DOCUMENT 02 · CONTEXTS, TOPICS AND ARTEFACT REGISTER

Contexts, topics and artefact register

Which rules are loaded before any phase, how work is grouped into topics and what is recorded for each artefact, each human validation and each model used

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

2	1	3	20
mandatory contexts	topic per piece of work	records per artefact	maximum characters in the topic

IN THIS DOCUMENT

01 Purpose

02 Contexts

03 Topics

04 Records per artefact

05 Checklist before starting

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. References to general regulation (such as the GDPR) may be incomplete, may not apply to a particular case or may become out of date. **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

Purpose

This document defines three mechanisms that make work with SPAD consistent across projects and reconstructible afterwards:

Mechanism	Question it answers
Contexts	Under which rules does the AI work in this organisation and in this project?
Topics	Where are all the artefacts of this feature, fix or incident?
Records	Who validated each output, with what outcome, and which model produced it?

02

Contexts

Two contexts are loaded before any phase is executed. No instruction (document 06) is executed without them.

2.1 Global context

Universal rules of the organisation, applicable to all projects that use SPAD, regardless of the technology or the domain.

Mandatory section	Content	Illustrative example
Artefacts	Location, names and format of the artefacts and diagrams.	`documentation/{{TOPIC}}/SPAD_NN_PHASE.md` ; diagrams in Mermaid.
Tests	Minimum coverage per layer; mandatory integration tests for interfaces; end-to-end tests for critical flows.	Minimum unit coverage of 80 % (starting value, to be calibrated by each organisation).
Versioning	Semantic versioning; change log with each release; breaking changes documented.	`MAJOR.MINOR.PATCH` .
Baseline security	No secrets in the code; input validation; authentication at sensitive points; technical reference for the security review.	OWASP Top 10 as the reference list.
Data and model providers	Which classes of information may be sent to each approved AI provider and environment, and which never (personal data, secrets, proprietary code according to its classification).	"No real personal data in instructions to external providers; synthetic test data."
Prohibitions	What no phase may do.	Code in the PLAN phase; design decisions in the implementation; self-approval.
Parameters of the complementary cycles	Maximum time limits for the simplified plan and for post-deployment monitoring in the urgent fix; severity scale and who accepts the residual risk.	Starting values, to be calibrated (document 04).
Approved models	Providers, models and versions authorised per role, and which may not coincide in building and reviewing.	"The AI Reviewer uses a model or provider other than the builder's for sensitive code."

> **Why it matters.** The data and providers section is the one most often forgotten and the one that prevents the most damage: without it, each person decides on their own which code, data or secrets to paste into an external tool. It is also the point of connection with the corporate AI use policy (SEVEN-G 31 · Corporate AI policy and acceptable use policy).

2.2 Project context

Rules of the specific project, which complete the global context or make exceptions to it.

Section	Content
Technology	Language, frameworks, versions, database, infrastructure.
Architecture	Patterns and constraints (stateless, multi-instance, queues, latency limits).
Domain rules	Business rules that the design cannot violate (for example, dual authorisation in monetary operations, record retention).
Compliance	Applicable obligations (data protection, sector regulation) and their effect on the design.
Team conventions	Naming, reviews, branches.
Deployment	Environments, windows, constraints.
Justified exceptions	Rules of the global context that this project does not meet, with justification and risk (section 2.3).

2.3 Precedence and exceptions

If the project context contradicts the global one:

1. **The project context prevails**, only if the exception is written down and justified.
2. The exception includes **the risk** of departing from the global rule and how it is mitigated.
3. The AI Reviewer **evaluates** the exception in the plan review and flags it in its verdict.
4. **It is approved by the person orchestrating** (or whoever the global context designates for exceptions), and the approval is kept in the validation record (section 4). No AI approves an exception.

Illustrative example. The global context requires 80 % unit coverage. A legacy code migration project without tests sets 70 % for new code and a 10 % increase per iteration for the legacy code. The AI

Reviewer evaluates it as reasonable because of the gradual plan; the technical owner approves it and it is recorded with the date.

03

Topics

Each piece of work with SPAD is associated with a **topic** (*TOPIC*): a short identifier that groups all its artefacts.

Rule	Detail
Format	`snake_case`, maximum 20 characters, descriptive.
Represents	A feature, a fix, an incident or a use case.
Set	In phase 0, explicitly (given by the person) or inferred (proposed by the AI and confirmed by the person).
Persists	Throughout the work; it does not change midway; it is not reused for another feature.
Valid examples	`user_auth`, `pdf_export`, `payment_gateway`, `cache_mem_leak`.
Invalid examples	`feature_123` (not descriptive), `tmp`, `fix` (generic), long names in another format.

3.1 Organisation of artefacts

```
documentation/
├── {{TOPIC}}/
│   ├── SPAD_01_PLAN.md
│   ├── SPAD_01_AUDIT_PLAN.md          (NO-GO)
│   ├── SPAD_02_PLAN.md                (second iteration)
│   ├── SPAD_02_AUDIT_PLAN.md          (GO)
│   ├── SPAD_02_CODE_PRIMER.md
│   ├── SPAD_02_TEST_STRATEGY.md
│   ├── SPAD_02_IMPLEMENTATION.md
│   ├── SPAD_02_TEST_IMPLEMENTATION.md
│   ├── SPAD_02_AUDIT_TESTS.md
│   ├── SPAD_02_AUDIT_CODE.md
│   ├── SPAD_02_FIX_PRIMERS.md
│   ├── SPAD_02_VERSION.md
│   ├── VALIDATION.md                  (human validation record)
```

```
│   └── diagrams/
└── {{TOPIC_2}}/
```

The **sequential numbering** keeps every iteration: a rejected plan is not deleted; it is kept with its review and the next iteration takes the next number. In this way it can be seen what was attempted, why it was rejected and what changed.

> **Why it matters.** The topic folder is the unit of audit. Anyone who needs to know, months later, why a component is built the way it is finds in a single place the initial intention, the plan, the review, the code, the tests, the fixes, the version and who validated each step.

04

Records per artefact

Each artefact carries three records. The first two are mandatory in all work; the third, in the complete version.

4.1 Model record

What produced the artefact. Without it, a result cannot be reproduced or explained when the provider changes.

Field	Content
Provider and model	Name and version of the model (or identifier of the tool).
Date and time	Of the generation.
Role	Planner, reviewer, builder, fixer, analyst, diagnostic, security reviewer.
Instruction	Reference to the instruction used (document 06) and its version; in the complete version, the full text of the additional instructions.
Contexts	Version of the global and project contexts loaded.

4.2 Human validation record

Who validated the artefact and with what outcome. It is the evidence of human control; without it, traceability only proves what the AI did.

Field	Content
Artefact	Name and iteration number.
Person	Who validates (orchestration role or human code review).
Date	Of the validation.
Outcome	Valid · Invalid (with the cause from the validation policy, document 03) · Exception approved.
Verdict accepted	If the artefact is a review: the AI's verdict and whether the person accepts or rejects it, with the reason.
Observations	What the person adds or disagrees with.

The record is kept in `VALIDATION.md` within the topic (or in the version control system, if the organisation prefers) and is completed **at the time**, not at the end.

4.3 Violation log

Each response discarded under the validation policy (document 03) is noted with date, phase, cause, model and action. It is the raw material for the process metrics (document 08) and for improving the instructions.

> **Why it matters.** The three records answer the three questions of any development audit: which tool did this, which person accepted it and how many times it had to be discarded and repeated. In SEVEN-G, those records are the input to the evidence of the design and delivery phases (SEVEN-G 53, section 7.2).

05

Checklist before starting

- [ ] Global context loaded, with its version.

- ☐ Project context loaded, with its version.
- ☐ Conflicts between contexts identified and exceptions approved by a person.
- ☐ Topic set, in the correct format, and folder created.
- ☐ Models assigned to each role in accordance with the approved models; the AI Reviewer is not the same as the builder.
- ☐ Validation record opened.
- ☐ Class of information authorised for the providers to be used, checked.

06

Related documents

Document	Relationship
document 01 · Operating guide	Phases in which the recorded artefacts are produced.
document 03 · Validation policy	Invalidation causes that feed the records.
document 07 · Input and output contracts	Structure of the artefacts, with the fields of the model record.
document 08 · Self-assessment and metrics	Use of the records as evidence and as the source of the metrics.
SEVEN-G 31 · Corporate AI policy and acceptable use policy	Origin of the rules on data and providers in the global context.
SEVEN-G 53 · Building solutions with AI	How the contexts are derived from SEVEN-G evidence and what traceability requires.

07

Version control

Version	Date	Changes
0.1	19-09-2026	First version. Develops the previous management of contexts and topics; adds to the global context the sections on data and providers, parameters of the complementary cycles and approved models; exceptions are approved by a person; incorporates the model record, the human validation record and the violation log.

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