – Once each unit is developed during the implementation phase, they are combined and integrated into a cohesive system. Prior to integration, thorough testing of each unit is conducted. Following integration, the entire system undergoes comprehensive testing to identify any faults or failures.

Implementation – With inputs from the system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality, which is referred to as Unit Testing.

Deployment of system – Once the functional and non-functional testing is done; the product is deployed in the customer environment or released into the market.

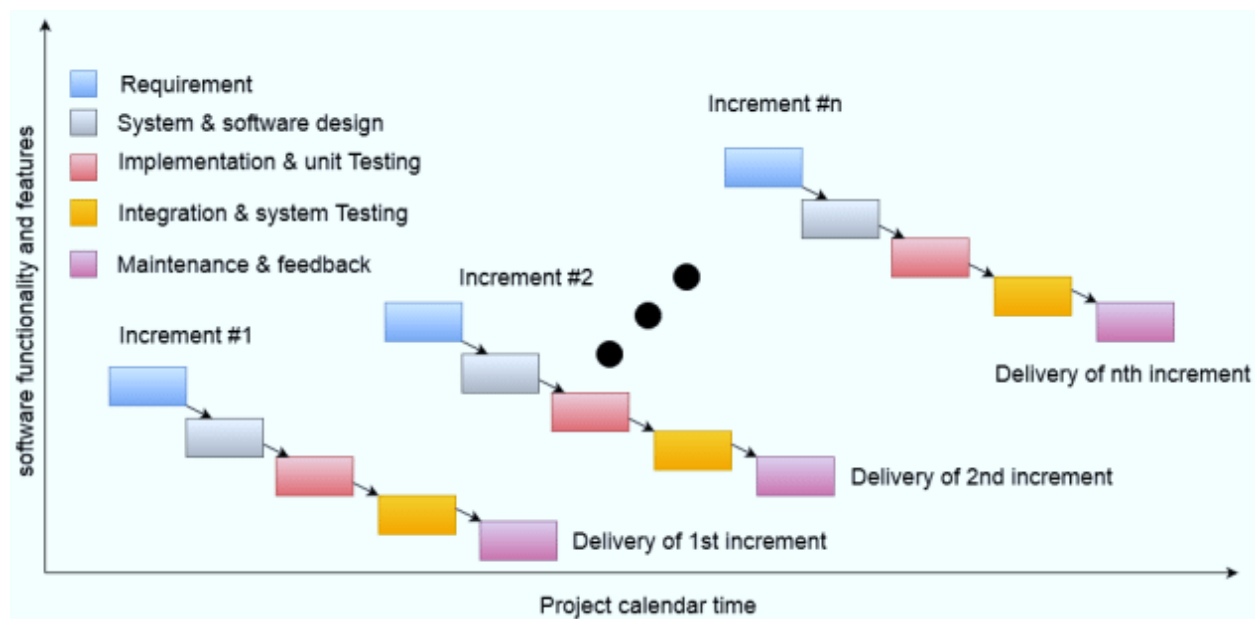
Maintenance – Certain issues may arise in the client's environment, prompting the release of patches to address and resolve those issues. Additionally, in order to improve the product, superior versions are introduced. Maintenance activities are carried out to implement these changes in the customer's environment.

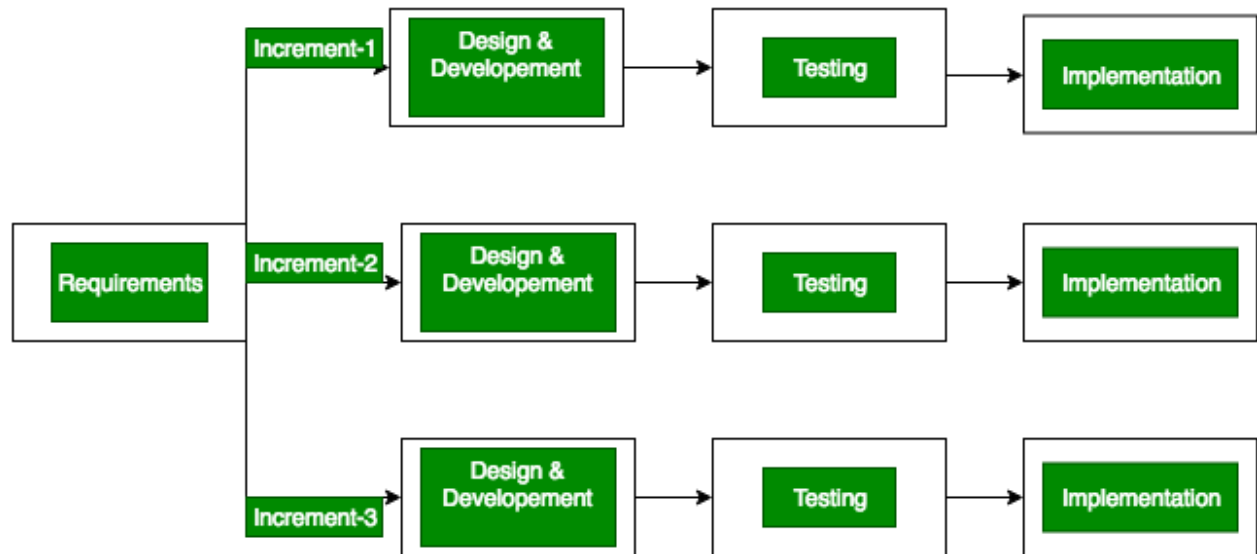
2. Incremental MODEL

The incremental model breaks down the development process into smaller modules or increments. Each increment represents a portion of the final software functionality, and they are developed and delivered sequentially. This approach enables early feedback and helps mitigate risks.

On the other hand, the iterative process begins with a basic implementation of a subset of the software requirements and gradually improves the evolving versions until the complete system is implemented. With each iteration, design modifications are made and new functional capabilities are incorporated. The underlying concept of this approach is to develop a system through repeated cycles (iterative) and in smaller portions at a time (incremental).

The following illustration is a representation of the Iterative and Incremental model –





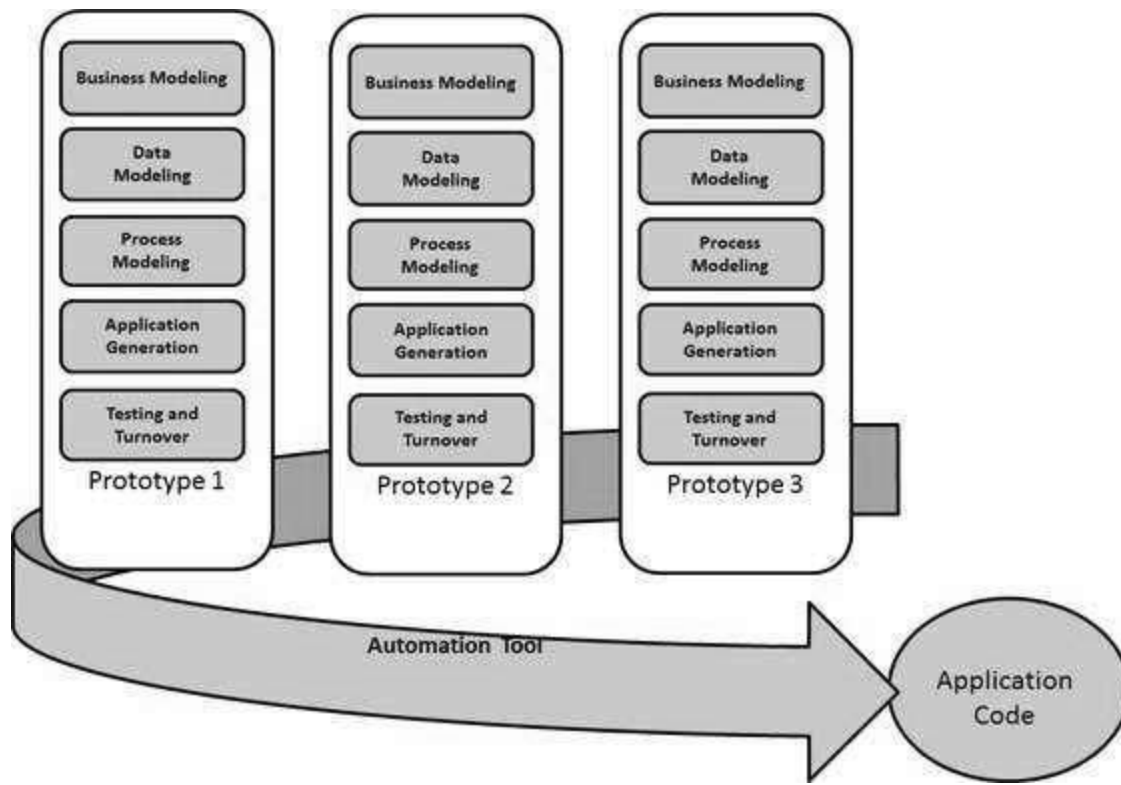
Iterative and Incremental development combines iterative design or method with the incremental build model for software development. It involves multiple iterations of the development cycle simultaneously. This approach can be characterized as an "evolutionary acquisition" or an "incremental build" approach.

In the incremental model, the overall requirement is divided into separate builds. Each iteration involves going through the requirements, design, implementation, and testing phases for the development module. With each subsequent release, additional functionality is added to the previous release. This process continues until the complete system is prepared according to the specified requirements.

3. RAD MODEL

The RAD (Rapid Application Development) model relies on prototyping and iterative development, without the need for extensive upfront planning. The software writing process incorporates the necessary planning for product development.

Rapid Application Development emphasizes the gathering of customer requirements through workshops or focus groups. It involves early testing of prototypes by customers using an iterative approach, reusing existing prototypes or components, continuous integration, and swift delivery.



Following are the various phases of the RAD Model –

Business Modelling: The product's business model, concerning the information flow and distribution across various business channels, is developed. A comprehensive business analysis is conducted to identify crucial business information, its acquisition methods, processing procedures, and the key factors that drive successful information flow.

Data Modelling: The information collected during the Business Modelling phase is examined and assessed to define essential sets of data objects for the business. The attributes of these data sets are identified and clearly defined. The relationships between these data objects are established and thoroughly described in relation to the business model.

Process Modelling: The data object sets defined in the Data Modelling phase are transformed to establish the necessary flow of business information required to achieve specific business objectives, in accordance with the business model. This phase defines the process model for any changes or enhancements to the data object sets. It provides detailed process descriptions for adding, deleting, retrieving, or modifying a data object.

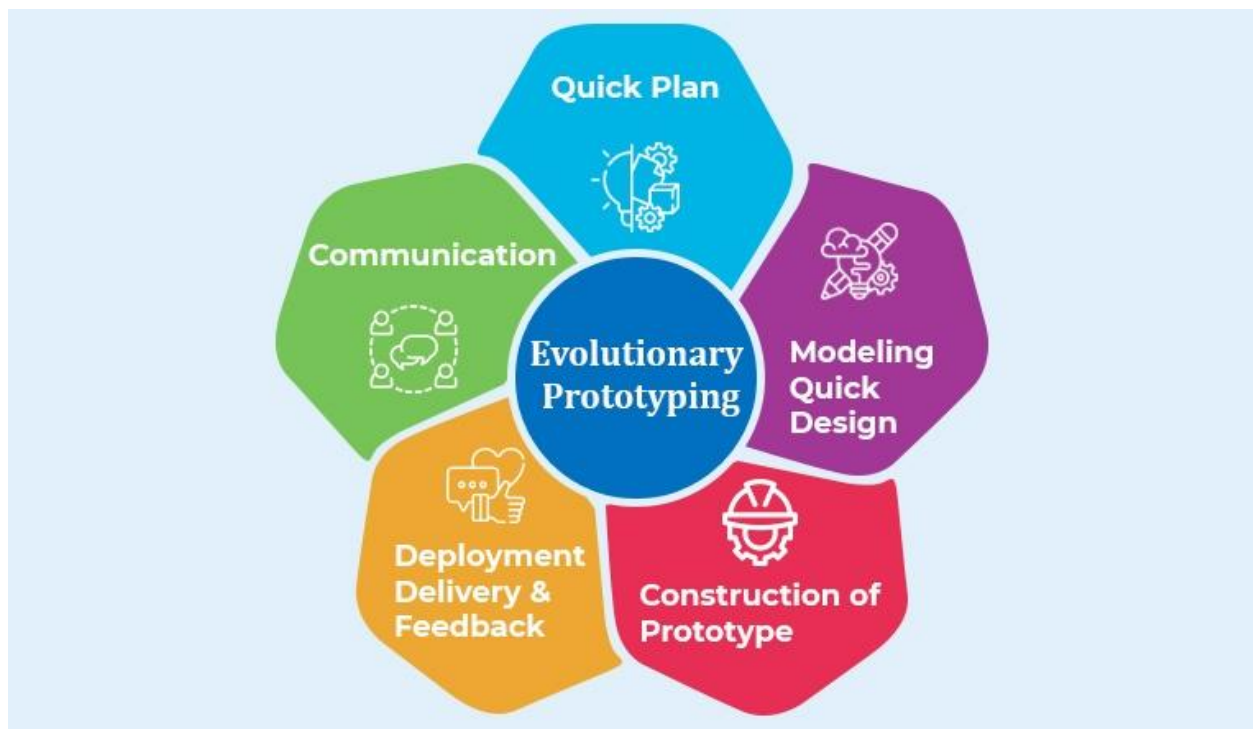
Application Generation: The actual system is constructed, and coding is performed using automation tools that convert the process and data models into tangible prototypes.

Testing and Turnover: The RAD model reduces overall testing time as prototypes are independently tested during each iteration. However, thorough testing of data flow and interfaces

between all components is necessary, ensuring complete test coverage. Since most programming components have already undergone testing, the risk of major issues is minimized.

4. EVOLUTIONARY - PROTOTYPE MODEL

A prototype model is a functional model of software that has limited capabilities. It may not precisely reflect the logic used in the final application and requires additional effort for estimation. Prototyping enables users to assess developer suggestions and test them before implementation. It also aids in identifying user-specific requirements that may have been overlooked during the initial product design.



5. SPIRAL MODEL

A prototype is a functional software model that has limited capabilities. It may not always incorporate the precise logic utilized in the final application, and its creation requires additional effort to be factored into the estimation process. Prototyping serves the purpose of enabling users to assess developer suggestions and test them prior to implementation. Additionally, it facilitates the understanding of user-specific requirements that may have been overlooked by the developer during the initial product design stage.

The spiral model consists of four phases, and a software project goes through these phases iteratively, known as Spirals.

Identification: The beginning stage of this phase involves collecting the business requirements during the initial spiral. As the product evolves in subsequent spirals, the identification of system requirements, subsystem requirements, and unit requirements are all carried out within this phase.

Continuous communication between the customer and the system analyst is an integral part of this phase to comprehend the system requirements thoroughly. Upon completion of the spiral, the product is deployed in the designated market.

Design: The Design phase commences with the conceptual design during the baseline spiral and encompasses the architectural design, logical design of modules, physical product design, and the final design in subsequent spirals.

Construct or Build: The Construct phase involves the creation of the actual software product in each spiral. In the baseline spiral, when the product is conceptualized and the design is being developed, a Proof of Concept (POC) is created in this phase to gather customer feedback.

As the subsequent spirals progress and there is a clearer understanding of requirements and design details, a working model of the software, referred to as a build with a version number, is produced. These builds are then shared with the customer for feedback.

Evaluation and Risk Analysis: Risk Analysis involves the identification, estimation, and monitoring of technical feasibility and management risks, including schedule delays and cost overruns. Once the build is tested, the customer evaluates the software and provides feedback at the conclusion of the first iteration.

The diagram below illustrates the Spiral Model, outlining the activities involved in each phase.

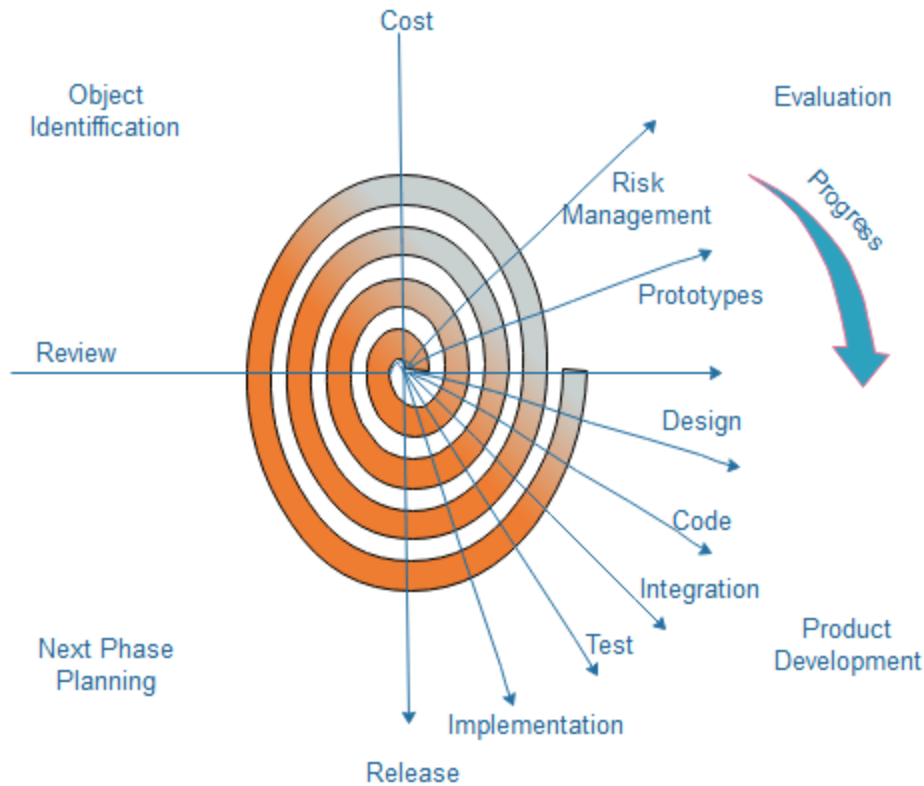
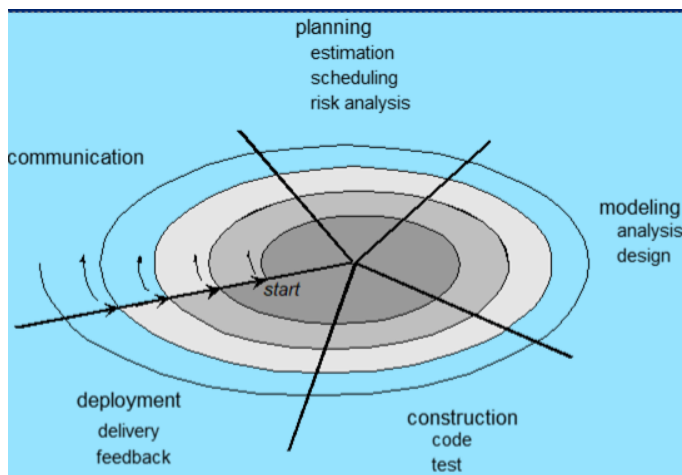


Fig. Spiral Model

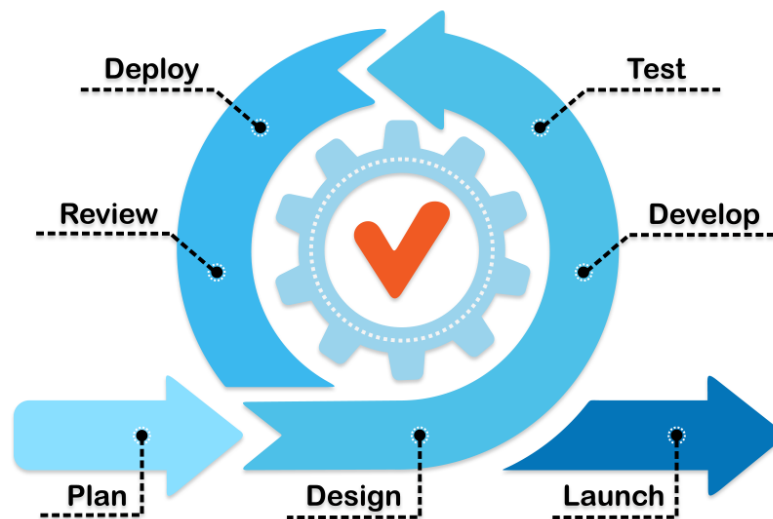


6. AGILE MODEL

The Agile SDLC model combines iterative and incremental approaches to prioritize process adaptability and customer satisfaction through the rapid delivery of functional software products. Agile methods involve breaking down the product into smaller incremental builds, which are provided in iterations. Each iteration, usually lasting from one to three weeks, involves

cross-functional teams simultaneously working on different aspects, including planning, requirements analysis, design, coding, unit testing, and acceptance testing.

At the end of the iteration, a working product is displayed to the customer and important stakeholders.



Following are the Agile Manifesto principles-

- Encouraging self-organization and motivation within the development team, along with promoting interactions such as co-location and pair programming.
- Prioritizing the demonstration of working software as the primary means of communicating with customers to understand their requirements, rather than solely relying on documentation.
- Recognizing the importance of continuous customer collaboration, as complete gathering of requirements at the project's outset may not be feasible due to various factors.
- Adopting a responsive approach to change, with a focus on quick adaptations and ongoing development.

OUTPUT:

MY PROJECT- Medicine Delivery App

The Medicine Delivery App is an innovative solution designed to revolutionize the way medications are obtained and delivered to patients. With the ever-growing demand for efficient and convenient healthcare services, this app aims to bridge the gap between patients and

pharmacies by providing a seamless and user-friendly platform for medicine ordering and delivery.

Key Features:

Medication Ordering: The app provides a comprehensive catalog of medications, allowing users to search and select the prescribed medicines they need. It offers detailed information about each medication, including dosage instructions and potential side effects, ensuring informed decision-making.

Prescription Upload: Users can conveniently upload their prescriptions within the app, eliminating the need for physical visits to pharmacies. This feature enables seamless verification and ensures the accurate dispensing of medications.

Personalized Recommendations: The Medicine Delivery App utilizes advanced algorithms and user data to provide personalized medication recommendations. By analyzing previous order history and considering individual health conditions, the app suggests suitable alternatives and reminders for medication refills, promoting adherence and patient safety.

Secure User Profiles: The app allows users to create secure profiles, ensuring confidentiality and privacy of their medical information. Users can store their previous medication history, dosage schedules, and allergy information for easy reference and seamless reordering.

Doorstep Delivery: One of the key advantages of the app is its efficient and reliable delivery service. Once an order is placed, users can track their package in real-time and receive updates on the estimated delivery time. This feature is especially beneficial for individuals with mobility issues or limited access to pharmacies.

Notification and Reminders: The app sends timely reminders for medication intake, ensuring patients adhere to their prescribed regimen. Additionally, it provides notifications regarding order status, delivery updates, and relevant healthcare information, keeping users informed throughout the process.

The Medicine Delivery App aims to enhance accessibility and convenience in the healthcare sector by simplifying the process of ordering and delivering medications. With its user-friendly interface, personalized recommendations, secure profiles, and reliable delivery service, the app empowers users to conveniently access their prescribed medications from the comfort of their homes.

SUITABLE PROCESS MEDICINE DELIVERY APP:

For Our Project we are using Agile Model. Agile development provides several benefits that align well with the requirements of the project:

Flexibility: The Agile approach allows for flexibility and adaptability, which is crucial in a dynamic project like a medicine delivery app. It enables you to respond quickly to changes, refine requirements, and incorporate user feedback throughout the development process.

Continuous customer collaboration: Agile methodologies emphasize continuous customer collaboration, which is essential for gathering user requirements, understanding their preferences, and incorporating their feedback into the app. This ensures that the app meets the specific needs of the users effectively.

Iterative and incremental development: Agile methodologies support an iterative and incremental development process. You can start with a minimal viable product (MVP) that includes core functionalities, release it to users, gather feedback, and then iteratively enhance and expand the app based on user needs and priorities.

Quick response to change: Agile development is designed to respond quickly to changes. In the context of a medicine delivery app, there may be regulatory changes, new medication guidelines, or user preferences that need to be accommodated promptly. Agile allows for these changes to be incorporated in shorter development cycles.

By utilizing the Agile development methodology, we can effectively manage the complexities of the medicine delivery app project, ensure customer satisfaction, and continuously improve and adapt the app based on user feedback and evolving requirements.

JUSTIFICATION FOR USING AGILE MODEL:

Rapid and Continuous Delivery: The Agile model enables rapid and continuous delivery of working software iterations. This means you can start delivering valuable features and functionality to users early in the development process. In the context of a medicine delivery app, this allows for quicker deployment of essential functionalities, such as medication ordering and delivery tracking, ensuring a faster time-to-market.

Adaptability to Changing Requirements: The Agile model embraces change and provides a flexible framework to accommodate evolving requirements. In the healthcare industry, requirements can be subject to regulatory changes, emerging medication guidelines, and shifting customer preferences. Agile's iterative approach allows for frequent reassessment of requirements and the ability to incorporate changes smoothly, ensuring the app stays relevant and up-to-date.

Customer Collaboration and Feedback: Agile methodologies emphasize continuous customer collaboration throughout the development process. In the case of a medicine delivery app, this

customer involvement is crucial for understanding their specific needs, preferences, and feedback. By engaging users throughout the development cycle, you can gather valuable insights, incorporate user feedback, and deliver an app that aligns closely with their requirements and expectations.

Focus on Quality and Testing: Agile methodologies encourage continuous testing and quality assurance practices. In the healthcare domain, where accuracy and reliability are paramount, rigorous testing is essential. With Agile, you can integrate testing into each iteration, ensuring that the app meets quality standards, user safety, and data security requirements.

Team Collaboration and Self-Organization: Agile fosters collaboration, self-organization, and cross-functional teamwork. This promotes effective communication among team members, ensures transparency, and allows for knowledge sharing and collective problem-solving. In the context of a medicine delivery app, this collaboration is vital for integrating various functionalities, such as user data storage, medication recommendations, and delivery logistics.

The Agile model's adaptability, customer focus, continuous delivery, and emphasis on collaboration make it a suitable choice for the medicine delivery app project. It enables you to address evolving requirements, incorporate user feedback, and deliver a high-quality, user-centric application efficiently.

Result:

Thus the analysis of various process models and identification of the suitable process model for the project “MEDICINE DELIVERY APP” is successfully done. The identified process model is “an Agile model”