**SmartSDLC-AI-**

**Enhanced Software**

**Development Lifecycle**

Project Documentation

Prepared by: [Your Name]

Date: June 28, 2025 at 03:41 PM IST

**Contents**

[**1 INTRODUCTION** **2**](file:///C:\Users\user%201\Desktop\New%20folder\c51c68c7-5363-4c8d-acc8-0a68c4d647b6.pdf#_Toc5237)

[1.1 Project Overview 2](file:///C:\Users\user%201\Desktop\New%20folder\c51c68c7-5363-4c8d-acc8-0a68c4d647b6.pdf#_Toc5238)

[1.2 Purpose 2](file:///C:\Users\user%201\Desktop\New%20folder\c51c68c7-5363-4c8d-acc8-0a68c4d647b6.pdf#_Toc5239)

[**2 IDEATION PHASE** **3**](file:///C:\Users\user%201\Desktop\New%20folder\c51c68c7-5363-4c8d-acc8-0a68c4d647b6.pdf#_Toc5240)

[2.1 Problem Statement 3](file:///C:\Users\user%201\Desktop\New%20folder\c51c68c7-5363-4c8d-acc8-0a68c4d647b6.pdf#_Toc5241)

[2.2 Empathy Map Canvas 3](file:///C:\Users\user%201\Desktop\New%20folder\c51c68c7-5363-4c8d-acc8-0a68c4d647b6.pdf#_Toc5242)

[**3 REQUIREMENT ANALYSIS** **4**](file:///C:\Users\user%201\Desktop\New%20folder\c51c68c7-5363-4c8d-acc8-0a68c4d647b6.pdf#_Toc5243)

[3.1 Customer Journey Map 4](file:///C:\Users\user%201\Desktop\New%20folder\c51c68c7-5363-4c8d-acc8-0a68c4d647b6.pdf#_Toc5244)

[3.2 Solution Requirement 4](file:///C:\Users\user%201\Desktop\New%20folder\c51c68c7-5363-4c8d-acc8-0a68c4d647b6.pdf#_Toc5245)

[3.3 Data Flow Diagram 4](file:///C:\Users\user%201\Desktop\New%20folder\c51c68c7-5363-4c8d-acc8-0a68c4d647b6.pdf#_Toc5246)

[**4 TECHNOLOGY STACK & PROJECT DESIGN** **5**](file:///C:\Users\user%201\Desktop\New%20folder\c51c68c7-5363-4c8d-acc8-0a68c4d647b6.pdf#_Toc5247)

[4.1 Technology Stack 5](file:///C:\Users\user%201\Desktop\New%20folder\c51c68c7-5363-4c8d-acc8-0a68c4d647b6.pdf#_Toc5248)

[4.2 Problem Solution Fit 5](file:///C:\Users\user%201\Desktop\New%20folder\c51c68c7-5363-4c8d-acc8-0a68c4d647b6.pdf#_Toc5249)

[4.3 Proposed Solution 5](file:///C:\Users\user%201\Desktop\New%20folder\c51c68c7-5363-4c8d-acc8-0a68c4d647b6.pdf#_Toc5250)

[4.4 Solution Architecture 5](file:///C:\Users\user%201\Desktop\New%20folder\c51c68c7-5363-4c8d-acc8-0a68c4d647b6.pdf#_Toc5251)

[**5 PROJECT PLANNING & SCHEDULING** **6**](file:///C:\Users\user%201\Desktop\New%20folder\c51c68c7-5363-4c8d-acc8-0a68c4d647b6.pdf#_Toc5252)

[**6 FUNCTIONAL AND PERFORMANCE TESTING** **7**](file:///C:\Users\user%201\Desktop\New%20folder\c51c68c7-5363-4c8d-acc8-0a68c4d647b6.pdf#_Toc5253)

[6.1 Performance Testing 7](file:///C:\Users\user%201\Desktop\New%20folder\c51c68c7-5363-4c8d-acc8-0a68c4d647b6.pdf#_Toc5254)

[6.2 Results 7](file:///C:\Users\user%201\Desktop\New%20folder\c51c68c7-5363-4c8d-acc8-0a68c4d647b6.pdf#_Toc5255)

[6.2.1 Output Screenshots 7](file:///C:\Users\user%201\Desktop\New%20folder\c51c68c7-5363-4c8d-acc8-0a68c4d647b6.pdf#_Toc5256)

[**7 ADVANTAGES & DISADVANTAGES** **8**](file:///C:\Users\user%201\Desktop\New%20folder\c51c68c7-5363-4c8d-acc8-0a68c4d647b6.pdf#_Toc5257)

[**8 CONCLUSION** **9**](file:///C:\Users\user%201\Desktop\New%20folder\c51c68c7-5363-4c8d-acc8-0a68c4d647b6.pdf#_Toc5258)

[**9 FUTURE SCOPE** **10**](file:///C:\Users\user%201\Desktop\New%20folder\c51c68c7-5363-4c8d-acc8-0a68c4d647b6.pdf#_Toc5259)

[**10 APPENDIX** **11**](file:///C:\Users\user%201\Desktop\New%20folder\c51c68c7-5363-4c8d-acc8-0a68c4d647b6.pdf#_Toc5260)

[10.1 Source Code (if any) 11](file:///C:\Users\user%201\Desktop\New%20folder\c51c68c7-5363-4c8d-acc8-0a68c4d647b6.pdf#_Toc5261)

[10.2 Dataset Link 11](file:///C:\Users\user%201\Desktop\New%20folder\c51c68c7-5363-4c8d-acc8-0a68c4d647b6.pdf#_Toc5262)

[10.3 GitHub & Project Demo Link 11](file:///C:\Users\user%201\Desktop\New%20folder\c51c68c7-5363-4c8d-acc8-0a68c4d647b6.pdf#_Toc5263)

**Chapter 1**

**INTRODUCTION**

## Project Overview

To create an AI-driven platform that augments and automates every phase of the Software Development Lifecycle (SDLC) — from requirements gathering to deployment and maintenance — thereby improving productivity, code quality, and delivery speed.

## Purpose

SmartSDLC\* aims to revolutionize the traditional software development process by embedding \*artificial intelligence and automation\* into every phase of the Software Development Lifecycle (SDLC). The purpose of this project is to:

\*Enhance Developer Efficiency

\*Improve Software Quality

\*Accelerate Time-to-Market

\*Facilitate Better Decision-Making

\*Enable Scalable Consistent Development

**Chapter 2**

# IDEATION PHASE

## 2.1 Problem Statement

"Current software development lifecycles are often manual, time-consuming, and prone to errors, resulting in delayed project timelines, increased costs, and reduced quality."

## 2.2 Empathy Map Canvas

An Empathy Map Canvas is a tool used to understand the needs, desires, and pain points of users. Here's an example of an Empathy Map Canvas for Smart SDLC:

1. Says: "I want to deliver high-quality software faster."

2. Does: Manages development lifecycle manually.

3. Thinks: "Automation can improve efficiency."

4. Feels: Overwhelmed by repetitive tasks.

**Chapter 3**

# REQUIREMENT ANALYSIS

## 3.1 Customer Journey Map

1. Awareness: Discover solutions to inefficiencies in SDLC.

2. Consideration: Evaluate SmartSDLC’s value proposition.

3. Onboarding: Quickly start using the platform.

4. Adoption: Use SmartSDLC regularly for development.

5. Value Realization: See measurable improvement in speed, quality.

6. Expansion: Explore more features and upgrade plan.

## 3.2 Solution Requirement

1.Prroject Overview: SmartSDLC\* is a next-generation software development lifecycle platform enhanced with Artificial Intelligence (AI).

2. Business Goals: Accelerate delivery through intelligent planning and task automation,

Improve code quality with AI-powered code analysis and reviews,

Reduce bugs and post-release issues using smart test generation and predictive monitoring,

Enhance team collaboration and visibility using AI-driven dashboards and feedback loops.

3. Functional Requirements: Planning, Development, Code Review and Testing.

## 3.3 Data Flow Diagram

1. User Inputs Story → NLP → Task Breakdown → Jira Update

2. Developer Pushes Code → Code Review AI → Suggest Changes

3. Build Triggered → Risk Analysis AI → Approve or Flag

4. Monitoring Alerts → Anomaly Detector → Notification to Slack

5. Sprint Ends → Analytics Engine → Dashboard/Report Generation

**Chapter 4**

# TECHNOLOGY STACK & PROJECT DESIGN

## 4.1 Technology Stack

1. Frontend: React or Angular for UI.

2. Backend: Python with AI/ML libraries (e.g., TensorFlow).

3. Database: Relational (e.g., MySQL) or NoSQL (e.g., MongoDB).

4. Integration: APIs using REST or GraphQL.

## 4.2 Problem Solution Fit

Smart SDLC solves manual, time-consuming development processes with AI-powered automation, improving efficiency and quality. It enhances collaboration and communication among team members, reducing errors and misunderstandings. With analytics and insights, teams make data-driven decisions. Smart SDLC delivers high-quality software products faster and more efficiently.

## 4.3 Proposed Solution

Smart SDLC proposes an AI-powered platform automating software development lifecycles. It streamlines requirements gathering, design, implementation, testing, and deployment. Real-time collaboration tools enhance teamwork and reduce errors. Analytics and insights enable data-driven decision-making.

## 4.4 Solution Architecture

1. Frontend: User interface for project management, collaboration, and monitoring.

2. Backend: AI-powered engine for automating software development tasks, such as requirements analysis, design, implementation, testing, and deployment.

3. Data Layer: Storage for project data, analytics, and insights.

4. Integration Layer: APIs and interfaces for integrating with other development tools and platforms.

**Chapter 5**

# PROJECT PLANNING & SCHEDULING

1. Requirements Gathering (10 days): Define scope and gather requirements.

2. Planning (10 days): Create schedule and assign tasks.

3. Development (40 days): Develop and test software.

4. Deployment (10 days): Plan and execute deployment.

5. Monitoring and Control (ongoing): Track progress and mitigate risks.

**Chapter 6**

**FUNCTIONAL AND PERFORMANCE TESTING**

# 6.1 Performance Testing

 Validate **speed** (response time)

 Ensure **scalability** (handles increased loads)

# 6.2 Results

* **Response Time (avg):** 800 ms → Acceptable if SLA is under 1 sec
* **Error Rate:** 0.5% → Might be acceptable, but should investigate causes
* **Throughput:** 150 requests/sec → Good for mid-sized app
* **CPU Usage:** 85% peak → Close to capacity, may require optimization or scaling
* **Memory Usage:** Constant increase → Possible memory leak (check endurance test)

## 6.2.1 Output Screenshots

[Insert placeholder for screenshots here]

**7.ADVANTAGES &**

**DISADVANTAGES**

* **Advantages**.
* Uses tools and automation for testing, integration, deployment, and monitoring, reducing manual errors.

**Disadvantages:**

* Requires setting up **dedicated test environments**, servers, and monitoring tools.

**8.CONCLUSION**

The **Smart SDLC** is a modern evolution of traditional software development life cycles, integrating automation, agile practices, and DevOps principles to increase **efficiency**, **speed**, and **collaboration**. It is highly effective for complex and scalable projects that require continuous delivery and adaptability. However, it also introduces **complexity**, **tool dependency**, and a **learning curve**, making it potentially unsuitable for smaller or simpler projects.

**9.FUTURE SCOPE**

1. Greater Integration of AI & Machine Learning

2. Advanced DevSecOps Adoption

**10.APPENDIX**

# 10.1 Source Code (if any)

GitHub Repositor[y](https://github.com/yourusername/citizen-ai)

# 10.2 Dataset Link

Dataset Sourc[e](https://example.com/dataset)

# 10.3 GitHub & Project Demo Link

Project Dem[o](https://example.com/demo)