**Problem–Solution Fit – SmartSDLC**

**Date**: 25 JUNE 2025  
**Team ID**: LTVIP2025TMID33777  
**Project Name**: SmartSDLC  
**Maximum Marks**: 5

**CUSTOMER SEGMENT(S)**

* Undergraduate and postgraduate students working on software engineering projects.
* Faculty mentors evaluating project documentation across SDLC phases.
* Early-stage developers needing structured guidance to organize project artifacts.
* Technical teams preparing SDLC-aligned documentation for product development or internal evaluations.
* Institutions seeking tools to automate SDLC evaluation and standardization.

**JOBS-TO-BE-DONE / PROBLEMS**

* Classify documents like Requirements, Design, and Test Plans into correct SDLC phases.
* Automatically identify if a document is aligned with SDLC structure (Requirement, Design, Implementation, Testing, Maintenance).
* Quickly validate whether a student's documentation meets academic or industrial expectations.
* Save time during evaluation and avoid manual checking errors.
* Receive auto-generated summaries or feedback on document quality and SDLC completeness.

**TRIGGERS**

* Students struggling to correctly categorize or format project documentation.
* Mentors or evaluators having to manually go through multiple unstructured reports.
* Confusion about how to break down project work into SDLC phases.
* Projects requiring SDLC traceability, documentation audits, or academic grading.

**EMOTIONS (BEFORE / AFTER)**

* **Before**: Confused, uncertain about document structure, overwhelmed by SDLC terminology, stressed during evaluations.
* **After**: Confident in documentation quality, reassured by AI-verified classification, clear on project structure, efficient in both writing and evaluation.

**AVAILABLE SOLUTIONS**

* **Manual classification by student**
  + *Pros*: Low-cost, allows full control.
  + *Cons*: Error-prone, time-consuming, inconsistent between students or teams.
* **Templates or static SDLC checklists**
  + *Pros*: Gives structure.
  + *Cons*: Not intelligent, doesn’t assess document content or provide feedback.
* **Mentor-based manual review**
  + *Pros*: Expert feedback.
  + *Cons*: Time-intensive, subjective, impractical for large batches.
* **Online SDLC tutorials**
  + *Pros*: Good for concept learning.
  + *Cons*: Not interactive, doesn’t process real documents or give project-specific guidance.

**CUSTOMER CONSTRAINTS**

* Limited exposure to structured SDLC documentation.
* Time pressure to complete reports before submission.
* Unclear guidelines from institutions on SDLC phase expectations.
* Need for a tool that runs locally or in the cloud without complex setup.
* Must ensure academic integrity and prevent plagiarism detection errors.

**BEHAVIOUR**

* Students often mix up SDLC phases in reports.
* Relying on past project reports for structure instead of understanding principles.
* Mentors manually reading and interpreting each PDF/Doc file.
* Teams googling SDLC terms or seeking sample formats without understanding context.
* Limited iteration or peer review of documentation.

**CHANNELS of BEHAVIOUR**

**ONLINE:**

* Searching for SDLC report formats or examples.
* Using ChatGPT or other tools for document generation.
* Submitting documentation via portals like Moodle, Google Forms, or GitHub.
* Collaborating via Google Docs, Notion, or MS Word.

**OFFLINE:**

* Peer feedback and review.
* Mentor evaluations during viva or submission.
* Document printing for physical review.
* Whiteboard or handwritten planning of SDLC stages.

**PROBLEM ROOT CAUSE**

* Lack of intelligent systems to classify or evaluate SDLC-aligned documentation.
* SDLC training often focuses on theory, not practical implementation.
* No feedback loop or validation tool to tell students if they followed SDLC properly.
* Current review processes are manual, subjective, and don’t scale well in academic or early development settings.

**YOUR SOLUTION: SmartSDLC**

* **SmartSDLC** is an AI-assisted platform that auto-classifies uploaded documents into appropriate SDLC phases using NLP-based zero-shot classification.
* Leverages **Hugging Face transformers** for document understanding and phase labelling.
* Converts messy or mixed-phase PDFs into structured SDLC outputs.
* Provides summaries, confidence scores, and suggestions for better alignment.
* **Built using Streamlit**, allowing a user-friendly, browser-based interface.
* Simulates IBM Granite/Watson-like behaviour through current integration with the **Google Gemini API**.
* Supports **local deployment** for individual student use or **Streamlit Cloud deployment** for institution-wide access.
* Future enhancements include batch uploads, plagiarism checking, downloadable reports, and mentor evaluation modes.