School of Computing
SRM IST, Kattankulathur – 603 203

Course Code: 21CSC303J

Course Name: Software Engineering and Project Management

Experiment No	**2**
Title of Experiment	**Analyse Stakeholder and User Description and Identify the appropriate Process Model**
Name of the candidate	**SAATVIK AGNIHOTRI**
Team Members	**RIYA RAI, MD. DILSHAD ALAM**
Register Number	**RA2211003011922, RA2211003011919, RA2211003011917**
Date of Experiment	**30-01-2025**

Mark Split Up

S.No	Description	Maximum Mark	Mark Obtained
1	Exercise	5	
2	Viva	5	
	Total	**10**	

Staff Signature with date

Aim : Analyse Stakeholder and User Description and Identify the appropriate Process Model

Team Members:

S. No	Register No	Name	Role
1	RA2211003011922	SAATVIK AGNIHOTRI	Lead
2	RA2211003011919	RIYA RAI	Member
3	RA2211003011917	MD. DILSHAD ALAM	Member

Project Title: **AI Technical Document Generator(AutoDocX)**

Project Description :

The AI-Based Technical Documentation Generator is an automated system designed to generate human-readable, concise, and accurate documentation for software projects. This system integrates directly with version control tools like GitHub, triggering the creation of documentation each time new code is committed. It analyzes the changes in the codebase (including new functions, classes, and modifications to existing ones), generates summaries, and organizes them in a centralized documentation file (e.g., Markdown or HTML). This eliminates the need for manual documentation, ensuring the documentation is always up-to-date and aligned with the latest code changes.

USER STORY

AI Technical Document Generator(AutoDocX)

Here's a high-level user story for your AI-Based Technical Documentation Generator:

User Story

Who are we building this for?

- Software developers, engineering teams, and technical writers who work on software projects and need up-to-date documentation.

What are they trying to achieve?

- They want to automate the process of generating and maintaining technical documentation without manually writing updates every time the code changes.

What's the overall benefit they're trying to achieve?

- Ensuring that documentation is always accurate, up-to-date, and aligned with the latest code changes.
- Reducing the time and effort required to maintain documentation.
- Improving collaboration and knowledge sharing across teams by providing structured, easy-to-read documentation.

User Story Format:

- As a software developer, I want an AI-powered system to automatically generate and update technical documentation based on code changes so that I can ensure my project documentation is always up-to-date without manual effort.
- As a project manager, I want to access well-structured, real-time documentation so that I can track project progress and maintain transparency within my team.
- As a technical writer, I want to integrate AI-generated documentation into my workflow so that I can focus on refining content rather than manually documenting every code change.

Linked Tasks

- **AnalyzeCodeChanges:** Detect and analyze new functions, classes, and modifications in the codebase.
- **GenerateDocumentation:** Automatically create structured and human-readable documentation from the analyzed code.
- **IntegrateWithVersionControl:** Connect with GitHub to trigger documentation updates on every commit.
- **FormatOutput:** Generate documentation in Markdown, HTML, or other user-specified formats.
- **NotificationSystem:** Notify users when new documentation updates are available.
- **Testing And Validation**: Ensure the AI-generated documentation is accurate and relevant.

Estimation of Effort

Effort Level: Hard

Reasoning:

- **Complexity of Code Analysis**: Extracting meaningful information from code changes requires advanced static analysis techniques and potentially AI-based summarization models.
- **Integration with GitHub**: Implementing webhook-based triggers and ensuring smooth API communication can be challenging.
- **Documentation Formatting & Structuring**: Generating well-structured Markdown/HTML documentation that remains human-readable and meaningful adds additional complexity.
- **Testing & Validation**: Ensuring AI-generated documentation is contextually accurate and useful requires extensive testing.

Acceptance Criteria

- **Automatic Documentation Generation**
  - Given a new code commit in the repository,
  - When the system detects changes (new functions, classes, or modifications),
  - Then it automatically generates or updates the technical documentation.
- **Integration with Version Control**
  - Given a software project hosted on GitHub,
  - When the system is configured with repository access,
  - Then it successfully integrates and triggers documentation updates on each commit.
- **Documentation Format Support**
  - Given the system generates documentation,
  - When a user selects Markdown or HTML as the output format,
  - Then the generated documentation is properly formatted and readable.
- **User Customization**
  - Given a user accesses the system settings,
  - When they configure documentation rules (e.g., which files or functions to include/exclude),
  - Then the system respects those configurations in future documentation updates.
- **Version History Tracking**
  - Given multiple updates to the codebase,
  - When a user accesses previous versions of the documentation,
  - Then they can retrieve and compare past documentation versions.

- **Notification System**
    - Given a successful documentation update,
    - When a new version is generated,
    - Then the system sends a notification (email, Slack, or dashboard update) to inform the users.
- **Accuracy & Readability**
    - Given a generated documentation file,
    - When reviewed by a developer or technical writer,
    - Then it should be human-readable, concise, and accurately reflect the code changes.
    -
- **Error Handling & Logging**
    - Given an issue occurs during documentation generation,
    - When the system fails to update or parse code changes,
    - Then it logs the error and provides an appropriate error message to the user.

Result:

Thus, the stakeholders and user story has been studied.