DOCUMENT 06 · INSTRUCTIONS BY PHASE

Instructions by phase

The reference text of the instruction that opens each SPAD phase: what is asked of the AI, what it is prohibited from doing and what it must return, so that any team executes the same phases in the same way

Fernando García Varela Version 0.1 19-09-2026 Under construction

11	7	1	3
instructions in the main cycle	instructions in the complementary cycles	common preamble	blocks per instruction

IN THIS DOCUMENT

01 How to use this document

02 Common preamble

03 Main cycle

04 Complementary cycles

05 Checklist for an instruction

06 Related documents

07 Version control

Version 0.x, under review. SPAD is currently at version 0.x. The content is operational and may be used, adapted and redistributed under the project licence (CC BY 4.0 for content and MIT for code), crediting the author and indicating changes. Documents and tools are still being reviewed: feedback and proposals for improvement are welcome.

Legal notice and disclaimer. SPAD is a reference methodology provided "as is" and for information purposes only. It does not constitute legal, regulatory or professional advice, does not guarantee results or compliance with any law or standard and is not a certification. **Each organisation that uses SPAD is solely responsible for validating its results, identifying the regulation that applies to it, verifying that it is current and certifying its own regulatory compliance**, with the appropriate qualified advice. This methodology is a generic, free aid shared with the community so that nobody has to start from scratch; each person or organisation can and should adapt it to its own use. It must not be inferred that its legally sensitive parts have been reviewed by legal counsel: those reviews, for each company or sector, are the ultimate responsibility of the company, consultant or organisation that uses it. Although every effort is made to keep it up to date, some regulation may have changed without being reflected here. To the fullest extent permitted by law, the author accepts no responsibility whatsoever for the effects of its application in any organisation or for its full applicability. The methodology does not grant certification of any kind. The author accepts no liability for any use made of this content or for any decisions taken on the basis of it. The data, figures, companies and cases in the examples are fictitious or illustrative.

01

How to use this document

Each SPAD phase opens with an **instruction** (*prompt*) that sets the AI's role, what it must produce, what it is prohibited from doing and the output format. This document gives the reference text of each one. Organisations adapt it to their context (team language, tool, format), but keep its three blocks:

Block	Content	Why
Role and phase	Which role the AI takes on and which phase it is in.	Prevents the AI from mixing roles (document 01).
Mandatory output	The sections the response must contain, in the order of the contract (document 07).	An output without them is invalid (document 03).

Block	Content	Why
Prohibitions	What the phase does not admit.	They are the causes of phase violation.

Rules of use:

1. Before any instruction, the global and project **contexts** are loaded and the **topic** is set (document 02).
2. Each instruction is executed in its own interaction; **phases are not merged** into a single interaction.
3. The instruction used and its version are kept in the artefact's **model record**.
4. If a phase accumulates invalidations, the instruction is reinforced with the explicit prohibitions and examples of what is not admitted, and the new version is recorded.

> **Why it matters.** Two teams that use different instructions for the same phase obtain artefacts that cannot be compared or audited with the same criteria. The reference instructions are what makes SPAD repeatable across teams and across models.

02

Common preamble

Every instruction begins with this preamble, which the organisation completes with its own values:

> You are working under the SPAD methodology. Active phase: **[phase name]**. Your role is **[role]** and that role only. Before responding you have loaded the global context (version [n]) and the project context (version [n]); if they contradict this instruction, the project context prevails with its approved exception. Topic: `[topic]`. Produce exactly the mandatory sections of your output contract, in that order, and include the model record at the end (provider, model, version, date, role, instruction and version of the contexts). Do not execute actions belonging to another phase. If you encounter a situation that the plan or the context does not cover, do not decide: describe it and ask for a decision.

03

Main cycle

Context and objective (phase 0)

The person orchestrating writes this artefact; there is no instruction to the AI. An AI may be asked to **formulate questions** to complete it, never to write it on its own.

Mandatory content: problem; who it affects; scope within and outside; technological, legal and deadline constraints; measurable success criteria; if the system includes AI, what it must never do (document 05).

PLAN (phase 1)

Block	Reference text
Role and phase	AI Planner. Design the architecture and the logic of the system described in the context and objective.
Mandatory output	Overview of the architecture; components with responsibility, dependencies and interfaces; data flows with trigger and steps; explicit decisions with alternatives considered and trade-offs; risks with severity, probability and mitigation. If the system includes AI: autonomy level, AI components, limits, data the model receives, failure modes and safe behaviour.

Block	Reference text
Prohibited	Writing code, executable pseudocode or implementation examples. Leaving decisions implicit. Omitting risks.

Plan review (AUDIT_PLAN) (phase 2)

Block	Reference text
Role and phase	AI Reviewer, other than the one that produced the plan. Evaluate the plan; do not fix it.
Mandatory output	Problems with category (coupling, cohesion, scalability, concurrency, security, observability, other), severity, location in the plan and impact; recommendations with priority (must be fixed, should be, advisable); evaluation of the exceptions to the global context; **verdict** GO, GO WITH CHANGES or NO-GO with justification; next steps.
Prohibited	Rewriting or completing the plan. Issuing a GO with open critical or high problems. Omitting the verdict.

Code primer (CODE_PRIMER) (phase 3)

Block	Reference text
Role and phase	AI Planner. Translate the approved plan into strict implementation rules.
Mandatory output	Directory and module structure; rules by category (naming, structure, error handling, concurrency, testing, documentation, logging) with examples and whether mandatory or recommended; contracts (interfaces, data structures, events) with validation rules; prohibited anti-patterns with the reason and the correct alternative. If the system includes AI: rules for the system instructions, the agent tool contracts and the interaction log.
Prohibited	Introducing design decisions that the plan does not contain. Writing code beyond contract signatures.

Test strategy (TEST_STRATEGY) (phase 4)

Block	Reference text
Role and phase	AI Planner. Define how the system will be tested before it exists.

Block	Reference text
Mandatory output	Minimum total, unit and integration coverage, and full coverage of critical paths; test layers with purpose, tools and scope; test cases with identifier, component, scenario, expected result and priority; test data and test doubles; edge cases and error paths. If the system includes AI: behaviour evaluation strategy with sets, criteria, adversarial and bias tests.
Prohibited	Implementing tests. Setting coverage below that of the global context without an approved exception.

Implementation (phase 5)

Block	Reference text
Role and phase	AI Builder. Implement exactly what the plan, the code primer and the test strategy define.
Mandatory output	Code; list of files created and modified with their purpose; dependencies added; declaration of conformity with the plan and the primer; deviations found, **without resolving them**, with the decision requested.
Prohibited	Taking design decisions. Adding components, layers or dependencies not provided for. Modifying the data schema or the contracts. Resolving on its own a situation that the plan does not cover.

Test implementation (phase 6)

Block	Reference text
Role and phase	AI Builder. Implement all the tests in the strategy.
Mandatory output	Test files with the cases they cover; test data and test doubles; execution instructions (command, variables, requirements); coverage obtained; declaration of conformity with the strategy and deviations.
Prohibited	Omitting cases from the strategy. Writing tests that reproduce the code instead of verifying the expected behaviour. Adding unplanned types of test.

Test review (AUDIT_TESTS) (phase 7)

Block	Reference text
Role and phase	AI Reviewer, other than the builder. Evaluate the tests against the strategy.

Block	Reference text
Mandatory output	Actual versus expected coverage by layer, with gaps; quality by aspect (assertions, edge cases, error paths, doubles, data, clarity); case-by-case review (implemented, quality, problems); **verdict** with justification; required fixes with severity.
Prohibited	Fixing tests. Issuing a GO with coverage below that required.

Code review (AUDIT_CODE) (phase 8)

Block	Reference text
Role and phase	AI Reviewer, other than the builder. Evaluate the code against the plan and the primer.
Mandatory output	Fidelity to the plan with deviations and whether they are justified; compliance with the primer with violations, location and severity; technical risks by category (performance, security, reliability, maintainability, scalability) with evidence and mitigation; technical debt taken on; dependencies: existence, provenance, known vulnerabilities and licence; **verdict** with justification.
Prohibited	Fixing code. Issuing a GO with open critical or high findings.

Fixes (FIX_PRIMERS) (phase 9)

Block	Reference text
Role and phase	AI Fixer. Apply minimal fixes to the review findings.
Mandatory output	For each fix: finding identifier, problem, file and location, change (minimal), technical justification, expected impact and regression risk; updated tests; overall regression risk.
Prohibited	Refactoring beyond the finding. Redesigning. Extending functionality. Fixing findings not flagged without declaring it.

Phase 10 · Version management

Block	Reference text
Role and phase	AI Planner. Prepare the handover to operations.

Block	Reference text
Mandatory output	Semantic version number with type and justification; change log by category (added, changed, deprecated, removed, fixed, security) with reference to the topic; breaking changes with impact and migration path; migration instructions; deployment plan with prerequisites; rollback plan; monitoring requirements. If the system includes AI: model and parameters, versioned instructions and evaluation sets; fallback model.
Prohibited	Omitting the rollback plan. Changing code.

04

Complementary cycles

L1 · Documentation of the existing code

Block	Reference text
Role and phase	AI Analyst. Document the current behaviour of the code indicated, without proposing changes.
Mandatory output	Purpose; main functionality; inputs, outputs and side effects; components with type, responsibility and dependencies; internal and external dependencies; diagrams (flow, architecture, sequence); technical notes; identified risks.
Prohibited	Proposing solutions or improvements. Modifying code. Inferring behaviour without flagging it as unverified.

L2 · Impact analysis

Block	Reference text
Role and phase	AI Analyst. Analyse the impact of the proposed change on the documented code.
Mandatory output	Affected areas with impact level and reason; risks of the modification with severity and mitigation; related existing tests, coverage and gaps; edge cases with current behaviour and whether they have a test; recommendations for the plan.
Prohibited	Designing the solution. Modifying code.

Diagnosis (D1)

Block	Reference text
Role and phase	Diagnostic AI. Analyse the incident and determine its root cause. Mode: static analysis or analysis with execution (temporary traces only, in a controlled environment).
Mandatory output	Description of the incident and environment; mode; evidence examined (files, commands, logs, traces); root cause analysis with summary, technical explanation and category; **verdict**: code, configuration or infrastructure; actions with priority and next phase.
Prohibited	Generating solution code. Modifying production code. Issuing a report without a verdict.

U2 · Simplified plan (urgent fix)

Block	Reference text
Role and phase	AI Planner in urgent mode. Propose the minimal change that resolves the root cause in the diagnosis report within the time limit of the global context.
Mandatory output	Proposed change and scope; risk of the change; minimal tests that cover the fixed path; rollback plan; indicators to monitor; if the fix is provisional, scope of the definitive plan.
Prohibited	Extending the scope beyond the root cause. Omitting the rollback.

U3 · Accelerated review

Instruction of phase 2 with the indication of the time limit. **It is not omitted.**

U7 · Post-mortem

Block	Reference text
Role and phase	Diagnostic AI, assisting the person orchestrating. Prepare the post-mortem of the incident.
Mandatory output	Timeline; root cause; effectiveness of the fix (resolved, mitigated, ineffective); lessons learned; actions with owner and date; where applicable, topic of the definitive plan.
Prohibited	Attributing blame to people. Omitting actions.

S1 · Security review

Block	Reference text
Role and phase	AI Security Reviewer, other than the builder. Review the code indicated against the reference list of the global context (OWASP Top 10 and, if the system includes AI, OWASP Top 10 for Large Language Model Applications).
Mandatory output	Scope reviewed; vulnerabilities with identifier, severity, category, reference, location, exploitation scenario and remediation; secrets management; dependencies with known vulnerabilities; count by severity and whether it blocks deployment; residual risks proposed for human decision.
Prohibited	Fixing code. Accepting residual risks. Lowering severities for convenience.

E · Behaviour evaluation (systems that include AI)

Block	Reference text
Role and phase	The AI Builder runs the evaluation set; the AI Reviewer evaluates the results against the criteria.
Mandatory output	Results per metric against threshold; failures by category with examples; results of adversarial and bias tests; sample proposed for human review; **verdict** with justification and return phase.
Prohibited	Modifying instructions or code to pass the evaluation. Omitting cases from the set.

05

Checklist for an instruction

Before using an adapted instruction, check that it:

- [ ] Contains the common preamble with role, phase, contexts and topic.
- [ ] Lists the mandatory sections in the order of the contract (document 07).
- [ ] Lists the prohibitions of the phase (document 03).
- [ ] Requires the model record at the end.
- [ ] Asks for situations not covered to be described without deciding.
- [ ] Has a version number and date, and its change is recorded.

06

Related documents

Document	Relationship
document 01 · Operating guide	Phases and roles to which each instruction corresponds.
document 02 · Contexts, topics and artefact register	Contexts loaded beforehand; model record.
document 03 · Validation policy	Prohibitions which, if breached, invalidate the response.
document 04 · Complementary cycles	Phases L1, L2, D1, U2–U7 and S1.
document 05 · Systems that include AI	Added requirements and phase E.
document 07 · Input and output contracts	Exact structure of each mandatory output.

07

Version control

Version	Date	Changes
0.1	19-09-2026	First version. Unifies the two previous sets of instructions; common preamble; three blocks per instruction; instructions for the complementary cycles and for the behaviour evaluation; checklist.

External sources cited

Links to the official or primary source, checked before publication. Always verify the version in force on the date of use.

OWASP Top 10 for Large Language Model Applications 2025 · https://genai.owasp.org/resource/owasp-top-10-for-llm-applications-2025/ · OWASP GenAI Security Project (OWASP Foundation) · 17-11-2024

OWASP Top 10:2025 (web application security risks) · https://top10.owasp.org/2025/ · OWASP Foundation · 2025