

## **LogBook : 12**

**Meeting No: 12**

**Date:** 17<sup>th</sup> March, 2023

**Start Time:** 10:00 am

**Finish Time:** 10:50 am

---

### **Items discussed:**

1. In the meeting, continuation to the discussion of the final report was given on which, the format and marking scheme of the final report were discussed thoroughly. The remaining chapters of the report that were yet to be discussed on the previous meeting were prioritized on the following day where, we discussed on the chapters like implementation, analysis ,and conclusion. Our progress up to date were also discussed on the meeting. We were reminded to accomplish the draft of final report within the end of March.

### **Achievements:**

#### **1. Methodology chapter of final year report:**

Gave continuity to the final year report and finalized the third chapter i.e Methodology chapter. Documentation on the considered methodologies, Selected methodology, Reason for choosing the selected methodology and the stages of the respective methodology was documented.

### **Problems (if any):**

Started frontend of the dashboard page but since the deadline of the coursework was on Wednesday, I couldn't manage time to fix the errors occurred while coding.

**Tasks for Next Meeting:**

1. Fix the error and complete frontend of profile/dashboard page.
2. Requirement gathering, feasibility study chapter of the final report.
3. Frontend of progress page.

.....

**Alisha Poudel**

.....

**Mr. Abhinav Dahal**

Proof of my work:

Chapter 3 of final report: Methodology

3	Other additional app for other purposes (step tracking, calorie burnt calculator)	Yes	Yes	No	No
4	Ideal Weight Display	No	No	Yes	Yes
5	Calorie Counter	Yes	Yes	Yes	Yes
6	Food Tracker	Yes	Yes	Yes	Yes
7	Macro Nutrition Counter	Yes	Yes	No	No
8	Premium Features	Yes	Yes	No	Yes

### 3. Methodology:

#### 3.1 Methodology Considerations

Since a one-size-fits-all methodology does not exist, different methodologies are practiced according to the needs of software development teams and the demands of the clients. Therefore, some of the effective, as well as often practiced software development methodologies, are elaborated below:

#### 3.1.1 Prototyping Model:

The fundamental concept behind the prototype model is that a throwaway prototype is made to understand the requirements rather than freezing them before a design or coding can be done. Based on the needs that are currently known, this prototype was created. Because of the interactions with the prototype, the client can better grasp the needs of the desired system by using it to gain a "real feel" for the system. For complex and massive systems without an established system or manual procedure to assist define the requirements, prototyping is an appealing option. (TryQa, 2022)

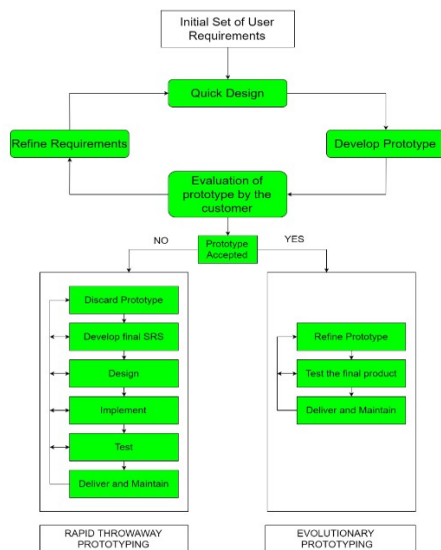


Figure 10: Prototyping Model

(Greeks, 2022)

#### Advantages:

- Active participation of users in the development
- Users using this methodology have a better grasp of the system being developed because a functioning model of the system is supplied.
- Much earlier error detection is possible.

#### Disadvantages:

- Leads to the use of systems construction techniques followed by repairs.
- Practically speaking, this process might make the system more complex because the system's scope might go beyond initial projections.
- A partially completed application could prevent it from functioning as the full system was intended.
- Improper or insufficient problem analysis. (Greeks, 2022)

### 3.1.2 Incremental Model

The incremental model, sometimes referred to as the successive version model, is a widely used method of software development in which the SDLC divides the software requirements into numerous independent modules or increments (Software Development Life Cycle). Every increment follows the SDLC incremental paradigm and is regarded as a sub-project throughout. This has an iterative model-like ring to it. The incremental model is also known as the iterative enhancement model because this model is an improvement over the iterative model. Instead of taking a giant stride forward, we accomplish our goals using the gradual strategy. (InterviewBit, 2022)

- By using this strategy, we can reduce the cost of initial delivery.
- When there is a change in the demand or the scope, this model is flexible and costs less.
- The choice in the subsequent stage will be made. (Pedamkar, 2022)

#### Disadvantages:

- It necessitates careful planning and design.
- Problems may arise because not all requirements are gathered up front for the full software lifecycle owing to system architecture.
- Each iteration phase is distinct from the others and is strict.
- Correcting an issue in one unit necessitates fixing the problem in all the units, which takes a lot of time.

### 3.1.3 Rational Unified Process

An object-oriented model software development process is called Rational Unified Process (RUP). A different name for it is the Unified Process Model. It was made by Rational Corporation, and

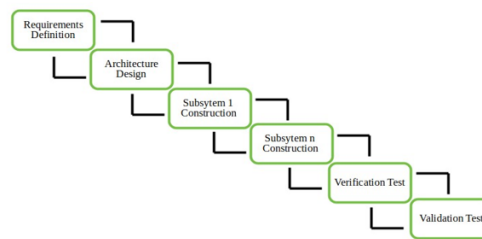


Figure 11:

Incremental Model (Greeks for greeks, 2022)

#### Advantages:

- It will be ensured that all requirements and software objectives are met since the object will be broken down into incremental steps.
- Since testing is done at each incremental phase, the program will be tested more often, which leads to better results and fewer faults.

UML was used in the design and documentation (Unified Modeling Language). The IBM Rational Method Composer (RMC) software includes this procedure. With the help of IBM (International Business Machine Corporation), we may build, create, and alter the unified process. Ivar Jacobson, Grady Bootch, and James Rumbaugh put forth RUP. Use-case driven, incremental (growth in value) by nature, iterative (process repetition), distributed online via web technology, adaptable in modular and electronic form, etc. are some traits of RUP. RUP prevents resource waste and lowers unforeseen development costs. (Lucid Chart, 2021)

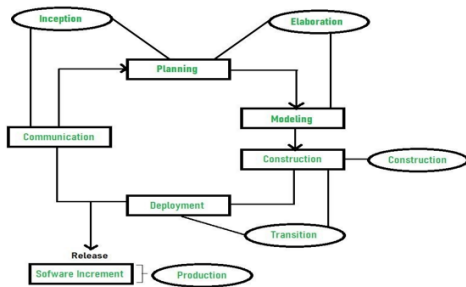


Figure 12: RUP Methodology

#### Advantages:

- Good documentation is provided, and the procedure is finished on its own.

they favor providing it through recurrent updates. Or the client demands some functionality upgrades while the project is still being developed. The iterative model is applied in these circumstances.

The software development life cycle is implemented using the iterative process model, in which additional features are added to the base software product over the course of successive iterations until the final system is produced. The first development is started based on the initial requirements. In essence, the iterative model divides a very large application's software development process into smaller chunks. (Martins, 2022)

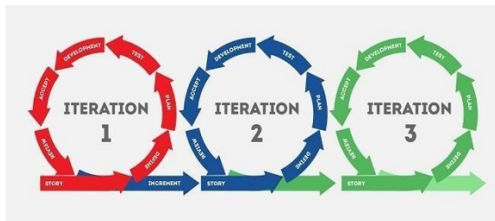


Figure 13: Iterative Methodology

#### Advantages:

- During the SDLC, this model creates a functioning program significantly more fast.
- This framework is very adaptable. Since it is still under construction, additional features can be added at any time.
- Compared to other process models, this one is far less expensive to modify the requirements.
- Quick input from the stakeholders or the end-user can then be incorporated into the system.

#### Disadvantages:

- It is a poor option for little projects.
- Greater resource requirements than the waterfall model.
- Checking the hazards in our system demands highly qualified people with experience in risk analysis.
- Managing the entire process is challenging. (Martins, 2022)

- It offers assistance with risk management.
  - Because the components are reused, the overall time is shorter.
  - Training and tutorials are both excellent forms of online support.
- Disadvantages:

- The process is complex; thus, a team of knowledgeable professionals is needed.
- Process that is difficult and poorly planned.
- increased reliance on risk management
- Repeated integration is challenging.

### 3.2 Reason for selecting the Iterative Methodology

Note: Previously in the proposal, I proposed to use RUP methodology for my final year project. But due to lack of research, it resulted as wrong decision and works could not be completed as per the Gantt chart of RUP. The concept of RUP was too vague and hence, I decided to select Iterative Methodology for my final year project.

The Iterative Software Development Life Cycle Model is the most appropriate software development technique for the selected project because it does not demand that the entire set of requirements be distinct before the project begins. The "Diet Diary" development process began with the functional part's needs, which were then enlarged in accordance with conversations with the supervisor. The repeating procedure enables the creation of fresh iterations of the

finished and functional product at each iteration that is possibly deliverable.

### 3.3 Stages of the chosen Methodology

A way to continuously refining a concept, design, or product is the iterative process. In order to come closer to the solution, developers create a prototype, test it, make necessary adjustments, and repeat the process. Following are the stages carried out during application development using Iterative Methodology.

#### First stage:

Planning and requirements gathering-Create a plan and timeframe for the first iterative cycle, sketch out the first requirements, and collect the pertinent documentation during this phase.

#### Second stage:

Analysis and design- Finalize the technological requirements, da-

product for each cycle, which lasts for two to three weeks. Every iteration of "Diet Diary" involves the creation of a unique system component, which is then added to the functional portion that was created before.

Since the project needs are hazy, unstable, and couldn't be determined precisely at the pre-development stage, the iterative process is mostly used to meet new requirements that arise as the project is developed. Due to the accelerated project delivery and scattered needs, an iterative model was chosen for the project. Nevertheless, the project's supervisor and friends acted as the product's user or client and provided input on the system while it was being developed through each iteration. The supervisor's and friends' comments helped to improve the product in subsequent revisions.

The processes were repeated several times until the project began to take shape as the finished system through the integration of diverse functionalities from previous project iterations. This methodology was chosen because it guarantees the development of a

algorithm, schematic, or working architecture that complies with your requirements.

#### Third stage:

Implementation- Create the functionality and style necessary to satisfy the requirements.

#### Fourth stage:

Testing - Find out what isn't working or performing as expected. Users, testers, and stakeholders offer their opinions and personal experiences.

#### Fifth stage:

Evaluation and Review- Evaluate this iteration against the specifications and goals. (Eby, 2019)

Errors encountered while coding frontend of dashboard page:

body.dart 6, U   profile\_menu.dart 2, U X   profile\_pic.dart 1, U

> profile\_menu.dart > ProfileMenu

```

    RoundedRectangleBorder(borderRadius: BorderRadius.
    backgroundColor: const Color(0xFFF5F6F9),
  ),
  onPressed: press,
  child: Row(
    children: [
      SvgPicture.asset(
        icon,
        color: kPrimaryColor,
        width: 22,
      ),
      const SizedBox(width: 20),
      Expanded(child: Text(text)),
      const Icon(Icons.arrow_forward_ios),
    ],
  ), // Row
), // TextButton
); // Padding
}

```

body.dart 6, U   profile\_menu.dart 2, U   profile\_pic.dart 1, U X   SvgPic

profile\_pic.dart > ProfilePic > build

```

    backgroundColor: const Color(0xFFF5F6F9),
  ),
  onPressed: () {},
  child: SvgPicture.asset("assets/icons/Camera Icon.svg"),
), // TextButton
), // SizedBox
) // Positioned
],
), // Stack
); // SizedBox
}

```

```
child: Column(  
  children: [  
    ProfilePic(),  
    const SizedBox(height: 20),  
    ProfileMenu(  
      text: "My Account",  
      icon: "assets/icons/User Icon.svg",  
      press: () => {},  
    ),  
    ProfileMenu(  
      text: "Notifications",  
      icon: "assets/icons/Bell.svg",  
      press: () {},  
    ),  
    ProfileMenu(  
      text: "Settings",  
      icon: "assets/icons/Settings.svg",  
      press: () {},  
    ),  
  ],  
)
```