

Ideation Phase

Define the Problem Statements

Date	24 June 2025
Team ID	LTVIP2025TMID38248
Project Name	SmartSDLC - AI-Enhanced Software Development Lifecycle
Maximum Marks	2 Marks

Customer Problem Statements:

Here are some filled-out problem statements focusing on the pain points of software development professionals:

Problem Statement(PS)	I am (Customer)	I'm trying to	But	Because	Which makes me feel
(PS-1): Requirement Ambiguity	A Project Manager leading a software development team, struggling with project timelines.	Ensure all software requirements are clear, well-defined, and correctly understood by the development team.	Unstructured and lengthy requirement documents often lead to ambiguity, misinterpretations, and overlooked details, causing delays and quality issues.	There's no efficient way to automatically classify and structure requirements from various documentation formats, making manual analysis prone to human error and inconsistency.	Frustrated and anxious, knowing that foundational issues in requirements will propagate throughout the entire development process, impacting delivery.
(PS-2): Inefficient Code Development & Debugging	A Software Developer working on complex features, constantly under pressure to deliver high-quality code quickly.	Rapidly write functional, bug-free code and efficiently resolve any issues that arise.	Manual code generation is time-consuming for boilerplate or repetitive tasks, and debugging complex errors or understanding unfamiliar code takes significant effort.	There's a lack of intelligent tools that can automate code generation and provide instant, accurate bug fixes or explanations for existing code.	Overwhelmed and inefficient, leading to burnout and a perceived slowdown in productivity.
(PS-3): Manual & Incomplete Testing	A QA Engineer responsible for ensuring software quality, facing tight release schedules.	Create comprehensive and effective test cases to cover all functionalities and edge cases, and ensure minimal defects reach production.	Manually writing test cases for every feature is labor-intensive and often leads to incomplete coverage, especially as software complexity grows.	There's no automated system that can intelligently analyze requirements or code to suggest and generate diverse and test scenarios.	Pressured and concerned, knowing that potential bugs might slip through, risking user dissatisfaction and reputational damage.

(PS-4): Difficulty Understanding Complex/ Legacy Code	A New Team Member or a developer tasked with maintaining a large, unfamiliar, or legacy codebase.	Quickly grasp the functionality and structure of complex code snippets or entire modules without spending excessive time on manual analysis.	The absence of comprehensive documentation, intricate logic, or the sheer volume of code makes it extremely challenging and time-consuming to understand its purpose and how it works.	There's no automated tool that can provide concise, human-readable summaries or explanations of code, forcing reliance on laborious manual tracing and asking experienced colleagues.	Frustrated and delayed, hindering my ability to contribute quickly and effectively to the project.
(PS-5): Lack of Immediate SDLC Knowledge/ Support	A Junior Developer or a team member encountering specific SDLC-related questions or best practice inquiries.	Get instant, reliable answers and guidance on various Software Development Lifecycle topics, methodologies, tools, or process-related queries.	Finding quick and accurate information often requires interrupting colleagues, extensive searching through documentation, or waiting for team meetings, which can cause delays.	There's no readily available, intelligent, and interactive assistant that can provide on-demand, context-aware information and support regarding SDLC principles.	Stuck and inefficient, leading to delays in decision-making and potential sub-optimal approaches.