

SmartSDLC – AI-Enhanced Software Development Lifecycle

Category: Generative AI with IBM

Skills Required: IBM Cloud, IBM Watsonx, FastAPI, LangChain, Streamlit, PyMuPDF

Prepared by:

SIREEN S

FATHIMA SAFA B S

ASWIN MEHARAJ BANU A

S NASIDHA BEGAM

Date: 06-09-2025

Abstract

SmartSDLC is an AI-powered platform designed to automate and streamline the Software Development Lifecycle (SDLC) using IBM Watsonx's generative AI capabilities. By integrating advanced tools like FastAPI, LangChain, and Streamlit, the system reduces manual effort in requirement analysis, code generation, testing, debugging, and documentation. The platform enhances efficiency, traceability, and software quality.

Introduction

Software Development Lifecycle (SDLC) is the process of planning, creating, testing, and deploying an information system. Traditional SDLC methods often involve extensive manual work, which is time-consuming and prone to human error. SmartSDLC leverages artificial intelligence (AI) to automate multiple stages of SDLC, making the process faster, more accurate, and more efficient.

Objectives

- Automate requirement gathering and classification- Enable AI-powered code generation- Provide automated bug fixing support- Enhance documentation and traceability- Improve overall SDLC efficiency

Literature Review / Related Work

Traditional SDLC methods require manual effort at every stage, from requirement analysis to maintenance. Recent advancements such as GitHub Copilot and ChatGPT show the potential of AI in code generation. IBM Watsonx brings enterprise-grade AI capabilities to SDLC automation. SmartSDLC bridges the gap by combining requirement classification, code generation, and bug fixing in one integrated system.

System Architecture

The architecture of SmartSDLC consists of a user-friendly frontend (Streamlit), a secure backend (FastAPI), and IBM Watsonx for AI-driven tasks. The workflow integrates PDF parsing (PyMuPDF), AI classification, code generation, and debugging services.

Technology Stack

- AI/ML: IBM Watsonx Granite-20B, LangChain- Backend: FastAPI, Uvicorn- Frontend: Streamlit- PDF Processing: PyMuPDF- Integration: REST APIs, Swagger UI- Deployment: IBM Cloud, Local hosting

Module Description

1. Requirement Upload & Classification: Upload PDF requirements → Extract text → Classify into SDLC phases → Structured user stories.
2. AI Code Generator: Input natural language prompt/user story → Generate optimized code using Watsonx → Display with syntax highlighting.
3. Bug Fixer: Input buggy code → AI detects syntax/logic errors → Provides corrected code for comparison.
4. User Interface: Streamlit dashboard to manage uploads, outputs, and chatbot assistance.
5. Backend & Security: FastAPI endpoints, authentication, Swagger UI documentation.

Use Case Scenarios

Scenario 1: Requirement Upload & Classification – Extracts and organizes requirements into SDLC phases.

Scenario 2: AI Code Generator – Converts natural language/user stories into production-ready code.

Scenario 3: Bug Fixer – Identifies and fixes errors in code snippets.

Advantages & Benefits

- Reduced manual workload- Faster development- Improved accuracy- Enhanced traceability- Scalable and modular system

Limitations

- Dependence on AI model accuracy- Requires internet/cloud for IBM Watsonx API calls- Limited programming language support initially

Future Enhancements

- Extend bug fixing to more languages- Integrate CI/CD pipelines- AI-driven test case generation & execution- Cloud-based multi-user collaboration

Conclusion

SmartSDLC demonstrates the power of generative AI in transforming the software development process, making it faster, more reliable, and efficient. By integrating AI across multiple SDLC stages, the system delivers significant productivity improvements and software quality enhancements.

Output

AI Code Analysis & Generator

Code Analysis

Code Generation

Upload PDF

Drop File Here

< OR >

Click to Upload

Or write requirements here

Describe your software requirements...

Analyze

Requirements Analysis

Built with Gradio · Settings

AI Code Analysis & Generator

Code AnalysisCode Generation

Code Requirements

Describe what code you want to generate...

Programming Language

Python

Generate Code

Generated Code