SPAD

Does your company build software with AI with a method, or just faster?

> *"AI generates plausible code at great speed. Without a method, that speed turns into design decisions nobody took, insufficient tests and vulnerabilities nobody reviewed."*

WHAT IT IS

An engineering method

SPAD solves a specific problem: building software with AI in a governed way. Work is sequential, blocking and auditable: eleven phases, from context to release management; none starts without reviewing the previous one and each leaves an artefact. It connects with the company's AI governance at a single point, SEVEN-G document 53.

Implication: Can your company show, for its latest AI-built development, the design approved before any code was written?

Risk: Without a method, code works in the demo and fails in production, and nobody can explain why it is built that way.

WHAT IT PROPOSES

Separate roles and verdicts

The AI that reviews is never the one that builds, and a human orchestrator validates every output. Each review ends in GO, GO with changes or NO-GO; with NO-GO, work goes back. An AI response that breaks its phase rules is discarded in full, even if it looks good: following the flow matters more than apparent quality.

Implication: Who reviews the code that AI generates in your company today, and are they independent of whoever wrote it?

Risk: If the same AI writes and reviews, design flaws and vulnerabilities go through without anyone seeing them.

FOR THE COMPANY

Build with a method and demand as a client

SPAD serves to build the company's own software with AI and to require suppliers and integrators to pass the productivity gain on to the company: lower cost or more scope for the same price. It is enough to ask for each phase's artefact, the model and human validation records, the verdicts and the process metrics.

Implication: What share of the savings AI gives your suppliers reaches your invoice or the scope of your projects today?

Risk: Without process evidence, the company cannot know whether the AI gain reaches its invoice or stays with the supplier.

FOUR QUESTIONS FOR MANAGEMENT

1. **How much of our software is already built with AI, and with what?**
   A model record for each artefact: provider, model, version, instruction and contexts used.
2. **Is the design decided before code is written?**
   An explicit plan reviewed by an independent AI and validated by a person before building.
3. **What do we ask of our suppliers when they build with AI?**
   Artefacts per phase, human validation and violation records, the verdict of each review and process metrics.
4. **Does the method work in our company?**
   Our own metrics: NO-GO rate, iterations to GO, time per phase, escaped defects and rework after delivery, with no promised figures.

WHAT FAILS WHEN BUILDING WITHOUT A METHOD

WHAT FAILS	LEVEL	WHAT TO DO
Code that works in the demo and fails in production	HIGH	Require a reviewed plan before code is written
The same AI writes and reviews the code	HIGH	Separate the reviewing AI from the building AI
Vulnerabilities in code that nobody reviewed	HIGH	Put sensitive code through a security review
Tests written at the end, if time allows	MEDIUM	Set the test strategy before building
No evidence of development when audited	MEDIUM	Keep each phase's artefact with its version
The AI gain stays with the supplier	MEDIUM	Ask the supplier for the process metrics
Full cycle applied to throwaway prototypes	LOW	Keep the method for what aims at production

FURTHER READING

What SPAD is and what it is for
Document 00 · method and what to ask for as a client

Operating guide
Document 01 · phases, roles and verdicts

Self-assessment and metrics
Document 08 · process metrics

OWASP Top 10:2025, web application security risks
OWASP Foundation · 2025

Fernando García Varela · CC BY 4.0 · Version 0.x, under review

Generic help offered free of charge, with no legal review for any specific case: each organisation adapts it, identifies the regulation that applies to it and is responsible for its own compliance. It is neither advice nor certification. To the maximum extent permitted by law, the author accepts no liability for the effects of its application.