

The AI-Ready SDLC 12 Control Gates

A field guide for coding assistants and agents.

Industry evidence, operating controls, worksheets, and a 30-day rollout.

The candidate change got cheaper. Responsibility did not.

Coding assistants and agents can reduce the effort required to produce a plausible change. They do not remove the need to bound work, protect context, inspect dependencies, test behavior, assign a reviewer, document a release, or recover from failure.

12 CONTROL GATES From task framing through post-release learning.	8 REQUIRED ARTIFACTS Evidence a reviewer can inspect outside a private chat.	30 DAY PILOT One change type, paired runs, and an explicit scale decision.	4 OUTCOMES Speed, quality, control burden, and customer impact.
--	---	---	--

Leadership decision	Minimum evidence before yes
Enable a tool	Approved data classes, contract and retention review, identity controls, and an accountable owner.
Use it on a change type	Bounded task, acceptance evidence, context rule, named reviewer, and recovery path.
Expand the pilot	Comparable before-and-after outcomes with no material weakening of review, security, or stability.
Allow more autonomy	A smaller failure radius, reliable observability, tested containment, and explicit action limits.

Operating principle

A reviewer should be able to reconstruct why the change was accepted without reopening a private model conversation. If the system cannot produce that evidence, it is not ready for more autonomy.

Adoption is broad. Trust and control remain uneven.

The useful industry signal is not that developers are using AI. It is that use is widespread while verification, stability, and task fit remain unsettled.

84% USE OR PLAN TO USE Stack Overflow 2025 AI question; 33,662 responses.	46% DISTRUST ACCURACY Versus 33% who trust AI output accuracy.	66% ALMOST-RIGHT FRUSTRATION Most-cited AI frustration in the same survey.	30% LITTLE OR NO TRUST Developers in DORA's 2025 AI-assisted SDLC research.
--	---	---	--

Evidence	What it supports	What it does not prove
Stack Overflow Developer Survey 2025	Use is mainstream; experienced developers remain cautious; deployment and monitoring are resisted.	A representative census of all developers, audited tool use, code quality, or business return.
DORA State of AI-assisted Software Development 2025	AI behaves as an amplifier. Higher adoption can increase throughput and instability at the same time.	That any single tool or team will reproduce an aggregate association.
NIST SP 800-218 and 800-218A	AI-specific practices belong inside an established secure-development system.	A product certification, a vendor approval, or a substitute for organization-specific risk analysis.

Operator interpretation

Do not set policy from adoption percentages. Use them to justify disciplined evaluation. The control question is whether the team can preserve evidence and stability as generation becomes faster and more autonomous.

Survey caution: Stack Overflow recruited through its own channels and DORA surveyed technology professionals. Both are valuable directional evidence, not population estimates for every engineering organization.

A control overlay, not a parallel SDLC

This field guide translates established secure-development guidance and current industry research into a release workflow for teams that consume AI coding tools or introduce agents into delivery.

Source	Role in this guide	Boundary
NIST SP 800-218 (SSDF)	Base practices for preparing the organization, protecting software, producing well-secured software, and responding to vulnerabilities.	Technology-neutral; it does not prescribe this 12-gate sequence.
NIST SP 800-218A	AI-specific profile for producers, acquirers, and users of AI systems, used with SSDF 1.1.1.	Focused on generative AI and dual-use foundation-model development and acquisition.
OWASP GenAI guidance	Concrete LLM and agent failure modes, including excessive agency, prompt injection, insecure output handling, and supply-chain exposure.	Community guidance; apply it to the architecture and threat model in front of you.
DORA and Stack Overflow	Current evidence about adoption, trust, throughput, stability, and developer experience.	Observational and self-reported evidence, not a causal guarantee.

Who owns what

Role	Accountability
Engineering manager	Select the change type, protect review capacity, and own the expansion decision.
Developer	Declare material tool use, inspect the change, and supply repeatable evidence.
Reviewer	Understand the behavior, challenge the evidence, and retain authority to reject.
Platform / security	Define approved tools, data boundaries, identity, logging, and containment.
Product / service owner	Define acceptable behavior, user impact, monitoring, and rollback triggers.

The controls apply to inline completion, chat, repository-aware assistants, autonomous agents, and custom pipelines. The evidence burden rises with consequence and autonomy.

The twelve gates on one page

Use this page in planning and release review. A gate is complete when its evidence exists and an accountable person can challenge it.

Gate	Decision	Minimum evidence
01 Frame	Bound the behavior and consequence.	Ticket, owner, acceptance evidence, failure cost, stop condition.
02 Constrain	Control what may enter model context.	Context manifest, exclusions, data class, source locations.
03 Select	Match capability and contract to the job.	Approved tool, model class, access mode, vendor evidence.
04 Record	Leave provenance another reviewer can use.	Tool use, material sources, model class, human decision.
05 Inspect	Find new code and supply-chain risk.	Diff review, dependency and license results, generated-file list.
06 Test	Prove requirements and regression safety.	Repeatable unit, integration, regression, and property tests.
07 Attack	Exercise misuse, boundary, and authorization cases.	Threat cases, adversarial results, unresolved risk decision.
08 Review	Keep accountable human judgment real.	Named approver, review scope, overrides, rejected outputs.
09 Reproduce	Make the result independent of a private chat.	Build instructions, pinned inputs, test output, environment.
10 Release	Bind evidence, ownership, and recovery.	Change record, limits, monitoring signal, rollback action.
11 Observe	Measure system behavior after deployment.	Stability, security, cost, user, and quality signals.
12 Learn	Turn failures into changed controls.	Incident review, false-positive log, control and training update.

Tailoring rule: combine gates inside existing tickets, pull requests, CI checks, release records, and incident reviews. Do not create a second workflow whose only purpose is proving that AI was governed.

Frame and constrain

Decide what the model may see before asking what it can do.

01 Frame Minimum practice Name the behavior, owner, affected users, acceptance evidence, maximum failure, and stopping condition. Reviewer question Can two reviewers agree on what good looks like before generation begins? Stop signal The request is 'improve this service' and has no boundary or test.	02 Constrain Minimum practice List files, repositories, records, tools, and external sources allowed in context. Explicitly exclude secrets and unrelated data. Reviewer question Could the context be reconstructed and approved without seeing a private chat? Stop signal Repository-wide access is granted because narrowing context is inconvenient.
--	---

Implementation note

Implementation note: store a short context manifest with the ticket or change. For agentic work, include tool permissions, write targets, network access, and maximum run scope.

Required evidence before exit

Gate	Artifact
Frame	Ticket, owner, acceptance evidence, failure cost, stop condition.
Constrain	Context manifest, exclusions, data class, source locations.

Select and record

Use the least-capable approved tool and leave a usable trail.

03 Select Minimum practice Match the tool, model class, plan, region, retention, training terms, and identity controls to the data and consequence. Reviewer question Would this job still justify the tool if the newest model name were removed? Stop signal The team enables a broad agent because it already owns the seats.	04 Record Minimum practice Capture material use without logging every keystroke: tool, model class, context sources, generated surface, and human decision maker. Reviewer question Can an incident reviewer identify what changed and who accepted it? Stop signal The only provenance is a developer's inaccessible conversation history.
---	--

Implementation note

Implementation note: disclose material assistance in the existing change record. Provenance should support review and recovery, not become employee surveillance.

Required evidence before exit

Gate	Artifact
Select	Approved tool, model class, access mode, vendor evidence.
Record	Tool use, material sources, model class, human decision.

Inspect and test

Compilation is not acceptance.

05 Inspect Minimum practice Review new packages, versions, licenses, copied patterns, generated configuration, build scripts, and transitive dependencies. Reviewer question What entered the supply chain that was not in the ticket? Stop signal A familiar package name is accepted without checking publisher, version, or maintenance.	06 Test Minimum practice Tie deterministic tests to acceptance criteria. Add regression and property tests where broad generated changes increase the state space. Reviewer question Would these tests reject a plausible but wrong implementation? Stop signal The model writes both behavior and tests, and the human reviews only green status.
---	--

Implementation note

Implementation note: compare test mutation or fault-detection ability when raw coverage rises quickly. More generated tests can still leave the important assertion missing.

Required evidence before exit

Gate	Artifact
Inspect	Diff review, dependency and license results, generated-file list.
Test	Repeatable unit, integration, regression, and property tests.

Attack and review

Put the feature under pressure before a user does.

07 Attack Minimum practice Exercise authorization boundaries, malformed inputs, prompt or tool injection, sensitive-data paths, excessive actions, and recovery from partial execution. Reviewer question Which misuse case would create the worst believable outcome? Stop signal Adversarial testing is deferred because the feature passed its happy path.	08 Review Minimum practice Assign a reviewer with domain competence, enough time, required evidence, and authority to reduce scope or reject the change. Reviewer question Can the reviewer explain the changed behavior and rollback? Stop signal Review becomes a ritual approval because the diff is too large to understand.
---	--

Implementation note

Implementation note: cap generated change size. If a patch is too large for meaningful review, split the task or reduce autonomy before adding reviewers.

Required evidence before exit

Gate	Artifact
Attack	Threat cases, adversarial results, unresolved risk decision.
Review	Named approver, review scope, overrides, rejected outputs.

Reproduce and release

Turn a private generation into a public engineering artifact.

09 Reproduce Minimum practice Pin dependencies and relevant configuration. Preserve repeatable build and test instructions without requiring access to the generation session. Reviewer question Can a different engineer reproduce the evidence in a clean environment? Stop signal The prompt is the specification and the original developer is the build system.	10 Release Minimum practice Attach evidence, known limits, deployment owner, monitoring signal, containment step, and rollback trigger to the change record. Reviewer question What observable condition causes rollback, and who has authority to act? Stop signal The release note says only 'AI improvements' and no one owns post-release behavior.
---	---

Implementation note

Implementation note: release records should name the material behavior change, not merely the tool used to produce it. Tool disclosure never substitutes for a real change description.

Required evidence before exit

Gate	Artifact
Reproduce	Build instructions, pinned inputs, test output, environment.
Release	Change record, limits, monitoring signal, rollback action.

Observe and learn

The last gate is after deployment.

11 Observe Minimum practice Monitor delivery stability, security signals, quality, cost, review burden, user outcomes, and agent actions that cross system boundaries. Reviewer question Which signal tells us the change is worse even if deployment succeeded? Stop signal The team measures suggestions accepted but not retries, incidents, or customer impact.	12 Learn Minimum practice Feed incidents, near misses, overrides, false positives, context failures, and reviewer misses into controls, examples, and training. Reviewer question What will change in the system before the same failure recurs? Stop signal The postmortem blames hallucination without changing task design or evidence requirements.
---	---

Implementation note

Implementation note: review control effectiveness monthly during the pilot. Remove controls that do not change a decision; strengthen controls that repeatedly catch real failures.

Required evidence before exit

Gate	Artifact
Observe	Stability, security, cost, user, and quality signals.
Learn	Incident review, false-positive log, control and training update.

Autonomy should rise more slowly than capability

Treat autonomy as a property of the workflow, not a personality assigned to the model. Increase it only after the prior level produces stable evidence.

Level	Permitted action	Required controls	Example
0 - Advise	Suggest or explain; no repository or system writes.	Approved context, source checks, human creates the change.	Explain a failing test or propose cases.
1 - Draft	Create a bounded patch in an isolated branch.	Scoped files, no secrets, diff review, deterministic tests.	Draft a unit-tested parser change.
2 - Execute	Run approved tools and tests inside a sandbox.	Tool allowlist, egress limits, resource ceilings, complete action log.	Patch, build, and test a named component.
3 - Propose release	Open a change request and assemble evidence.	Protected branch, named reviewers, signed CI evidence, rollback plan.	Prepare a dependency update PR.
4 - Bounded production action	Take a reversible action within a narrow policy.	Independent authorization, live monitoring, kill switch, tested recovery.	Roll back a canary after a defined signal.

Permission design

Boundary	Question before granting it
Read scope	Which repositories, files, tickets, logs, and records are necessary for this task?
Write scope	Where can the agent create or change state, and how is that state isolated?
Tool scope	Which commands, APIs, package managers, and browsers are allowed?
Time / spend	What run duration, token spend, compute cost, and retry count ends execution?
Human checkpoint	Which irreversible or high-consequence transition requires fresh approval?
Containment	How is access revoked and partial work identified if the run behaves unexpectedly?

A 30-day pilot that can produce a decision

Use one change type and a small group. The purpose is to learn about the delivery system, not maximize seat count or generated lines.

Week	Work	Evidence at the gate	Decision
1 - Baseline	Choose one recurring change type. Sample current cycle time, review, rework, defects, and recovery.	Task definition, 10-20 comparable changes where available, data and tool boundaries.	Is the work stable and frequent enough to evaluate?
2 - Paired runs	Run comparable assisted and current-workflow changes. Log touch time, wait, review, corrections, and failures.	Context manifests, tool records, test output, reviewer notes.	Does assistance improve a bounded step without hiding work elsewhere?
3 - Stress	Run misuse, dependency, permission, rollback, and clean-environment reproduction exercises.	Failed cases, time to contain, reviewer disagreements, revised controls.	Can the team recover from the worst believable failure?
4 - Decide	Compare outcomes and control burden. Review with engineering, product, security, and service ownership.	One-page decision packet with scope, economics, limits, and stop rule.	Stop, continue unchanged, redesign, or expand one adjacent change type.

Pilot integrity rules

- Do not compare the best assisted change with an average unassisted week.
- Include review and correction time in both paths; faster generation can create slower verification.
- Freeze major tool and process changes during the measurement window when possible.
- Record severe failures separately; an average can hide an unacceptable security or customer event.
- Pre-write the conditions for stopping, continuing, and expanding before results are known.

Expansion is earned by a better delivery outcome without weaker evidence, stability, security, or recovery.

Measure the delivery system, not model activity

Acceptance rate and generated lines describe tool interaction. They are not business or delivery outcomes.

Dimension	Primary measure	Guardrail	Common trap
Flow	Lead time for the selected change type	Queue wait and handoff count	Counting only coding time
Review	Reviewer minutes per accepted change	Batch size and override rate	Treating approval as quality
Quality	Rework and escaped-defect rate	Severity-weighted failures	Using test coverage alone
Stability	Change-fail rate and recovery time	Rollback success and alert quality	Averaging away severe incidents
Security	Material findings and time to contain	Dependency, secret, and authorization checks	Counting scanner alerts as risk
Economics	Cost per accepted, reviewed change	Seats, usage, review, integration, incidents	Valuing every saved minute at salary
Human	Cognitive load and confidence	Review fatigue and skill development	Surveying only enthusiastic users
Customer	Task success or service outcome	Complaints, latency, accessibility, trust	Assuming delivery speed equals value

Decision packet

Claim	Observed value	Range / limitation	Owner
The selected step became faster			
Review burden did not erase the gain			
Quality and stability stayed within limit			
Data and vendor controls matched the work			
Recovery worked in the tabletop			

AI-assisted change evidence record

Attach this record to the existing ticket, pull request, or release packet. Keep answers short enough that reviewers will use them.

Field	Team entry	Minimum evidence
Task and owner		Bounded behavior, accountable owner, affected users
Acceptance		Tests or observable behavior that prove the requirement
Permitted context		Sources, exclusions, data class, repository scope
Tool and access		Approved tool, model class, identity, permissions
Generated surface		Files, configuration, dependencies, tests, documentation
Inspection		Dependency, license, secret, and copied-pattern results
Test and attack		Normal, regression, misuse, authorization, boundary cases
Reviewer		Named approver who can explain and reject the change
Release and rollback		Owner, monitoring signal, trigger, containment action
Post-release result		Delivery, quality, security, cost, and user measures

Release decision: ☐ Reject ☐ Rework ☐ Approve within current scope ☐ Approve and review for adjacent expansion

Reviewer: _____ Date: _____ Follow-up owner: _____

The almost-right dependency update

Use this 45-minute exercise with engineering, platform, AppSec, and service ownership. The purpose is to test the control system, not catch participants out.

Scenario

A repository agent opens a pull request to address a critical library vulnerability. The patch passes unit tests and updates an adjacent package to resolve a compatibility error. The new package name is legitimate, but the selected version was published from a newly compromised maintainer account. The agent also changes a deployment permission so its integration test can reach an external service. A reviewer sees green CI and approves the change. Twenty minutes after canary release, outbound traffic and authentication errors rise.

Minute	Inject	Team question
0	The agent PR is ready; CI is green; the security update is urgent.	Which gates and evidence are required before review begins?
10	The package version is new and the permission diff is outside the ticket.	Who can reduce scope or stop the change? What evidence is missing?
20	The canary emits unusual outbound traffic and authentication failures.	Which signal triggers containment? Who can revoke credentials and roll back?
30	Rollback succeeds, but the external credential remains active.	How is residual access found, revoked, and verified?
40	A similar agent configuration exists in three repositories.	Which system control changes before work resumes?

Debrief evidence

- Time from first signal to a named incident owner.
- Time to stop the agent, revoke access, and restore known-good behavior.
- Evidence that revealed the dependency and permission changes.
- Control changes applied across repositories, not only to the incident patch.
- Decision on whether the agent returns at the same autonomy level.

Choose the least complex safe mechanism

The highest-value decision may be to keep a task manual or use deterministic automation. Use probabilistic generation where ambiguity is useful and recovery is credible.

Work characteristic	Manual / deterministic	Assistant with review	Bounded agent
Acceptance rule	Exact and stable	Judgment required, but reviewable	Machine-checkable milestones plus human checkpoints
Failure consequence	High or difficult to reverse	Moderate and recoverable	Low within sandbox; high-impact transitions separately approved
Data	Restricted or tool terms do not fit	Approved data and context	Approved, minimized, logged, and permission-scoped
Change surface	Tiny deterministic transformation	Bounded diff a person can understand	Multiple steps within an explicit action envelope
Observability	Outcome already visible	Change and test evidence visible	Actions, state, cost, and stop signals visible in real time
Best first move	Script or checklist	Draft, explain, classify, or test	Repetitive multi-step work in an isolated environment

Do not use an agent when

- The task has no owner, acceptance evidence, or stable source of truth.
- The team cannot review the likely change size or reconstruct material actions.
- The required data, permissions, or external actions exceed approved boundaries.
- A short deterministic rule is safer, cheaper, and easier to audit.
- The organization cannot detect, contain, and recover from the worst believable failure.

AI belongs inside a controlled delivery system. It is not a substitute for one.

Primary guidance and current industry evidence

Edition 2.0 uses public primary guidance and source-published survey results. Percentages are dated and include sample context where the source provides it.

Source	Use	URL
NIST SP 800-218	Secure Software Development Framework (SSDF) Version 1.1	csrc.nist.gov/pubs/sp/800/218/final
NIST SP 800-218A	Secure Software Development Practices for Generative AI and Dual-Use Foundation Models	csrc.nist.gov/pubs/sp/800/218/a/final
OWASP GenAI Security Project	Top 10 for LLM Applications and Securing Agentic Applications Guide 1.0	genai.owasp.org
DORA 2025	State of AI-assisted Software Development; nearly 5,000 technology professionals	dora.dev/research/2025
DORA analysis	Balancing AI tensions: throughput, instability, trust, and the verification tax	dora.dev/insights
Stack Overflow 2025	Developer Survey AI results; 49,019 total survey respondents; per-question counts shown in source	survey.stackoverflow.co/2025/ai
AI Adoption Trends	AI in the SDLC training agenda and supporting exercises	aiadoptiontrends.com/training/ai-sdlc

Method notes

- The twelve gates are an AI Adoption Trends operational synthesis. They are not a NIST or OWASP certification scheme.
- Survey results describe respondents and question wording at the time measured. They should not be treated as audited deployment rates or causal performance estimates.
- Controls must be tailored to system consequence, data class, regulatory obligations, architecture, and recovery capability.
- This paper is educational guidance, not legal advice, a security assessment, or a substitute for professional review.

Field edition 2.0 - September 2026. Share freely with attribution. Send corrections and implementation notes to gary@aiadoptiontrends.com.