



AI-POWERED SDLC PLAYBOOK

A practitioner's framework for embedding AI across every phase of enterprise software delivery

AIEnterprise Inc. | Consult. Design. Build. Modernize. Operate.

www.aienterprise.com

Why an AI-Powered SDLC?

THE PROBLEM

~35% of delivery effort in traditional enterprise projects goes to mechanical, non-value-adding work: boilerplate code, documentation, test authoring, compliance artifacts, and format translation.

This inflates team sizes, extends timelines, and burns budget on work that doesn't directly produce business value.

THE APPROACH

Embed AI into every SDLC phase — not as a novelty, but as infrastructure. Senior practitioners direct AI to handle mechanical work while they focus on decisions, integration, and business value.

Result: ~50% smaller teams, ~50% faster timelines, zero compromise on quality or compliance.

Seven Phases, AI Everywhere

Each phase of the SDLC maps to specific AI tools and practices. This playbook covers all seven.

01



**Requirements
& Planning**

02



**Architecture
& Design**

03



**UX Design
& Prototyping**

04



**Development
& Code Review**

05



**Testing &
Quality**

06



**Compliance &
Documentation**

07



**SDLC Analytics
& Optimization**

AI-Augmented Requirements

BEFORE AI

- 2–3 BAs writing BRDs for 6–8 weeks
- Manual translation from BRD to Jira stories
- Inconsistent acceptance criteria
- Hours of grooming to fill documentation gaps

WITH AI

- 1 Senior BA + Claude in under 3 weeks
- AI generates structured stories from transcripts
- Consistent AC, edge cases, NFR annotations
- Direct push to Jira + linked Confluence pages

Process

1. Feed stakeholder transcripts and legacy docs into Claude
2. Claude generates user stories with AC, edge cases, NFRs
3. BA reviews, refines, and approves
4. Approved stories auto-pushed to Jira with labels, components, linked Confluence context

TOOLS

- Claude (story generation)
- Jira (story management)
- Confluence (context docs)
- Slack (async collaboration)

Human Decisions, AI Documentation

Architects focus on trade-off decisions. Claude handles the documentation around those decisions.

ARCHITECT (Human Decisions)

- Service boundary definitions
- CQRS domain boundaries
- Event schema evolution strategy
- Technology trade-off decisions
- Performance and security architecture
- Review and approval of all AI output

CLAUDE (AI-Generated Artifacts)

- Technical Design Documents (TDDs)
- OpenAPI/Swagger specifications
- Kafka topic schemas
- Integration Specification Docs (ISDs)
- Sequence and context diagrams
- Architecture Decision Records (ADRs)

Architecture phase: 4 weeks → 2 weeks | Documentation stays current because regeneration is trivial

Design in Code, Not in Mockups

Lovable enables rapid UX prototyping by translating design concepts directly into working React components. Stakeholders review functional prototypes instead of static mockups, and WCAG accessibility is built in from day one.

Rapid Prototyping

Design concepts become working React components in hours, not days. Iterate in code with real stakeholder feedback.

Design System First

Establish component library and design tokens early. AI-generated components inherit the system automatically.

Accessibility Built In

WCAG compliance is a design-time concern, not a retrofit. Lovable generates accessible markup by default.

Tools: Lovable • React / Next.js / TypeScript • Figma (reference) • Storybook

AI-Assisted Development at Scale

GITHUB COPILOT (In-IDE)

- Spring Boot service scaffolding
- REST / GraphQL endpoint definitions
- Kafka producer/consumer configs
- JUnit + Mockito test stubs
- Dockerfile / K8s manifests
- Database migration scripts

CLAUDE (Complex Generation)

- Event-driven workflow design
- CQRS read-model projections
- Complex integration logic
- Pre-review code analysis (null safety, transactions, OWASP)
- Architecture pattern implementation

Developer productivity: 30–40% gains on boilerplate-heavy tasks | Senior engineers focus on business logic

From Test Authors to Test Curators

AI generates test suites from user stories. QA engineers shift from writing tests to reviewing, curating, and running exploratory testing.

Functional Tests

Playwright & Selenium scripts generated from acceptance criteria. Happy path, alternate flows, edge cases.

API Tests

Python-based test utilities for REST & GraphQL endpoint validation, data setup, and integration verification.

Performance Tests

Gatling scenarios generated from production traffic patterns. Load, stress, and soak testing from Sprint 1.

Accessibility Tests

WCAG compliance integrated into CI/CD pipeline. Automated axe-core scans on every build.

Test automation coverage established 60–70% faster | QA focuses on exploratory testing

Compliance as a Byproduct

SOX compliance artifacts are generated continuously from the development process — no dedicated compliance coordinator required.

AUTO-GENERATED ARTIFACTS

- Change management records (Jira → GitHub → Jenkins → Deploy)
- Requirement-to-code-to-test traceability matrices
- Segregation of duties documentation
- Audit trails across the full pipeline
- API documentation (auto-synced from OpenAPI specs)
- Operational runbooks (Claude-drafted, DevOps-reviewed)

TRACEABILITY CHAIN



Real-Time SDLC Intelligence

An AI analytics layer monitors delivery health in real time — surfacing bottlenecks before retrospectives.

Velocity Trends

Sprint-on-Sprint

Defect Density

Per Service

Code Quality

SonarCloud

Build Stability

Pipeline

Deploy Frequency

Per Week

PROACTIVE INSIGHTS

- Services with rising defect density flagged automatically
- Flaky test suites degrading pipeline reliability identified
- Requirements volatility trends surfaced in real time
- Data-driven sprint adjustments, not gut-feel retrospectives

The AI-Augmented Team

~50% smaller than traditional teams. Every member is senior. AI amplifies expertise — it doesn't replace it.

Role	Count	AI Force Multiplier
Delivery Lead / Scrum Master	1	SDLC analytics dashboard, automated reporting, AI-assisted status reporting, risk analysis
Senior Architect	1	Claude for TDDs, API specs, ADRs, diagrams
Senior Business Analyst	1	Claude for story generation, Jira/Confluence push
Backend Developers (Java/Spring)	4	Copilot + Claude for scaffolding, code review
Frontend Developers (React/Next.js)	2	Copilot + Lovable for component generation
QA Engineers	2	AI-generated test suites (Playwright, Selenium, Python)
DevOps Engineer	0.5	Claude for IaC, runbooks, pipeline configs
UX Designer	0.5	Lovable for rapid prototyping, design system

Total: ~12 people | Traditional equivalent: ~30 people

The Stack Behind the Playbook

Backend

Java, Spring Boot
REST, GraphQL, Swagger
Kafka, Event Streaming, Pub/Sub

Frontend

React, Next.js, TypeScript
SPA + SSR
WCAG-compliant design system

Data

PostgreSQL (transactional)
MongoDB (catalog flexibility)
Redis (caching, sessions)

Cloud & DevOps

GCP, Docker, Kubernetes
Jenkins CI/CD
Terraform IaC

AI Tools

Claude (architecture, docs, tests)
GitHub Copilot (inline code)
Lovable (UX prototyping)

Quality & Ops

JUnit, Mockito
Selenium, Playwright, Python
SonarCloud, Jira, Confluence

The Impact: Before vs. After AI

Metric	Before AI	With AI	Change
Team Size	~30 people	~12 people	–55%
Requirements Phase	6–8 weeks	<3 weeks	–50%+
Architecture & Design	4 weeks	2 weeks	–50%
Developer Productivity	Baseline	+30–40%	↑ Gains
Test Automation Speed	Baseline	60–70% faster	↑ Speed
Compliance Overhead	Dedicated FTE	Automated	~0 effort

Five Principles of the AI-Powered SDLC

01

AI as Infrastructure, Not Novelty

AI isn't a demo or an experiment. It's built into every workflow, every sprint, every phase.

02

Senior Talent + AI > Large Teams

AI amplifies expertise. Junior devs don't become productive with Copilot the way senior engineers do.

03

Decisions by Humans, Documentation by AI

Architects make trade-offs. AI handles the mechanical work of capturing and maintaining those decisions.

04

Compliance as a Byproduct

Audit artifacts are generated continuously from the development process — not bolted on as a separate workstream.

05

Measure Everything, Improve Continuously

SDLC analytics provide real-time visibility. Course-correct mid-sprint, not after the fact.

Getting Started: Adoption Roadmap

Week 1–2

Foundation

- License AI tools (Copilot, Claude)
- Establish prompt libraries for your stack
- Define AI-use guidelines and review gates

Week 3–4

Pilot Sprint

- Run one sprint with AI-augmented workflow
- Measure velocity, quality, documentation output
- Refine prompts and integration points

Week 5–8

Scale

- Roll out across all SDLC phases
- Integrate SDLC analytics layer
- Establish compliance automation

Ongoing

Optimize

- Track AI ROI metrics sprint-on-sprint
- Update prompt libraries as patterns emerge
- Share learnings across teams



Ready to Transform Your Delivery Model?

This playbook is a living framework. Adapt it to your stack, your team, and your compliance requirements. Or let us help you implement it.

www.aienterprise.com

sales@aienterprise.com | +1 833-234-0247

AIEnterprise Inc. | Consult. Design. Build. Modernize. Operate.