Software Engineering:

Case Studies

Team Members:

Anish Sahu– 56

Aarav Bhandari– 47

Neeraj Kumar– 57

Anup Pai– 45

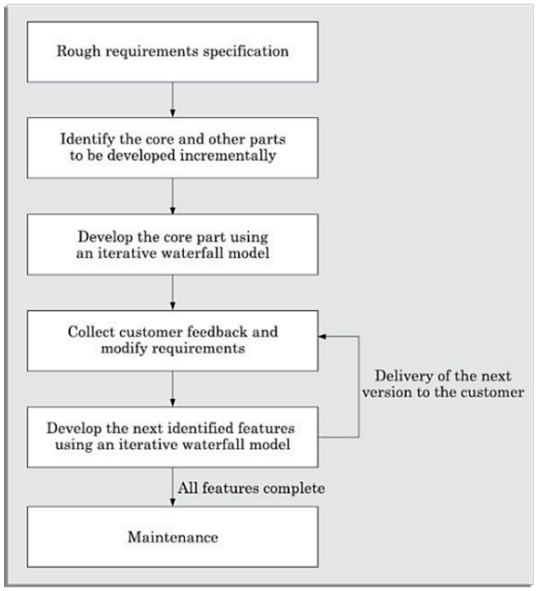
Case Studies:

Case 1:

Title: Time Management Software for a company

Model Name: Evolutionary Model

Diagram:



Explanation: The best approach to use in this case is the Evolutionary Model for TMS development. This model provides for continuous improvement, enhancement and meeting the changes of the business. The Evolutionary Model allows interim realization of the product, which means that continuous updates and suggestions from the executives are encouraged until the completed product is delivered, which would fully address their issues whenever it is needed.

Also, it can contain optional features, which would be non-invasive in nature such as, emailing or performing statistical analysis or even multi users using it but inside the friendly confines of new users.

Why it's better than the other models?

Waterfall Model:

- One major drawback is that this approach is not ideal when beginning any project since there is a likelihood that requirement may change during the course of the project.
- No intermediate feedback which leads to implementation of all changes at the last phase, which is quite expensive.
- Not suitable for inexperienced end users, because a triad of a user interface and a Process Management System have to be developed and hosted.

Iterative Waterfall Model

- Still maintains system development processes that are dependent on a linear conclusion since there exists most models that provide for gradual improvement which may be very useful.
- All iterations are client boundary oriented, constructions works for each phase which are prescriptive in practice leave out other essential alterations or constructions that may be required.

V-Model

- Very difficult model in addressing the changes is the v model which is inflexible.
- These concurrent phases do not make any modifications subsequently which constrains the freedom of testing.

Incremental Model

- There might be difficulties in merging increments to complete the system model due to integration among other factors.
- Good planning is needed to subdivide the increment effectively into the system.

Prototyping Model

- Ÿ Those working with prototypes may risk giving too much attention to the prototype rather than the entire system.

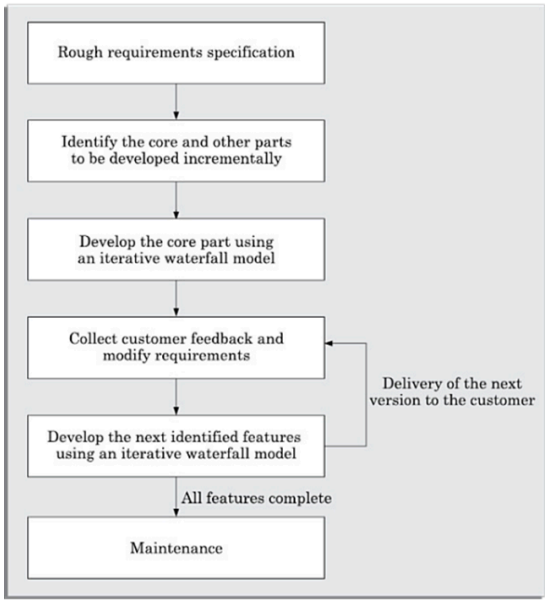
- Ÿ If these variations are not controlled, there is a very high possibility that scope – creep shall be experienced.

Case 3:

Title: Road Repair and Tracking Software (RRTS)

Model Name: Evolutionary Model

Diagram:



Justification: The Evolutionary Model is the most appropriate choice when designing the Road Repair and Tracking Software (RRTS) due to the following factors:

- Reconception: The system should be responsive when there is the need to adjust the repair emphasis, when changing the level of resource contributions, and when taking users' views to account. The Evolutionary Model incorporates incremental and iterative processes so that the system may be able to cope with changes.

- Incremental Changes Provision: The model offers possibilities to perfect the system during and after its deployment by collecting input from supervisors, city administrators and others including the public. This responsive development ensures that the final product supports the various processes within the Public Works Department.

- Timely reactions: One of the challenges in the Evolutionary Model is its capacity to deal with frequent change in product features and requirements. This consideration

is useful in order to be effective in road repair tracking since all interventions are subject to constant change.

All in all, the Evolutionary Model is an appropriate model to adopt as it simplifies the issue of balancing structure, adaptability and the need for iterative enhancement in the creation of DSTMS (a system to improve the reportage and tracking of road repairs).

Why it's better than the other models?

Waterfall Model

- Changes are impossible once the project moves beyond a step no matter the reason.
- If problems or changes in requirements are found at a late stage, confirming testing at advanced project stages may result in expensive resolution.

Iterative Model

- We can still appreciate that it has a linear structure despite all the time allowances which can affect its response to changes across phases.
- Less appropriate for projects which demand ongoing completion and correction of the components of the software.

V-Model

- Frame is very well defined and allows little to no stability changes.
- Testing is included within the development process and changes are hard to include, so newer requirements.

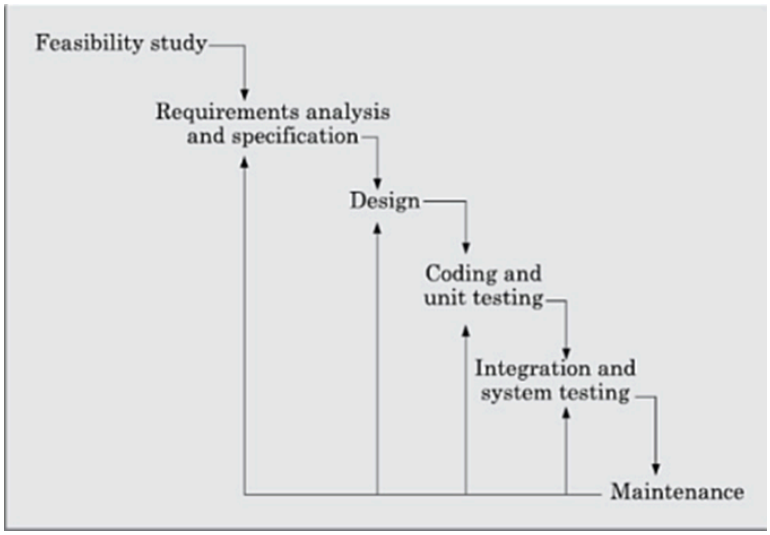
Incremental Model

- Problems of multidimensional integration may occur when we decide to synthesize the trust built-in.
- Every increment interdependent with the other incrementation necessitates thoughtful time management to prevent time from running out.

Prototyping Model

- Project risk of concentrating on prototype instead of final system.
- Changes control problem has led to large scope creep and challenged many project managers

Case 4:
Title: Judiciary Information System (JIS) software:
Model Name: Iterative Waterfall Model
Diagram:



Justification:

The JIS needs to be implemented in a systematic way though there should be room for accommodating certain shifts. The Iterative Waterfall model provides the advantages of normal development and the evolutionary model:

• Clear Phases: The structure assists the process of development in an orderly way with distinct phases of requirements analysis, design, development, testing and deployment.

• Iterative Development: The repetitive mechanism allows the project to be further subdivided in phases. This is important in a complex system like JIS where the requirements tend to change or be added through the course of the development process.

• Risk Mitigation: Focusing the development on one functionality of features in each iteration makes it possible for risks to be raised early in development and mitigated.

• User Feedback: It is critical that user needs are sought and addressed after every iteration, to develop a system that meets the expectations of judges, lawyers and registrars inclusive.

• Manageable Scope: Project scope can be easily controlled by implementing virtually all areas of the project in small sized increments.

Why its better than the other models?

Waterfall Model

• The shortcomings of this model are that it follows a defined structure; it does not respond to changes, which is one of the main features of JIS.

V-Model

• This is well suited in the case of well-defined requirements pertaining to the projects but as stated above JIS entails dynamic legal processes which can result in change even to the legislation. The rigor of the V Model of steady structure may not be quite satisfactory.

Incremental Model

• Although it encourages the delivery of a portion of the functionalities of the system at the very beginning, it may provide inconsistent project wise global project initiative and even cause disintegration for risks.

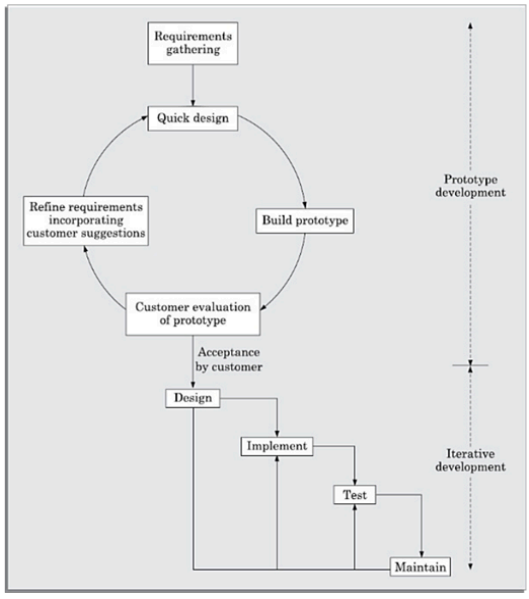
Prototyping Model

• Although it is good for identifying user needs, it may not be appropriate for a system like JIS on a large scale because it has to be built.

Evolutionary Model

• This model is often the case in projects which are more of an exploratory nature and have not clearly defined requirements. But in JIS there are discernible primary functions.

Case 5:
Title: Work processing software
Model Name: Prototyping
Diagram:



Justification: The Prototyping Model is the best option for designing the word processing software for the following reasons:

- Responding to Users' Concerns. The Prototyping Model supports formation of a prototype which a user may use at the early stages of development. This is particularly important in perfecting features such as text formatting, justification and interactive editing that are critical in determining user satisfaction.

- Understanding and Refining Requirements. When a user is using the software prototype, there are many features that they will interact with and hence help specify more requirements for many functionalities that may be offered via the software. This is a rapid feedback loop that is important in the evolution of an efficient word processing application.

- Consideration of Usability Constraints: The prototype also highlights usability constraints if any so that corrective measures can be taken with respect to the final design. This is most desirable more so in the case of word processing software which includes features that have to be user friendly.

In summary, the Prototyping Model allows a proper balance of flexibility of development processes, user participation, and the sequence of events enabling a good word processing software to be developed that is usable and satisfying to its users.

Why its better than the other models?

Waterfall Model

• It is however said that this model is quite rigid and changes cannot be made once any particular phase is complete.

• Testing takes place towards the end of the process at great cost if issues or even changes are identified then.

Iterative Waterfall Model

• Though changes are somewhat likely to happen, most of this model still wades in the direction of the sequential process grid.

• Each of the fore-mentioned and referred to iterations is very durable and quite expensive if it comes to the need to elevate performed changes.

V-Model

• It is very strict and since the idea is forged it is very difficult to entertain the chance for any change in the development process.

• Testing is done alongside the development phase but there is always room for disruption due to traded expectations at any stage in the process.

Incremental Model:

• The placing of increments together may take longer than anticipated and may involve a plethora of steps.

• This also entails the management of increments in terms of the interdependencies that exist between them in a bid to achieve their incremental goals.

Evolutionary Model

• Managing the risks and scale of the project can be a challenge to keep the scope and direction of the project within limits.

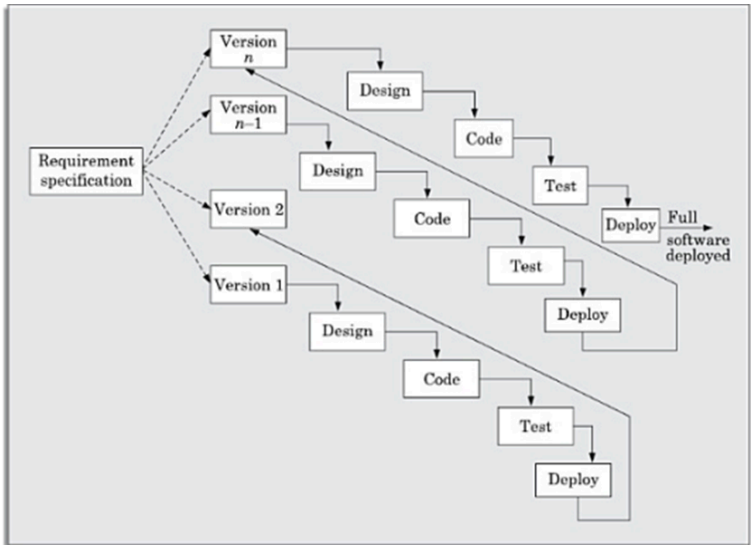
• To maintain the project, it may be less complicated to schedule the work although it is important to update changes regularly into backlogs.

Case 6:

Title: Restaurant Automation Model

Model Name: Incremental Model

Diagram:



Justification: The Incremental Model is the best option for designing the Restaurant Automation System for the following reasons:

• Developmental Disintegration into Stages: The Incremental Model permits the system to be built in stages, where each increment implements, partially or completely, a functional portion of the system. This is very suitable for RAS whereby the order processing functionality, then billing, and the inventory management and so on can be branched and each of them completes its cycle of development, testing and improvement.

• Core Features can be Delivered Early: Core features can be given higher parts and it can be developed fast enough for the restaurant to start using it and give feedback on it. This makes sure that the core features are working properly and are attractive to users lord of activities that are critical are completed before adding more activities that are less critical.

• Adaptability: The Incremental Model gives room for any adjustments and improvements with regard to user feedback after every segment has been released. This is useful for a system like RAS where the users and the business processes may change with time as they the development continues.

By and large, the adoption of the Incremental Model is advantageous, as it eliminates the compromise between flexibility, user participation where it is necessary and development of the system in stages in a reasonable way.

Why it is better than other models?

Waterfall Model:

• This model has a rigid structure; any change can become difficult after certain phase is completed.

• If changes or issues are identified late in the project, they may incur costs while being solved during the implementation of the system.

Iterative Waterfall Model:

• It is somewhat sequential, although emphasized tasks once accomplished do not include going back.

• Cost and time to make such changes can be high.

V- Model:

• Stick oriented and rigid. The Waterfall Model contains branches, everything is carried out on straight lines.

• It is hard to abate changes when the development stage is already being undertaken.

Prototype Model:

• Any overemphasis on the prototype runs the danger of providing the final deliverable way too late in the project lifecycle.

• Change control can be hard as scope control practices are not followed.

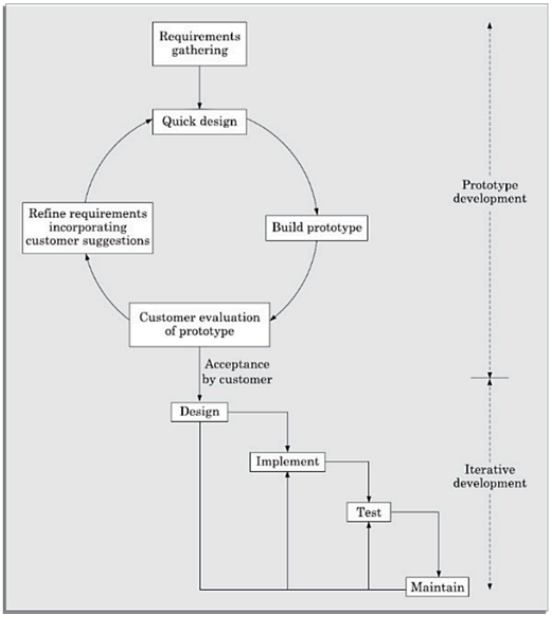
Evolutionary Model:

• Some level of project management is required so that one does not digress from the goal intended by the plan.

• Changes or modifications that are always made regularly on ones writing can also be very difficult to deal with.

Case 8:

Title: Factory Management System
Model Name: Prototyping
Diagram:



Justification: When it comes to the improvement of machine utilization and adjustment efficiency in the factory simulation software development, the Prototyping Model is the best approach for the following reasons:

- The Prototyping Model facilitates iterative development whereby a working prototype has been developed and tested and where refinements of the prototype take place several times before the actual software is finally produced. This is particularly important for simulation software, which should be done correctly with complex elements such as machine MTTF, adjuster availability, queuing management, and many other variables.

- Incorporating the User: principal testers' involvement is greatly appreciated due to the feedback on whether the model has been used positively and accurately. The feedback is implemented in later rounds which enables building a very strong system that is not complicated in scope and raw power.

- Reducing Risks: prototyping possesses this benefit where potential problems are raised much early thus limiting the risks that are attributed to fallacy or something that was lost in translation of the workings of the Factory.

It is quite evident that due to its provisional characteristics, orientation towards torrents opinions and approach of carrying on adjusting design until all needs of the factory are addressed, Prototyping Model is the best to create factory simulation software. This way, the guarantee of correctness and appropriateness of the simulation remains with all the needed and even more detailed aspects of machines and computing processes that will support and power the simulation.
Why its better than the other models?

Waterfall Model:

• This model tends to be very inflexible; once any stage of the phase is completed, modifications become very hard.

• Testing is done too late in the process and problems discovered late in the process could be very expensive to fix.

Iterative Waterfall Model:

• Still very sequential, to the extent that once a phase is completed, there is little or no possibility of going back.

• Modification of initial concept may require spending resources which may prove unwise and take time.

V-Model

• Very rigid and sequential like that of Waterfall model.

• It is not appropriate in projects which expect heavy feedback from users and constant iterations.

Incremental Model:

• It is more complicated in that upon adding increments, integration between these units can get difficult, whereas in this case few modules (i.e. machine failure modeling, adjuster management) would require to work together.

• Simulation may not be possible to test fully, and holistic approach concerning the simulation may not be executed until all increments are done.

Evolutionary Model

• In the evolved model, such changes might be very hard to undertake as there are always changes being made during the life of the program.

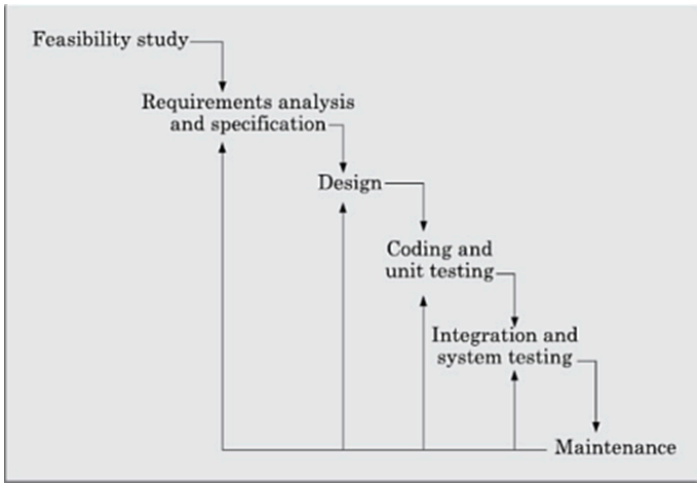
• The growth of the ambit of the project in terms of costs should be controlled.

Case 10:

Title: Supermarket automation software (SAS)

Model: Iterative Waterfall Model

Diagram:



Justification: The Iterative Model is suitable for development of supermarket automation software (SAS) due to its ease of adjusting to the ongoing changes in need as well as accepting some level of feedback. The system is able to deliver functional parts in working cycle, making it possible to test early and often, improvise, and factor in real world concepts. This expensive method aims to minimize risk by making major development decisions when it is least costly to do so by making improvements to the system design as requirements of the stakeholders change and incorporate them to the present design.

Why it is better than other models?

Waterfall Model:

Ÿ This model is not utilized due to Bureaucratic in Fulfilling Requirements since it is an excessively linear and sequential model that is unsuitable for some projects whose specifications are subject to change.

Ÿ The Iterative Model has the strong advantage of achieving improvement and enhancement through continued practice which is important for an application like SAS which must function effectively in real life.

Incremental Model:

Ÿ This model is not used as this supports the development m, instead, a nonrelease focus on iteration may be pursued.

Ÿ The consideration of the factors within the system is essential in the Iterative Model which is centered around improvement and feedback.

Ÿ Prototyping

Ÿ This model is not used as it is not suitable for interface prototyping and proving concepts, but it does not reach issues of full development cycles as well as Iterative models.

Ÿ The Iterative Model focuses on the improvement of the system and the whole process is more staged towards achieving the development and improvement of the system.

V-Model

Ÿ This is not employed in courtesy of his order in the development and testing process.

Ÿ Despite its structure, it might lack that feedback loop and flexibility that allows for iteration in the refinement of such a complex system such as SAS.

Evolutionary Model

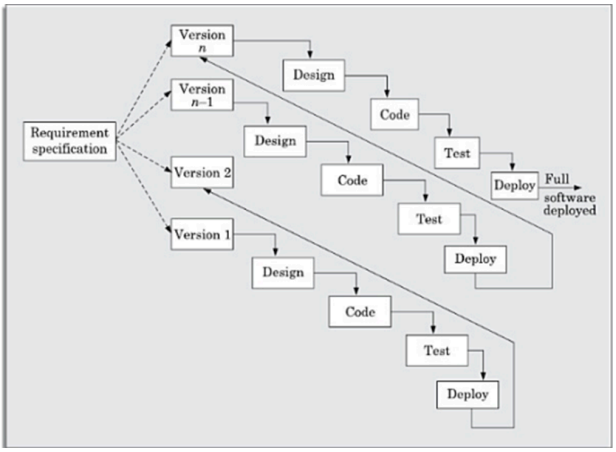
- ☐ This model of development may not be fitting for the SAS as it encourages prototype development in cycles at the expense of a functional system and may lead to scope creep and integration issues.

Case 13:

Title: CASE tool for Structured Design

Model Name: Incremental Model

Diagram:



Justification: Choosing the incremental model for developing your CASE tool is advantageous because it allows for modular development, early delivery of core functionalities, and ongoing user feedback. By breaking the project into manageable increments, you can focus on essential features first, progressively add complexity, and adapt to changing requirements with flexibility. This approach ensures early issue detection, efficient resource management, and better control over costs, leading to a more stable and user-aligned final product.

Why is it better than other models?

1. Waterfall Model

One of the models that is not put into practice is the Waterfall model due to its sequential and straight-jacketed nature, which makes it less preferable for iterative design alterations.

2. Iterative Model

Unlike in the Waterfall model these phases marks a stop in development and the techniques that can not be used are not necessarily acquired even when a model allows for repetitive improvement.

3. Prototyping

This model is not used because its focus is on building a systems working model within which alternative designs are considered and hence may not work well with the right discipline needed for detailed design and hierarchy of CASE tools.

4. Evolutionary Model

This model is not used because in this model of building design, there may not be so much of the civilized systematic design style which may frustrate the use of a CASE tool which expects orderly design management and interrelationships.

5. V model

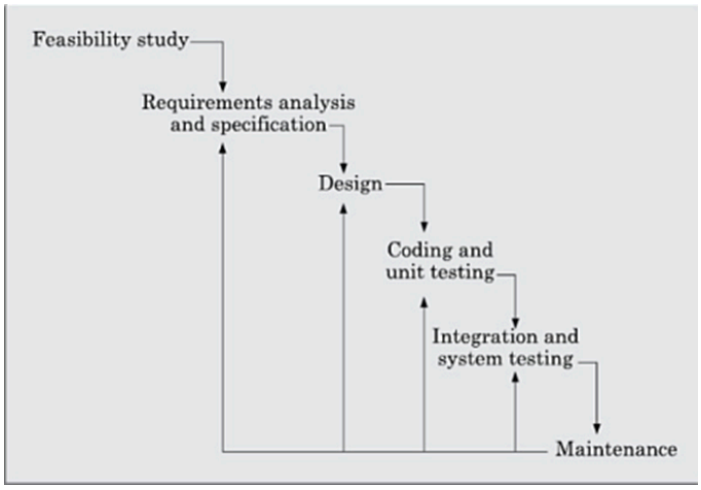
This model is not used as validation and verification approaches are conducted in all development milestones. The model components also provides for little or no flexibility on how often changes and modification of the design process can be done.

Case 14:

Title : Newspaper Agency Automation Software

Model Name : Incremental Model

Diagram :



Justification :

The Incremental Model is suitable for creating Newspaper Agency Automation Software because it addresses complex requirements and changing requirements effectively. In this model, involving subscription and delivery management, billing and payment processing are designed separately and implemented step wise, so that rather balanced incremental design and verification is possible. The periodic restraint of the system model also provides for fast delivery of useful incorporate features, feedback on the features to enhance enhancements and reduce the development risks associated with such improvements. This way the end system comes out to be sound and usable with respect to the expectations of the news agency.

Why is it better than other models?

1. Waterfall Model

This model is not suited because of changes in requirements like newspaper subscription. Customer address update as well subscription.

2. V-Model

This model is not suited as changes in requirements or feedback may not be effectively integrated due to the model's sequential nature.

3. Iterative Waterfall Model is not suited, as their constant changes in requirements of the agency like, changing subscription during special event.

4. Evolutionary Model is not suited as it focuses on iterative development with frequent releases and changes in requirements even in release of the product.

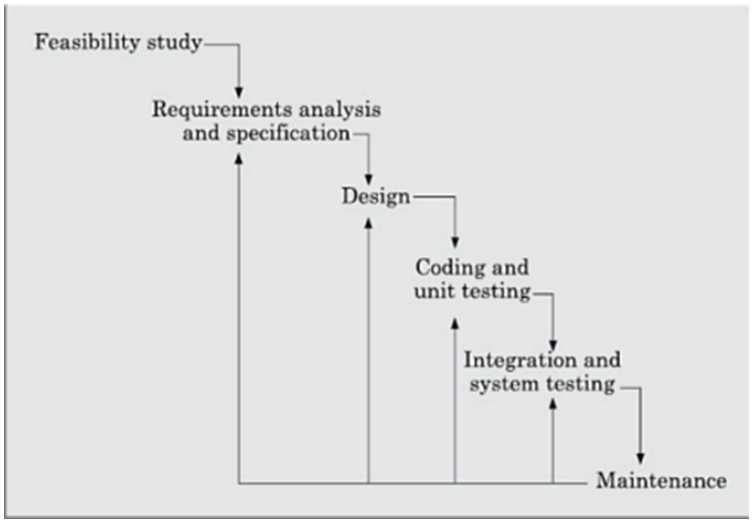
5. Prototyping is not used as the requirements are relatively well-defined.

Case 16:

Title: Medicine Shop Software (MSS):

Model Name: Iterative Waterfall Model

Diagram:



Justification: The ***Iterative Waterfall Model*** will be more suitable for MSS, with its complex evolving requirements. Through this approach, gradual development and refinement of functionalities related to inventory, sales tracking, and reporting can be accomplished. This incremental construction of the system will allow for frequent testing, incorporation of user feedback, and risk management by adapting to changing needs. It's a technique whereby each component of the system is guaranteed to be working and according to the needs of the shop owner before moving to the following phase. This approach, secondly, gives room to add new features or make any adjustment on demand from the user while development is going on.

Why it is better than other models?

1. Waterfall Model is not suited because of changes in requirements like threshold value, expiry date of medicine and so on.

2. V-Model is not suited as there is no feedback from the retailer, in improvement of the automation.

3. Incremental Model is not suited as it focuses on creating successive versions at each iteration.

4. Evolutionary Model is not suited as it focuses on iterative development with frequent releases and customer feedback.

5. Prototyping is not used as the requirements are relatively well-defined.

Functional Requirements:

CASE 1 : Time Management Software for a company

1. Appointment Registration

• Description: Top management will register their daily appointments. The appointments may include meeting person(s), venue, time, duration, and purpose.

• Input:

o Meeting person(s)

o Venue

o Meeting time and duration

o Purpose of the meeting, which may be for a project work

• Output:

o Indicate that the appointment has been successfully saved so that same can

be observed by the executive in the calendar.

o Stored the data of appointment made so that at the end of a week or day

2. Group Meeting Scheduling

- **Description**: Enable arranging meetings involving multiple executives, updating all participants' diaries with the meeting details.
- **Input**:
  - List of participating executives
  - Venue
  - Meeting time and duration
  - Purpose of the meeting
- **Output**:

3. Automatic Email Notification of Meetings

• Description : Send email notification to the concerned executives regarding a newly scheduled meeting.

• Input :

  o Details of the meeting - list of executives involved, time, date, venue and purpose.

• Output :

  o Automatic Email notifications to all the concerned executives.

4. Slot Rescheduling

• Description: If there is no common slot available to accommodate all participants, the secretary should be given the facility to reschedule appointments to arrive at a mutual slot.

• Input:

  o Meeting participants

  Available appointment slots for each executive

• Output:

  o Suggested rearranged schedule or free slot.

  o Notification to executives for confirmation.

5.Daily Schedule View

• Description: Provide easy-to-handle graphical interface displaying the daily schedule of executives for any given day.

• Input:

  o Date selected by executive

• Output

  o. Displaying the Executive's daily schedule- Meetings and Engagements.

6.Meeting Statistics by Executive

• Description: The statistics about how much time each executive has spent in meetings.

• Input:

  o Time period statistics are requested for

  o Details of executives

• Output:

  o Total meeting hours spent by the selected executive within the given period.

7. Project Meeting Statistics

• Description: To provide statistics about the number of meetings held for each project, the time spent on them, and man-hours spent concerning the same.

• Input:

  o Name of the project

  o Time period for which statistics are to be found

• Output

  o No. of meeting, time spent and man-hour spent on the selected project stated.

CASE 2 : Hotel Automation Software

1. Room Type and Tariff Management

• Description: Provides single/double bed with AC/Non-AC options. Have a tariff that may be seasonal, depending upon the month and the season of bookings.

• Input:

- o Kind of Room: Single/double, AC/Non-AC
- o Time in a Year: Month
- o Occupancy
- o Percentage revision of tariff: upwards/downs

• Output:

- o Computed revised tariff of room based on the entered conditions.
- o Current tariff displayed for every type of room.

2. Room Availability and Reservation

• DESCRIPTION: The facility allows its guests to book the rooms in advance or even at the moment based on the availability of the rooms.

• INPUT:

- o Guest information: name, time of arrival, estimated time of stay
- o Guest room type: single/double; AC/Non-AC
- o Type of reservation: advance/on-the-spot.

• OUTPUT:

- o Token number and the room number for the guest.
- o Message of apology, in case no rooms available.

3. Guest Data Entry by Receptionist

• DESCRIPTION: The receptionist enlists information on the guest, the time of arrival, advance paid, approximately how long the guest intends to stay, what type of

- o Guest details - name, arrival time, advance paid.
- o Room type required - single/double, AC/Non-AC.

- Approximate duration of stay

• Output:

- Room allotment confirmation along with room number and token number.
- Sorry message if no rooms are available.

4. Room Allotment

• Description: The system will automatically give a room depending on the guest requirement and the availability.

• Input:

- Room type required by the guest.
- Availability of room.

• Output:

- Room number, token number to be given to the guest.
- Apology message along with the reason if no suitable room is available.

5. Catering Service Data Entry

• Description: Hotel catering manager records food consumed by the guest along with the type of food item and quantity, date, and time of consumption.

• Input:

- Guest token number
- Type and amount of food items
- Date and time of consumption

• Output:

- Food items are recorded against the guest bill.
- Excerpts of the updated guest bill on account of food consumed.

6. Check-out and Bill Generation

• Remark: The final bill is generated to the guest while checkout, which includes room charge along with food consumption.

• Input:

- o Guest token number
- o No. of days stayed in the room
- oQuantity Food consumption details
- o Discount if any

• Output:

- o Final bill showing room charges, food charges, and total payable amount.
- o Balance amount payable (or refundable) by the guest.

7. Occupancy Rate Calculation and Display

• Description: To calculate and display the average occupancy rate for the selected month to enable the manager to revise tariffs.

• Input :

- o Month
- o No of rooms occupied during that month
- o Total available rooms

• Output:

- o Average occupancy rate for the month selected.

8. Room Availability Query

• DESCRIPTION: The staff shall be able to view room availability for a given type and date range.

• INPUT:

    o Room type - single/double, AC/Non-AC

    o Date range

• OUTPUT:

    o List of available rooms as per the selected criteria

    o No match message if no rooms are available matching the typed criteria.

Case 3:Road Repair and Tracking Software (RRTS)

1. User Management

Description: This includes administration of system users, which consists of user authentication, access management based on roles, and activities within users' profiles.

Input: User credentials (username, password), assignment of roles, allocation of profiles such as names and emails, contacts

Output: Messages of authentication either successfully completed or failed, availability of functionality relating to the specific user, verification of profile changes made.

2. Repair Request Management

Description: This gives the clerks an option of entering road repair requests and tracking the requests made by maintaining a record of all the requests.

Input: Road locality, kind of problem, name of requestor, and the status of request

Output: Request entry confirmation, request state updates-beg, processes, end results and logs.

3. Supervisor Management

Description: This allows repair requests to be approved by the supervisors who can allocate tasks to workers and determine what resources will be required to complete said tasks and create work plans.

Input: How inspection was carried-its intensity and the texture of area, the level of prioritization, the abilities in terms of resources-machinery, personnel.

Output: Priority levels that have been assigned, information about the resources that are needed, completion of schedules related to the repair works.

4. Resource Management

Description: Oversee and the actual numbers of manpower, machines and materials for undertaking various repair works.

Input: Resource constraints data (such as man power, machines, materials), variations of the constraints.

Output: Report on the availability of resources for use on the project, Internal reports on resource allocation, Revised operational repairs scheduled where there are changes in the resources.

5. Repair Scheduling and Tracking

Description: Deals with automation and tracking of schedules of repair work, bearing in mind the available necessities and the set priorities.

Input: Amount of work for a given repair, the available resources, editing of the repairs' schedules, reporting progress.

Output: Plans for repairs of structures developed, status of repairs conducted(commenced, on hold, completed) as at that point of time: notifications issued by the system to staff members.

6. Reporting

Description: Prepares and submits various reports within the scope of repair activities, overdue work and usage of resources.

Input: Inquiry concerning the type of the document, how to generate the report, its filters (time frame, location, magnitude).

Output: Reports on activities of repair works including statistics, reports on uncompleted repair works, reports on resources usage, Reports on demand.

7. Audit and Logging

Description: Log every change affected on the requests, schedules, resource data as well as the access details of the users.

Input: All actions carried out in the system including requests made by the users, changes in schedules, updates to resource information, user activity in accessing the system.

Output: Logs of the changes made, logs of the users who accessed the systems.

8. Integration and Communicative features

Description: Support the working of alerts for communication and interlinking with third party systems such as GIS for better performance.

Input: Event triggers (status change, resource allocation), GIS data (if applicable).

Output: Email/SMS alerts, mapped repair locations (If GIS is integrated).

9. Dashboard

Description: Key performance indicators such as the current repair status of assets, resources deployed, etc. will be easier and will all be visible on a single page.

Input: Status of repair requests, schedules of repair requests and resource management in real time.

Output: Screenshot of real-time dashboard showing number of pending requests, repairs in progress, resources being worked on and completed tasks.

10. Data Backup and Recovery

Description: Allow for thorough system backups to be carried out on a regular basis and allow for data recovery in the event of a system crash.

Input: System data, backup schedules.

Output: Backup files, confirmation for data recovery.

Case 4:Judiciary Information System

1. Case Management

1.1 Case Entry

Description: This enables the registrar to enter the new case information into the case management system.

Input: Defendant's name Defendant's address Crime type eg; theft, arson Date of the crime Location of the crime Arresting officer's name Arrest date

Output: A unique Case Identification Number (CIN) is generated. Case details are stored in the system.

1.2 Hearing Date Assignment

Description: This enables the registrar to assign a hearing date by selecting an appropriate date from a list of available dates.

Input: Case Identification Number (CIN) Requested hearing date or date range

Output: Display of available time slots on the requested date or time frame provided within Earlier hypothetically defined system practices. Assigned hearing date is recorded in the system.

1.3 Adjournment Handling

Description: This allows the registrar to enter details of reason for adjournment and enter some new hearing date.

Input: Case Identification Number (CIN) Reason for adjournment New hearing date (e.g. selected out of options)

Output: Details of capital adjournment are stored in the system. The system is informed of the new hearing date.

1.4 Entry of Court's Proceedings

Description: This enables the registrar to summarize the proceedings of court as heard after every hearing.

Input:

Case Identification Number (CIN)

Summary of the proceedings made in the court

Next hearing date, if any.

Output:

The summary is attached to the record of the case.

Next hearing date is also recorded.

1.5 Case Initiation

Description: Enables the registrar to close a case when inserting irrelevant judgment summary by clicking on completion.

Input:

Case Identification Number (CIN)

Summary of the cases made or given judgment

Date on which the case was lastly closed.

Output:

The status reflects ‘Closed’ on the case.

All related information is stored in the database, away from the interface screen.

2. Accessibilities and Security of Users

2.1 User Account Creation

Description: Provides the opportunity to create user accounts for judges, lawyers, other court personnel, and registrars.

Input:

User Details (Name, Role, and Contact)

Access level (e.g. judge, lawyer)

Output:

A new user account that has been created is in existence with relevant permission.

2.2 Deletion of User Accounts

Description: Permits the registrar to remove a user's account from the system.

Input:

User ID or username

Output:

The particular user account has been cleaned/deleted from the system.

2.3 Access Control

Description: Ensures different access levels for different users (for example judges have full access while lawyers have limited one).

Inputs:

User role (i.e. judge/lawyer)

Access permissions Outputs:

Users can only access functionalities and data as per their assigned role.

2.4 Usage Tracking

Description: Tracks the number of cases seen by each lawyer for billing purposes.

Inputs:

Lawyer's User ID

Case Identification Number (CIN) of case being viewed Outputs:

Log entry is created with a lawyer's ID, CIN and timestamp.

The number of cases seen is documented for billing.

3. Query and Reporting

3.1 Pending Cases Query

Description: Allows registrar to get a list of all pending court cases.

Input:

No specific input (or alternatively filters like date range)

Outputs:

A list of pending cases sorted by CIN which includes:

Start date

Defendant name and address

Crime details

Lawyer’s name

Public prosecutor's name

Attending judge’s name

3.2 Resolved Cases Query

Description: Allows registrar to retrieve a list of resolved cases within a specified period.

Input:

Date range Outputs:

Chronological list of solved cases including:

Start date

CIN

Date judgment was delivered

Name attending judge

Summary judgment

3.3 Upcoming Hearings Query

Description: This gives the registrar an opportunity to retrieve historic hearing dates information that are on schedule for a particular day.

Input:

Date in question

Output:

A list of cases scheduled for hearing on this particular date comprising:

Case Identification Number (CIN)

Defendant’s Name

Crime Details

Lawyer Name

Public Prosecutor Name

Attending Judge Name

3.4 Case Status Query

Description: The same allows the registrar to know about the status of one specific case.

Input

Case Identification Number which is known as CIN

Output

: Current status of the case including;

Progress of the case

Next hearing date if applicable

Any notes or updates entered by registrar

4. Resource Management

Description: Repair tasks regarding manpower, machines and materials can be made available or allocated by management that deals within them in order to accomplish vehicle repair jobs

Input

Data regarding resource availability (manpower, machines, materials), alterations to resource availability.

Output

: Up-to-date information concerning resource availability; resource allocation reports; adjusted repair jobs when resources undergo modifications.

5 Repair Scheduling And Tracking

Description: Automates and tracks the scheduling of repair work according to priority levels and existing resources

Input

Repair preference order; availability of resources; manual alteration of schedule; progress reports

Output

Generated schedules for repairs; current status updates on repairs (started paused finished) notifications sent to relevant users.

6. Report

Description: Different reports are generated in relation to repair statistics, tasks which are not yet done or ongoing and how resources are being used.

Input: Request for report type; filters (date range, area and severity).

Output: Repair statistics reports, outstanding repair reports, resource utilization reports and custom reports.

7. Audit and Logging

Description: Changes made to requests, schedules, and resource data should be recorded while maintaining user access logs.

Input: System actions (request changes, schedule changes, resource updates); user access activities as they occur.

Output: Change logs and access logs.

Case 6: Restaurant Automation System

processOrder

Description: The system generates a bill and also performs the update in the inventory when it processes a customer's order by taking note of the item code and the quantity sold.
input: item code, quantity sold are provided.
Output: invoice, currentStock

generateBill

Description: Using the order ID, a bill for a specific order is generated.
input: Order_id is provided.
output: a bill

updateItemPrice

Description: The item pricing is updated in the system.
Input: Provide item code and new price.
output: attestation

updateInventory

Description: For each issuance or sale of the item, the system will update the inventory.
input: Provide quantity issued and item code.
output: inventory_status output

calculateThreshold
Description: the application computes from historical consumption data, the reorder threshold value.
input: item_code, past_consumption as input
output: threshold_value as the output

createPurchaseOrder
Description: In case the level of stock falls below the cutoff, a purchase order is produced.
input: Provide code of an item and current stock.
output: The purchase_order output

transcribeInvoice
Description: Keeps the record of an invoice when ingredients or goods are received.
input: Provide item code, received quantity, and price.
output: The invoice_status

procedure Payment
Description: Processes the invoice payment when there is a sufficient cash balance.
Input: cash balance and invoice id
output: The payment_confirmation output

createSalesReport
description: A sales report for the chosen time period of your choice is generated.
input: Time_period as input
output: a report

createExpenseReport
description: An expense report for the given time period is generated.
input: Time_period as input
output: a report

verifyCashBalance
Description: Validates and returns the current cash balance.
input :No input taken.
output: The cash balance output

generateCheque

Description: Creates a check towards an invoice, as per amount due.
Input: amount due, invoice_id
output: cheque

Case 8: Factory Management System

startFactory
description : Machine categories and numbers along with adjusters and MTTF are given as input to form the initial setting of the factory.

Input: mttf_values, num_adjusters, num machines, and machine categories
output: The factory_state

simulateWorkings of a Machine
description: Simulates the working of machines of a particular class.
input: Machine category
output: The machine_status

trackMachineState
Description: Keeps track of and updates the status of machines of a certain category.
input: Machine category
output: The updated_status

Add Machine to Queue
Description: It puts a broken machine in the service queue so that it could be fixed.
input: apparatus
output: The queue_status

putAdjusterToMachine in charge
Description: Assigns a machine which is in need of repair to an available adjuster.
Input: machine, adjuster
output: The confirmation of assignment

recordAdjuster's Availability
Description: It keeps track of the adjuster's availability.
input: Adjuster

output: availability_status is the output.

replicatingRepairProcess
description: Summarizes the machine repairing process by an adjuster.
Input: machine, adjuster
output: The repair_status

updateQueue:
Description: Concerned with waiting machines and adjustments for queue updates.
Input: None
output: The updated_queue

computeMachineUptime
Description: Computed the uptime of a machine given the operating and repairing time.
input: No input taken.
output: The machine_uptime returned

Calculate Adjuster Utilization
Description: Computed adjusters utilization rate based on business of the adjusters.
input: nothing is put
output: Adjuster utilization is the return value.

generate Simulation Report
description: Generates a report with machine uptime and adjuster utilization summarizing the manufacturing simulation.
Input: adjuster utilization, machine uptime, and factory configuration
output: Simulation_report

adjuster allocation optimization
Description: Optimizes adjuster allocation to improve adjuster utilization and machine uptime.
Input: None
Output: Optimal Adjuster Allocation

Factory Parameter Adjustment
Description: Factory parameters are updated according to the results of simulation to gain more efficiency.
input: Factory_configuration.
output: The adjusted_configuration

Case 10: Supermarket Automation Software (SAS)

handle Sales Transaction
Description: This application develops a bill and lists down all the items along with their quantities to generate a sales transaction.
input: item_list as the input
output: bill

generateBill
Description: In respect of a sales transaction, it prepares detailed information of a bill.
input: Sales_transaction is provided.
output: The bill_details

post-sale inventory updates
Description: It shows the amount present in the stock after a particular item has been sold.
input: item_list
output: The inventory_status

Details of Observation
Description: It returns all information about the inventory in terms of the query parameters.
input: Query parameters
output: The inventory_details

inventoryOnSupply
Description: It updates the inventory once fresh products reach the supermarket.
input: entered Supply_list

output: The inventory_status

printSalesData Analysis
Description: It prints the sales data, which consists of quantity sold, revenue, and profit on a particular day.
input: Date on which record has to be entered.
output: Sales_statistics

updateItemPrice
Description: Change item current price according to the changes in market.
input: Item code with new price is given.
output: Price_update_status.

generateSalesReport
Description: Generate report on sales, showing performance for the given date or period.
input: Provide date on which/which the sales report has to be generated.
output: A report

calculateProfit
Description: Calculates the revenue from sold items by multiplying their cost with their selling price.
input: item_list
output: revenue

Case Study 13: CASE tool for Structured Design

R.1: Components

Description: The user should be able to draw modules, control arrows, and data flow arrows. Also symbols for library modules should be provided. The data flow arrows are annotated with the corresponding date name.

Input: Module Selection

Output: The software will detect the module and implement the corresponding symbol or art.

R.2: Operations

Description: The diagrams should be organized in neat hierarchical levels. The user should be able to modify his design. Please note that when he deletes a data flow arrow, the annotated data name automatically gets deleted. For large software, modules may be hierarchically organized and clicking on a module should be able to show its internal organization.

Input: Options selected

Output: The software will detect the option selected and works with respect to it.

R.3 : Save Art

Description: The user should be able to save his design and also be able to load previously created designs.

Input: Filename

Output: The software will prompt for the filename to save the file the user was working on.

Case Study 14: Newspaper Agency Automation Software:

R.1: Shortest-way Destination

Description: For each delivery person, the system must print each day the publications to be delivered to each address. The addresses should be generated in consecutive order as far as possible so that the commutation of the delivery person is minimal.

Input: Source address, destination address

Output: The software must generate the shortest, efficient and safe path with proper indications.

R.2: Publication Generation

Description: For each delivery person, the system must print each day the publication to be delivered to each address. The system should also print for the newsagent the information regarding who received what publications and summary information of the current month.

Input: delivery person id, customer id, destination address, publication name, name of the month

Output: The software should classify the publications for each customer, providing the suitable path to the delivery person to deliver the things. It should also print these details to the newsagent by summarizing monthly.

R.3: Bill Generation:

R.3.1: Issue Bill

Description: At the beginning of every month bills are printed by the system to be delivered to the customers. These bills should be computed by the system automatically and should include the publication type, the number of copies delivered during the month, and the cost for these.

Input: Publication amount per day.

Output: The software computes the amount to be paid by the customer and notifies them every month in detail.

R.3.2: Pay Bill

Description: Customers usually pay their monthly dues either by cheques or cash. Once the cheque number or cash received is entered in the system, receipt for the customer should be printed.

Input: Paid Amount, cheques_id, bank details of customer in cheque.

Output: The system computes the amount to be paid by the customer, and notifies once the payment is completed with no dues.

R.4 : Subscription Update

R.4.1 : Create Subscription

Description: Customers usually subscribe to one or more newspapers and magazines. They are allowed to change their subscription list by giving one week's advance notice.

Input: Newspaper or/and magazine name, subscription type

Output: The system creates a note of choice done by customer, and classifies for creating subscription

R.4.2 : Pause or Modify Subscription

Description: The customers may ask for stopping the deliveries to them for certain periods when they go out of station. Customers may request to subscribe to new newspapers/magazines, modify their subscription list, or stop their subscription altogether.

Input: customer_id, subscription info, request type

Output: The system pauses the subscription and notifies it to the customer and news agency.

R.4.3 : Subscription Reminder

Description: If any customer has any outstanding due for more than one month, a polite reminder message is printed for him and his subscription is discontinued if his dues remain outstanding for a period of more than two months.

Input: Customer_Id, count of month

Output: The system will issue a notification to customers regarding the outstanding and inactivity of subscription made.

R.5 : Commission Generation

Description: The software should compute and print out the amount payable to each delivery boy. Each delivery boy gets 2.5% of the value of the publications delivered by him.

Input: Commision rate, publication value, bank details of delivery person

Output: The software computes the commission amount and notifies the delivery guy about the commision amount being paid to their account.

Case Study 16: Medicine Shop Software(MSS):

R.1: Order Item

Description: The shop owner would request the computer to generate the items to be ordered. The shop owner must be able to order items as soon as the number of items in the inventory reduces below a threshold value.

Input: Item name, count of items

Output: Notification in the form of a popup to be informed to shop owners by the software.

R.2: Operations:

R.2.1: Average sale

Description: To calculate the threshold value for each item, the software must be able to calculate the average number of medicines sales for one week for each part.

Input: Medicines name, count of week.

Output: Software should display the average number of medicines sales for one week for each part.

R.2.2: Medicine Details

Description: The computer should print out the medicine description, the quantity required, and the address of the vendor supplying the medicine.

Input: Medicine Name, quantity, vendor address

Output: Software should display the details of medicine and its supplier details.

R.2.3: List Vendor

Description: It should also prepare a vendor-wise list of the expired items so that the shop owner can ask the vendor to replace these items.

Input: Medicine name, expiry date

Output: The software should display the details of the vendor, who tried to sell the expired items.

R.3 Bill Generation:

R.3.1: Cash Receipt

Description: Whenever any sales occur, the shop owner would enter the code number of each medicine and the corresponding quantity sold. The MSS should print out the cash receipt.

Input: Code number, medicine unique_id,

Output: The software should display the popup, that item is sold, and the cash receipt has to be printed.

R.3.2: Shop Budget-Expenditure

Description: The computer should also generate the revenue for each day and at the end of the month, the computer should generate a graph showing the total sales for each day of the month and also these figures for any given medicine

Input: Daily sale amount, medicine name

Output: The software should display the details of the revenue for each day, month and graphical representation using graphs for respective medicine.