



TEMPLATE

Algorithm Maturity Assessment Template

A practical, vendor-neutral template for scoring algorithm maturity using evidence from correctness, performance, edge cases, observability, security, and rollback readiness.

Algorithm Maturity Assessment Template

Use this template to evaluate whether an algorithm is ready for prototype, limited rollout, or production use. The goal is to make the decision based on evidence, not opinion.

1) Assessment overview

Algorithm name:

Version / commit / release:

Owner:

Date:

Reviewer(s):

Environment evaluated:

Assessment purpose:

- ☐ [] Prototype review
- ☐ [] Limited rollout review
- ☐ [] Production readiness review



- ☐ Post-incident reassessment
- ☐ Other: _____

Summary decision:

- ☐ Level 1 ? Experimental
- ☐ Level 2 ? Validated in a limited scope
- ☐ Level 3 ? Operationally tested
- ☐ Level 4 ? Production-ready
- ☐ Level 5 ? Production-hardened

Decision notes:

2) Scope statement

Describe exactly what the algorithm is intended to do.

Problem it solves:

Non-goals:

Input domain:

Expected output shape:

Assumptions:

Constraints:

- Latency target:
- Throughput target:
- Memory target:
- Other safety or business constraints:

Acceptable failure behavior:



3) Maturity scoring criteria

Score each area from 0 to 4:

- 0 = Not assessed / no evidence
- 1 = Weak evidence
- 2 = Basic evidence
- 3 = Strong evidence
- 4 = Production-grade evidence

Scoring table

Criterion	Score (0-4)	Evidence summary	Gaps / risks
Problem fit			
Correctness			
Complexity / performance			
Boundary and failure cases			
Observability			
Security / misuse risk			
Rollback / containment			

Total score: __ / 28

Recommended maturity level:

- ☐ Level 1
- ☐ Level 2
- ☐ Level 3



- ☐ Level 4

- ☐ Level 5

Rationale for level selection:

4) Evidence checklist

A. Problem fit

- ☐ The algorithm solves the actual business or system problem
- ☐ Input characteristics match real-world conditions
- ☐ Simplifying assumptions are explicitly documented
- ☐ Known non-goals are accepted by stakeholders

Evidence / notes:

B. Correctness

- ☐ Unit tests cover core branches
- ☐ Golden test cases with known outputs exist
- ☐ Property-based or randomized tests were used where appropriate
- ☐ Cross-check against a reference implementation was performed
- ☐ Manual review covered tricky transitions or state changes

Evidence / notes:

C. Complexity and resource use



- ☐ Time complexity is documented
- ☐ Memory growth is documented
- ☐ I/O or network cost is documented if relevant
- ☐ Performance was measured against representative input sizes
- ☐ Pathological input sensitivity was evaluated

Evidence / notes:

D. Boundary and failure cases

- ☐ Empty inputs tested
- ☐ Minimal inputs tested
- ☐ Maximum-sized inputs tested
- ☐ Null / missing / duplicate values tested where relevant
- ☐ Invalid formats tested
- ☐ Timeout or partial-failure conditions tested

Evidence / notes:

E. Observability

- ☐ Input size or distribution can be observed
- ☐ Output counts or key result categories can be observed
- ☐ Error rates and failure reasons are visible
- ☐ Latency and resource consumption are measured
- ☐ Version or configuration is recorded at runtime

Evidence / notes:

F. Security / misuse risk



- ☐ Input validation gaps were reviewed
- ☐ Denial-of-service risk was evaluated
- ☐ Information leakage risk was considered
- ☐ Trust boundaries and authorization assumptions were reviewed
- ☐ Sensitive data handling was assessed if applicable

Evidence / notes:

G. Rollback / containment

- ☐ Feature flag or runtime switch exists
- ☐ Safe fallback path exists
- ☐ Deployment is versioned and reversible
- ☐ Bad inputs can be isolated or quarantined
- ☐ Rollback steps are documented and tested

Evidence / notes:

5) Test record

Use this section to record the evidence used to validate maturity.

Test summary

Test type	Dataset / case	Expected result	Actual result	Pass / fail	Notes
Unit test					
Integration test					



Golden case					
Boundary case					
Invalid input case					
Load / performance test					
Security / abuse case					

Notable findings

Passed with confidence:

Failed and fixed:

Open issues:

6) Operational readiness checklist

- ☐ Monitoring is in place
- ☐ Alerts are defined for abnormal behavior
- ☐ Logs are sufficient to reconstruct failures
- ☐ Traces or equivalent diagnostics are available if needed
- ☐ Runbook exists for common incidents
- ☐ Owners and escalation path are known
- ☐ Rollback procedure has been validated
- ☐ Known limitations are documented

Operational notes:



7) Release decision

Recommended action:

- ☐ Do not release
- ☐ Limited rollout only
- ☐ Production release with safeguards
- ☐ Production release approved
- ☐ Reassess after further evidence

Conditions required before approval:

Residual risks accepted:

Approver:

Approval date:

8) Level guide

Level 1 ? Experimental

Use when the algorithm is still a rough prototype.

Typical evidence:

- Minimal or no testing
- Performance unmeasured
- Failure behavior undefined

Level 2 ? Validated in a limited scope



Use when the algorithm works for a narrow use case.

Typical evidence:

- Core logic tested
- Constraints documented
- Boundary cases partially covered

Level 3 ? Operationally tested

Use when the algorithm has been exercised under realistic conditions.

Typical evidence:

- Representative tests or integration tests exist
- Performance measured under expected load
- Monitoring or logging available

Level 4 ? Production-ready

Use when the algorithm is stable enough for controlled production use.

Typical evidence:

- Repeatable correctness evidence
- Performance within defined limits
- Security and rollback reviewed

Level 5 ? Production-hardened

Use when the algorithm has proven reliability over time.

Typical evidence:

- Operational history exists



- Monitoring and alerts tuned
- Changes are versioned and regression-tested

9) Reviewer notes

What most strongly influenced the score?

What evidence was missing?

What should be re-tested later?

Final reviewer comments:

10) Quick use instructions

- Define the exact problem and success criteria.
- Gather tests, measurements, and failure cases.
- Score each criterion using evidence.
- Assign a maturity level based on the documented results.
- Record any release conditions, rollback steps, and follow-up actions.

11) Suggested decision rule

If any of the following are true, do not mark the algorithm production-ready:

- Correctness is not demonstrated for the intended input space
- Performance is unmeasured or exceeds the service budget
- Important edge cases are untested



- Observability is insufficient to troubleshoot failures
- Security or misuse risks are unresolved
- Rollback or containment is not available

12) Sign-off

Prepared by:

Reviewed by:

Approved by:

Date:

Next review date: