SDLC AI – Comprehensive Documentation
Team ID: NM2025TMID10099
Team Size : 4
Team Leader : SENTHAMIL SELVAN.J
Team member :ABDUL BASITH.I
Team member :SANJAI.S
Team member :GANESAN.V
SDLC AI is an intelligent system designed to automate, optimize, and enhance various phases of the Software Development Lifecycle (SDLC). It integrates Artificial Intelligence and Machine Learning capabilities to reduce manual efforts, ensure consistency, and accelerate software delivery processes from requirements gathering to deployment and maintenance.
This documentation outlines the structure, capabilities, and architecture of SDLC AI for developers, architects, stakeholders, and maintainers.
☐ System Overview
SDLC AI is built as a modular AI-driven framework that can either operate independently or integrate into existing DevOps pipelines. It assists teams by:
Generating requirements from client inputs
Suggesting design patterns
Automating code generation
Executing tests
Predicting deployment issues

Assisting with maintenance through anomaly detection and log analysis

The system can be deployed on-premises or in the cloud and is accessible via a web interface, CLI, or RESTful API.



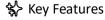
Minimize human errors in development

Automate repetitive tasks

Improve collaboration and traceability

Speed up time-to-market

Provide actionable insights throughout SDLC



SDLC Phase Feature Description

Requirement Analysis Natural Language Processing (NLP) to extract features, use cases, and technical specs from client documentation.

Design Auto-generates UML diagrams and design templates using Al recommendation engine.

Development Al-assisted code generation based on user stories or requirement definitions. Supports multiple languages.

Testing Auto-generates unit, integration, and regression tests using code analysis. Includes AI-based bug detection.

Deployment Integrates with CI/CD pipelines to suggest optimal deployment strategies and rollback plans.

Maintenance Predicts failure points, monitors logs, and suggests patches or improvements using anomaly detection.

Documentation Auto-generates technical documentation and API references based on code and commits.

☼ Module-wise Breakdown
1. NLP Engine
Parses client documents, chat logs, and emails
Identifies functional and non-functional requirements
2. Design Generator
Suggests software architecture based on system needs
Auto-generates UML class, activity, and sequence diagrams
3. Al Code Generator
Converts structured requirements into boilerplate or production-ready code
Supports Python, JavaScript, Java, C#, etc.
Adapts to custom architecture constraints
4. Al Test Suite
Creates test cases based on logic flow and historical bugs
Uses mutation testing and fuzzing
5. Deployment Assistant
Integrates with Docker, Kubernetes, Jenkins, GitHub Actions

Suggests blue-green, canary, or rolling deployment based on app type
6. Maintenance and Monitoring
Al-driven log analyzers
Root cause analysis
Ticket prediction and automated issue tagging
© Workflow
Input: Raw requirements (text, PDFs, voice)
Requirement Parsing: NLP extracts user stories
Design: AI recommends architecture
Development: Generates and optimizes code
Testing: Auto-generates and runs test cases
Deployment: Connects to CI/CD and deploys
Maintenance: Monitors system and recommends fixes
☐ Technologies Used
Category Tools / Frameworks
Programming Python, JavaScript
AI/ML OpenAI APIs, spaCy, scikit-learn, LangChain

Testing Selenium, PyTest, JUnit

Deployment Docker, Kubernetes, Jenkins, GitHub CI/CD

Monitoring ELK Stack, Prometheus, Grafana

Documentation Sphinx, Swagger, MkDocs

Backend Frameworks FastAPI, Node.js

Database PostgreSQL, MongoDB

Integrations

JIRA: For issue tracking and user story ingestion

GitHub/GitLab: For code versioning and trigger pipelines

Slack/MS Teams: Notifications and AI chatbot assistance

VS Code Plugin: Developer assistance inside the IDE

CI/CD Tools: Jenkins, GitHub Actions, CircleCI

Security Considerations

Role-based access control (RBAC)

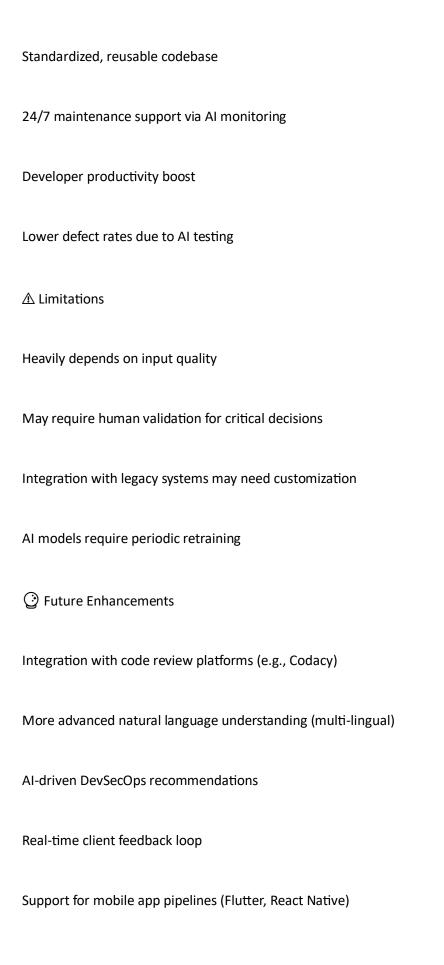
Secure storage of credentials using Vault

Data encryption at rest and in transit

LLM safety mechanisms to prevent insecure code generation

✓ Benefits

Up to 60% reduction in development cycle time



SDLC AI represents a significant shift in how modern software is built. By bringing AI into each phase of the development lifecycle, teams can produce high-quality software faster, with fewer bugs and better alignment to business goals. While not a full replacement for human expertise, SDLC AI acts as a powerful assistant to developers, architects, and product managers.