Self-Diagnostic: Five Signs Your AI Coding Adoption Needs Structure

Before diving into the framework, take two minutes to assess where your organization stands. If three or more of these signs are present, you are likely accumulating hidden risk behind a productivity dashboard that looks healthy.

#	SIGNAL	DIAGNOSTIC QUESTION
1	Adoption is broad but shallow. Developers use AI for autocomplete and boilerplate. No team has changed how it specifies, reviews, or delivers software.	Can you name one team that has fundamentally changed its delivery process because of AI rather than simply speeding up existing processes?
2	Every team uses AI differently. No shared standards for context, prompting, quality review, or governance. You get faster in silos but not smarter as a whole.	If a developer moves from Team A to Team B, do they encounter a completely different AI-assisted workflow?
3	Code quality is flat or declining despite velocity gains. Sprint output is up, but defect escape rates, code churn, and review cycle times are not improving at the same rate.	Are your defect escape rates and time-to-resolve metrics improving at the same rate as your velocity metrics?
4	You cannot trace a deployed feature back to a business decision. Answering "why was this built this way?" requires archaeology across Jira, Confluence, Slack, and meeting notes.	Pick any feature deployed last quarter. Can you produce the business rationale, key trade-offs, and decision-makers in under five minutes?
5	Security findings are outpacing your AppSec team. AI-generated code produces a volume of findings that exceeds review capacity. The backlog grows. Risk accumulates without visibility.	Has the ratio of new to resolved security findings worsened since you deployed AI coding tools?

SCORING

3+

SIGNALS PRESENT

You are accumulating hidden risk.

Read the SDLC white paper →

1-2

SIGNALS PRESENT

You are ahead of most, but the structural risks above will still compound as your AI adoption scales.