|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Experiment Number: 1** | | | | | |
| **Date of Performance:** | | **01-08-2022** | | | |
| **Date of Submission:** | | **08-08-2022** | | | |
| **Program Execution/ formation/ correction/ ethical practices (07)** | **Documentation (02)** | **Timely Submission (03)** | **Viva Answer to sample questions (03)** | **Experiment Total (15)** | **Sign** |
| **7** | **2** | **3** | **3** | **15** |  |

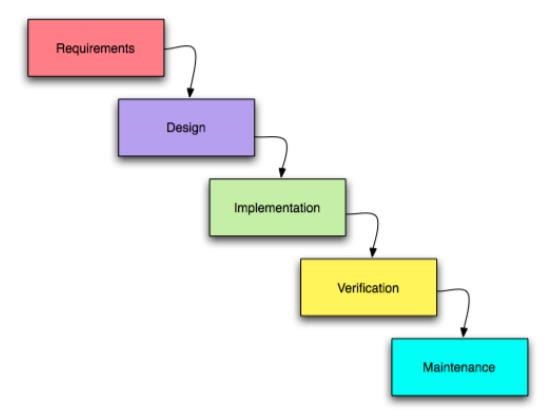
Experiment No: 1

**Aim:** Application of at least two traditional process models.

● **Waterfall Model**

* **Problem Statement: Banking System**

**Waterfall model**



* **Description:**

# Requirements & Documents

First, you must gather all the requirements and documentation you need to get started on the app.

▪ Project Scope: This is one of the most important documents in your project, where you determine what the goals associated with building your app are: deliverables, features, deadlines, costs, and so on.

▪ Stakeholder Expectations: In order to align the project scope with the expectations of your stakeholders—the people who have a vested interest in the development of the app—you want to conduct interviews and get a clear idea of exactly what they want.

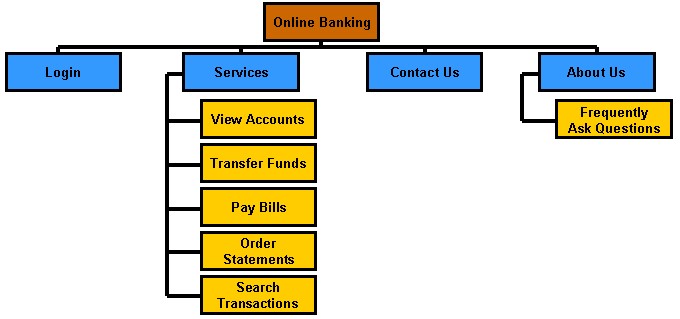
▪ Research: To better serve your plan, do some market research about competing apps, the current market, customer needs and anything else that will help you find the unserved niche your app can serve.

▪ Assemble Team: Now, you need to get the people and resources together who will create the app, from programmers to designers.



**2. System Design:** The members of the team work on the software architecture, high level and low level design for the project.

It is decided that the banking application needs to have redundant backup and failover capabilities such that system is accessible at all timeThe project members creates the Architecture diagrams and high level / low level design documents



**3. Implementation:** Now you’re ready to get started in earnest. This is the phase in which the app will be built and tested.

▪ Assign Team Tasks: Team members will own their tasks and be responsible for completing them, and for collaborating with the rest of the team. You can make these tasks from a Gantt chart and add descriptions, priority, etc.

▪ Monitor & Track: While the team is executing the tasks, you need to monitor and track their progress in order to make sure that the project is moving forward per your schedule.

▪ Manage Resources & Workload: As you monitor, you’ll discover issues and will need to reallocate resources and balance workload to avoid bottlenecks.

▪ Report to Project Guide: Throughout the project, Project Guide need updates to show them progress. Meet with them and discuss a regular schedule for presentations.

**4. Testing:** The testing process tests the complete application and identifies any [**defects**](http://tryqa.com/what-is-defect-or-bugs-or-faults-in-software-testing/) in the application.

These defects are fixed by the developers and the testing team tests the fixes to ensure that the defect is fixed.

They also perform [**regression testing**](http://tryqa.com/what-is-regression-testing-in-software/) of the application to see if any new defects were introduced.

Testers with banking domain knowledge were also prefferd for the project so that they could test the application based on the domain perspective.

Security testing is perform test the security of the banking application.

**5. Maintenance:** Ofcourse, the nature of any software project is that, through use by customer, new bugs will arise and must be squashed. During the maintenance phase, the team ensures that the application is running smoothly on the servers without any downtime.

Issues that are reported after going live are fixed by the team

# Advantages of waterfall model

* This model is simple and easy to understand and use.
* deliverables and a review process.
* In this model phases are processed and completed one at a time. Phases do not overlap.
* Waterfall model works well for smaller projects where requirements are clearly defined and very well understood.

# Disadvantages of waterfall model

* Once an application is in the [**testing**](http://tryqa.com/what-is-a-software-testing/) stage, it is very difficult to go back and change something that was not well-thought out in the concept stage.
* No working software is produced until late during the life cycle.
* High amounts of risk and uncertainty.
* Not a good model for complex and object-oriented projects.
* Poor model for long and ongoing projects.
* Not suitable for the projects where requirements are at a moderate to high risk of changing.

# When to use the waterfall model

* This model is used only when the requirements are very well known, clear and fixed.
* Product definition is stable.
* Technology is understood.
* There are no ambiguous requirements
* Ample resources with required expertise are available freely● The project is short.

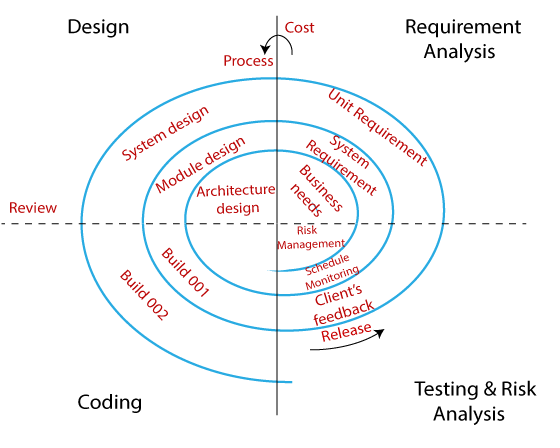
In Waterfall model, very less customer interaction is involved during the development of the product. Once the product is ready then only it can be demonstrated to the end users.

Once the product is developed and if any failure occurs then the cost of fixing such issues are very high, because we need to update everything from document till the logic.

## ● Spiral Model

o  **Problem Statement:** Banking System

**Spiral Model**



The biggest problem we face in the waterfall model is that taking a long duration to complete the product, and the software became outdated. To solve this problem, we have a new approach, which is known as the Spiral model. The spiral model is also known as the cyclic model.

In this model, we create the application module by module and handed over to the customer so that they can start using the application at a very early stage. And we prepare this model only when the module is dependent on each other. In this model, we develop the application in the stages because sometimes the client gives the requirements in between the process.

### Requirement Analysis

The spiral model process starts with collecting business needs. In this, the following spirals will include the documentation of system requirements, unit requirements, and the subsystem needs. In this stage, we can easily understand the system requirements because the business analyst and the client have constant communication. And once the cycle is completed, the application will be deployed in the market.

### Design

The second stage of the spiral model is designed, where we will plan the logical design, architectural design, flow charts, decision tree, and so on.

### Coding

After the compilation of the design stage, we will move to our next step, which is the coding stage. In this, we will develop the product based on the client's requirement and getting the client's feedback as well. This stage refers to the construction of the real application in every cycle.

And those spirals had an excellent clarity of the requirements, and the design details of an application are known as the build with having version numbers. After that, these builds are transferred to the client for their responses.

### Testing and Risk Analysis

Once the development is completed successfully, we will test the build at the end of the first cycle and also analyze the risk of the software on the different aspects such as managing risks, detecting, and observing the technical feasibility. And after that, the client will test the application and give feedback.

**Advantages of Spiral Model:**

* Flexible changes are allowed in spiral model.

* The development can be distributed into smaller parts.

* The customer can use the application at an early stage also.

●

More clarity for Developers and Test engineers

* It will provide the wide use of prototypes.

**Disadvantages of Spiral Model:**

* It is not suitable for the small and low-risk product because it could be costly for a smaller project.

* It is a traditional model, and thus developers only did the testing job as well.

* There is no requirement of review process and no parallel deliverables allowed in the spiral model.

* In the spiral model, management is a bit difficult; that's why it is a complex process.’

* The maximum number of intermediate phases needs unnecessary paperwork.