

Q1. Explain the problem statement

Many academic conferences are held annually which leads to an increase in the number of submitted papers and the amount of work necessary to manage the submissions and review process. Such an intricate workflow of conference management results in this process being anticipated with great apprehension.

In order to meet the requirements & automate the activities and processes of a conference, we propose a website which will offer management tools.

Q2. Identify the different types of users for the project.

Different Users:

- Administrator
- Author
- Reviewer
- Chair

Q3. Identify different major functionalities and their subfunctions for your project.

Main Function & Sub Function:

1. The author of the research paper will register for the conference.

Subfunctions:

1. Generate a unique id on registration.
2. Request the user to upload the research paper.
3. Payment of registration fee.

2. Selection of experts for evaluation and review.

Subfunctions:

1. Research scholars are selected and papers are provided to them.
2. Allocation of domains according to their expertise.

3. Acceptance or rejection of paper as per review.

Subfunctions:

1. Letting the author know about the result.
2. Providing information about changes to be done
3. Re-evaluation or submission

4. Making arrangements for the event.

Subfunctions:

1. Selection and booking of the venue.

2. Supplying the hardware and software requirements.
 3. Printing and distribution of research papers.
5. Taking care of security in the event.

Q4. For each function state its feasibility.

Feasibility Table:

No.	Functions		Technical	Operational	Economical
1.	Registration for conference	Generation of unique id on registration	Yes	Yes	Yes
		Request user to upload the research paper	Yes	Yes	Yes
		Payment of registration fee	Yes	Yes	Yes
2.	Selection of experts for evaluation and review	Research scholars are selected and papers are provided to them	Yes	Yes	Yes
		Allocation of domains according to their expertise.	Yes	Yes	Yes
3.	Acceptance or rejection of paper as per review	Letting the author know about the result	Yes	Yes	Yes
		Providing information about changes to be done	Yes	Yes	Yes
		Re-evaluation or submission	Yes	Yes	Yes
4.	Making arrangements for the event	Selection and booking of the venue.	Yes	Yes	Yes

		Supplying the hardware and software requirements.	Yes	Yes	No
		Printing and distribution of research papers.	Yes	Yes	No
5.	Taking care of security in the event		Yes	No	Yes

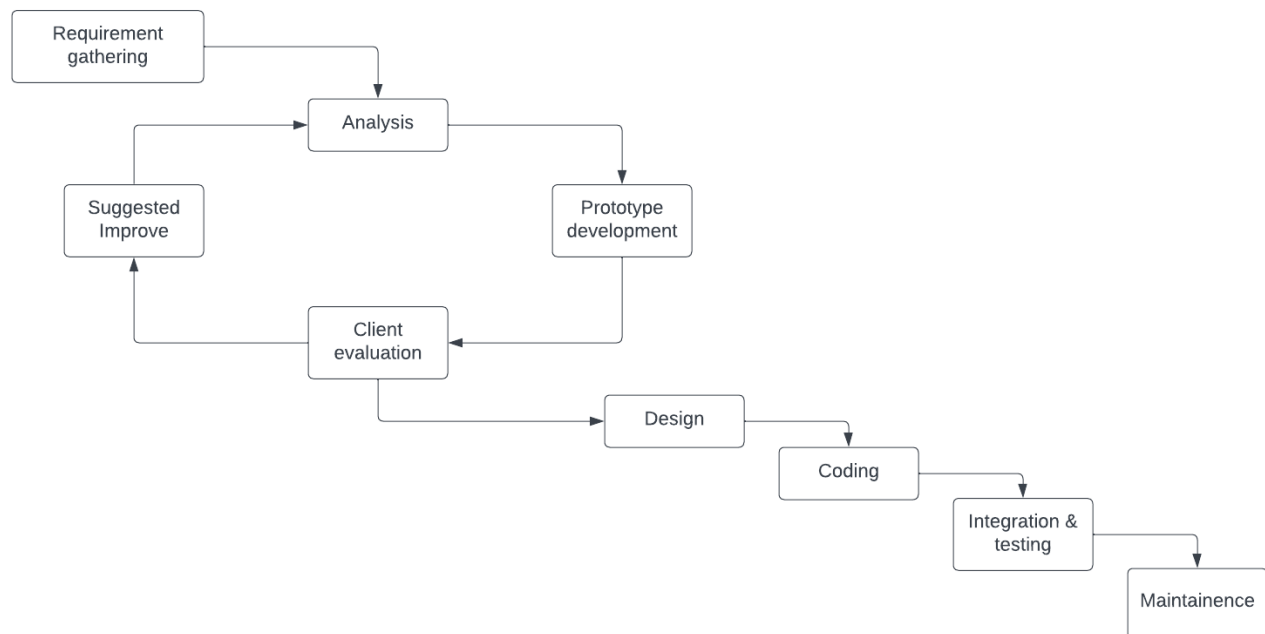
Q5. Identify the SDLC model for your project based on the three criteria- based on software, based on the development team, and based on the customer.

The software development model we will choose for our project is an **evolutionary model**.

This is based on designing a little, building a little, testing a little, and deploying a little. Hence here we do not have to wait for all the functionalities to get specified, as soon as one feature is elaborated we can quickly start its implementation.

Reason:

1. It is based on feedback and then proceeding further.
2. Well suited for projects with dynamic changes in design in execution
3. This model at the end gives a refined final product



Reasons to choose this model over all others:

a) Prototyping Model

- This model is costly with respect to time and, money
- Poor documentation

b) Iterative Waterfall Model

- This process goes traditionally with step by step implementation
- Takes a lot of time

c) Spiral Model

- This is a very complex model & hence requires time & effort to manage

As the conference management system is a big project there would be many functionalities, and incorporating client-side changes is also very easy in the case of an evolutionary model. Hence we prefer it over an iterative waterfall and prototyping. We use the spiral model in case of a risky project conference system not being a risky project we avoid using the spiral model.

Q. Prepare SPMP document for your project

INTRODUCTION

Conference Management System is a powerful tool for simplifying the management of conferences, meetings, and events. With this system, it is possible to easily create, manage, and organize events, invite guests, and manage bookings and payments. The system is designed to streamline the process of organizing and managing conferences, meetings, and other events.

The Conference Management System offers a range of features to simplify the management and organization of conferences, meetings, and events. It includes a comprehensive suite of event management tools, including registration management, ticketing, scheduling, and event reporting. It also includes features to aid in the preparation of events, such as budgeting, venue selection, and guest list management. The system also provides customizable guest invitations and booking forms, along with automated email communications for event management.

OBJECTIVE

To create a user-friendly, efficient, and cost-effective online platform for managing conferences and associated events. The platform should allow for easy registration and payment processing, as well as allow for the efficient management of events, speakers, exhibitors, and other aspects of the conference. It should also provide an effective platform for communication and collaboration between conference organizers and other stakeholders. Additionally, it should provide a secure environment for data storage and management.

MAJOR-FUNCTIONS:

- Registrations for conferences
- Submitting papers
- Reviewing Papers
- Organising Events
- Event Management

COST ESTIMATION:

There are two methods to calculate the size of our application:

- LOC (i.e. Lines Of Codes)
- FP (Function Point)

LOC (Lines of Code): It is the quick and popular method done by dividing the project into smaller parts, calculating the lines of code for each part and adding it.

Disadvantage:

1. Complex for large projects
2. Not Accurate
3. Don't take logic into account

FP (Function Point): It overcomes the shortcomings of LOC. It is used to estimate the size of a project based on no. of functionality, no. of files, no. of interfaces, and no. of input and output.

Steps to finding Function Point:

Step 1: Find Unadjusted Function Point (UFP)

$$\text{UPF} = [\text{no. of inputs} * 4] + [\text{no. of outputs} * 5] + [\text{no. of inquiry} * 4] + [\text{no. of files} * 10] + [\text{no. of interface} * 10]$$

No. of inputs = 25

No. of outputs = 45

No. of inquiries = 5

No. of files = 4

No. of interface = 5

$$\text{UPF} = [25 * 4] + [45 * 5] + [5 * 4] + [4 * 10] + [5 * 10] = 435$$

Step 2: Find Degree of Influence (Di)

1. Di is ranging from 0 to 6. 0 is the least influence, 6 is the most influence.
2. Di depends upon the following factors.

Influence factors	Degree of Influence
Data Communication	6
Distributed Data Configuration	0
Performance	5
Heavily Used Configuration	5
Transaction Rate	6
Online Data Entry	5
End User Efficiency	2
Online Update	5
Complex Processing	1
Reusability	5
Installation Ease	5
Operational Ease	4
Multiple Sites	2
Facilitated Change	3
Total DI	54

Step 3: Finding Technical Complexity Factor (TCF)

$$\text{TCF} = 0.65 + 0.01 * \text{Di} = 0.65 + 0.01 * 54 = 1.19$$

Step 4: Finding Function Point (FP)

$$\text{FP} = \text{UFP} * \text{TCF} = 435 * 1.19 = 518$$

Disadvantage:

1. It assumes all the functionality having same complexity on each module.

COST ESTIMATION:

1. Empirical Estimation Technique
2. Heuristic Technique
3. Analytical Estimation Technique

Heuristic Technique: we will be using this technique to estimate the cost of the application.

It categorizes every project into three categories.

1. Organic
2. Semi-Detached
3. Embedded

In Semi-Detached, there are three different types of COCOMO models.

1. Basic COCOMO
2. Intermediate COCOMO
3. Complete COCOMO

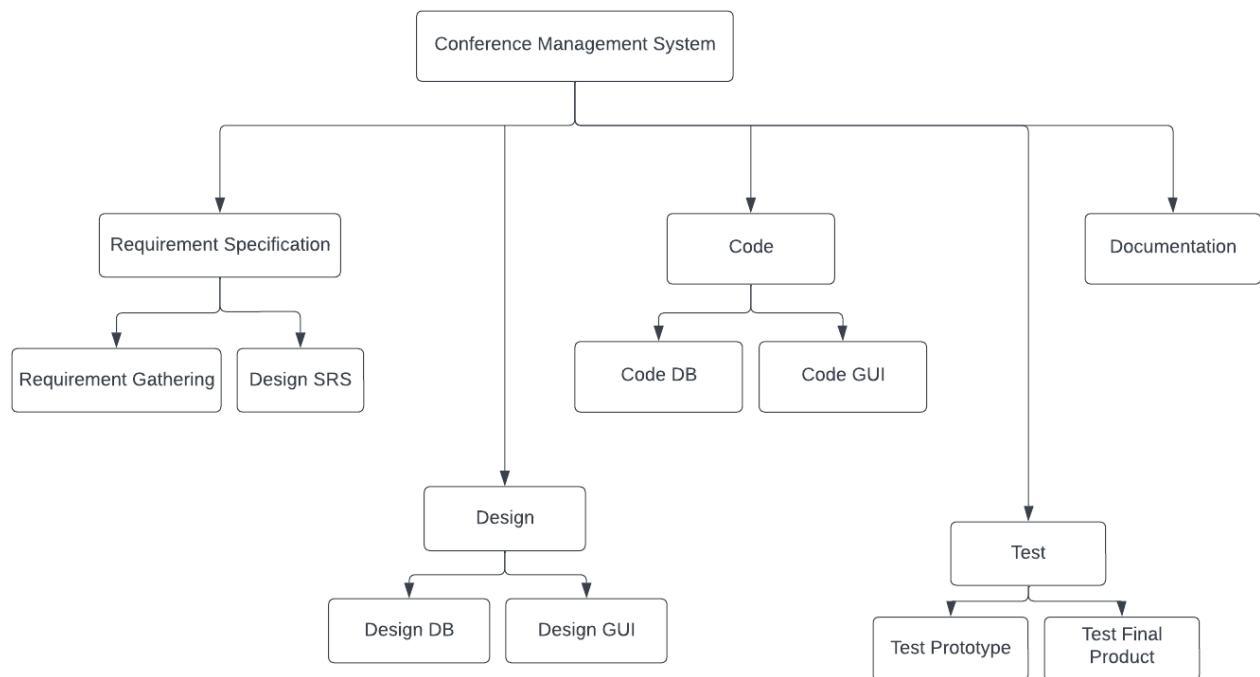
For our project we can use Basic **COCOMO**. In Basic COCOMO,

- **Effort** = $a_1 * (KLOC)^{a_2}$ PM
- **T dev** = $b_1 * (Effort)^{b_2}$ MONTHS
- **Cost** = Effort * Monthly salary * T dev

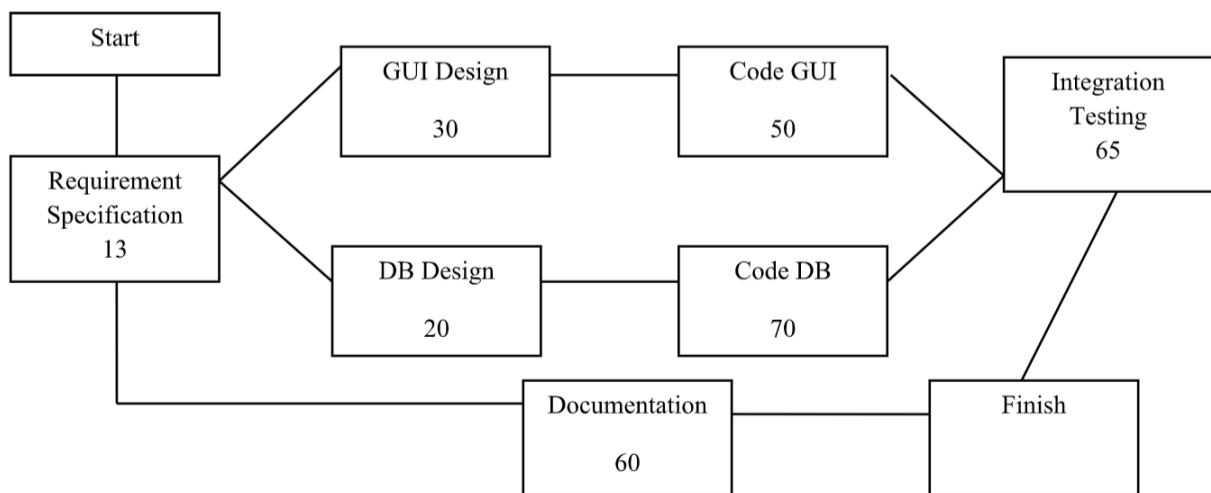
In our project, KLOC = 1.2, $a_1 = 3.0$, $a_2 = 1.12$, $b_1 = 2.5$, $b_2 = 0.35$, monthly salary = 15000

- **Effort** = $3.0 * (1.2)^{1.12} = 3.67 = 4$ PM
- **T dev** = $2.5 * (4.72)^{0.35} = 3.94 = 4$ MONTHS
- **Cost** = $4 * 15000 * 4 = 2,40,000$

Q. Work Breakdown Structure



Q. Activity Network Diagram



Activity Selection Table

Tasks	ES	LS	EF	LF	Slack Time
Requirement Specification	0	0	13	13	0
Design GUI	13	23	43	53	10
Design Database	13	13	33	33	0
Code GUI	43	53	93	103	10
Code DB	33	33	103	103	0
Integration Testing	93	103	158	168	10
Documentation	13	128	53	168	115