AI-Native Development Criteria

Purpose

A structured diagnostic to score your development environment's AI-Native readiness on a 100-point scale across four perspectives.

How to Use

1. Read each perspective and its sub-items
2. Score each item based on the criteria provided
3. Sum scores to determine your level

Score Summary

Level	Score	State
Excellent	90–100	Fully structured AI-Native workflow with high-quality context
Good	70–89	Workflow established with conscious quality practices
Fair	50–69	Core workflows in place, but operations and autonomy are immature
Poor	30–49	Initial efforts underway, but workflows are fragmented and unintegrated
Critical	0–29	Not started or early stage

Evaluation Scope and Premises

Scope: Evaluate against the product's primary working repository (or single monorepo).

Context fragmentation: If core context is distributed across multiple repositories, and cross-repository search, reference, and update do not complete within a single workflow, context closure scores low.

External sources of truth: If the SSoT (Single Source of Truth) for PRDs, design documents, or tasks resides in an external tool, that item scores **0**.

Why "Updatable" Is Non-Negotiable

Read-only documents are not true context. Context is something you **grow**—not something you write once and reference forever.

As LLMs execute development work, they need to:

- Record design decisions discovered during implementation
- Add constraints that emerge from technical investigation
- Update specifications that prove incomplete or incorrect

Context that cannot be updated drifts from reality. "Readable but not updatable" context becomes stale and misleading as the codebase evolves—and misleading context is worse than no context, because the LLM trusts it.

Allowing external SSoTs would mean tolerating this drift. This criteria does not. **Search, reference, and update must all complete within the LLM's workflow.** Without this, LLMs cannot operate autonomously.

Point Allocation Rationale

Perspective	Points	Rationale
Context Closure	15	Necessary foundation, but a prerequisite—not a differentiator
Codified Principles	30	Highest leverage: determines every LLM decision's quality
Workflow and Guardrails	25	Operationalizes principles into enforced practice
Context Quality	30	Existence of principles is meaningless if poorly expressed

Principles and Quality together account for 60 out of 100 points. This reflects a core belief: an LLM with well-written, project-specific principles in a messy repository will outperform an LLM in a perfectly organized repository with no principles.

Scoring Rules

- Perspectives 1–3 are evaluated **cumulatively**. If foundational items are not met, dependent items score **0** (no renormalization).
- Perspective 4 (Context Quality) is evaluated across all artifacts, independent of Perspectives 1–3.
- Qualitative evaluation requires actual file content review.
- Evaluate the essence of "providing appropriate context to LLMs"—avoid tool-specific assessments.
- **Evidence-based scoring**: Score based on what you can verify in the repository, not what you believe to be true.
  - "We have documentation" is not evidence—open the files and evaluate their content
  - "We do reviews" is not evidence—check whether review integration is defined in the workflow

Perspective 1: Context Closure (15 points)

Whether all product context is in a state where search, reference, and update complete within the LLM's workflow.

All three capabilities must complete within the workflow:

Capability	Meaning	Met	Not Met
Searchable	Discoverable without knowing exact paths	Files in repo (grep/glob), semantically searchable docs	Single file reference in external tool

Readable	Loadable into LLM context	File read, API retrieval	Documents behind authentication walls
Updatable	LLM or workflow can maintain content	Markdown in repo (edit/write)	Read-only external tool integration

A. Product-Level Closure (9 points)

Item	Points	Criteria
Design documents	3	PRDs, Design Docs, ADRs, and similar artifacts are searchable, readable, and updatable within the workflow
Task management	3	Tasks (issues/tickets) are searchable, readable, and updatable within the workflow
Infrastructure/platform definitions	3	Terraform, k8s manifests, CI definitions, and similar configurations are searchable, readable, and updatable within the workflow

Scoring per item:

Condition	Points
Search, reference, and update complete within primary repository	3
Separate repository, but cross-repository workflow completes all three	2
Search and reference possible, but update not possible	1
Reference only, or external SSoT	0

B. Application-Level Closure (6 points)

Item	Points	Criteria
Feature-based directory structure	4	Core services, modules, contract definitions (API, SDK, Proto), and infrastructure definitions are organized by feature/domain within the same repository

Colocation	2	Tests, type definitions, and styles colocated within corresponding features (holds for most core features)

Scoring:

- Feature-based: Same repo + all core domains = 4, partial = 2, layer-based = 0
- Colocation: Most core features = 2, some = 1, separated = 0

When scoring: Feature-based structure with bloated files undermines closure effectiveness. Complexity concentrated in specific directories indicates insufficient separation of concerns.

Perspective 2: Codified Principles (30 points)

Whether project-specific principles are documented in a form LLMs can reference. Not generic rules—**judgment criteria specific to this project**.

Foundation (5 points)

Item	Points	Criteria
Principle document exists	5	AGENTS.md (or equivalent) functions as entry point with universal principles stated concisely, routing to skills/references for details

Scoring:

- 5: Entry point contains minimal universal principles + routing only (no task procedures)
- 3: Entry point mixes task procedures or multi-domain details
- 0: No principle document, or external links only

Coverage (25 points)

Domain	Points	Criteria
Coding standards & design principles	5	Project-specific coding rules, design patterns, naming conventions documented

Domain-specific knowledge	5	Business logic, domain terminology, business workflows documented
Architecture decision criteria	5	ADRs exist with current decision criteria summarized separately
Done definitions	5	Mechanical completion criteria (tests, verification, acceptance) documented so LLM can judge autonomously
Structured review perspectives	5	Project-specific review criteria structured, not reliant on generic templates alone

Scoring notes:

- Done definitions: PR template only = 1, per-task mechanical criteria = 5
- Review perspectives: Generic-dominant = 1, project-specific-dominant = 5

When scoring: Abundant technical debt markers (TODO/FIXME/HACK) without codified principles are evidence of subjective frustration accumulation, not healthy management. Debt markers grounded in principles are evidence of conscious debt tracking.

Perspective 3: Workflow and Guardrails (25 points)

Whether operational mechanisms exist to translate principles into execution.

Foundation (2 points)

Item	Points	Criteria
Commands/agents/skills exist	2	Some form of task execution definition exists

Task Execution Environment (6 points)

Item	Points	Criteria
Navigation design (entry point → skills → references)	3	Progressive context loading from entry point—minimal context acquired at each stage

Context reference	2	Skills/commands specify reference targets and purposes, loading only when needed
Specialized agent definitions	1	Single-responsibility agents defined for context control and bias elimination

Scoring:

- Navigation: Entry point is minimal principles + routing = 3, entry point is overloaded / commands carry excessive context = 1, no navigation = 0

Review Workflow (7 points)

Item	Points	Criteria
Review integration point	3	Review occurs **during implementation**, not dependent on PR/CI post-stage
Design/test priority	2	Design doc review and test review are built in as **highest priority**
Review perspective reference	2	Structured review perspectives are referenced

Scoring:

- Integration point: During implementation = 3, PR/CI only = 1, none = 0
- Design/test priority: Documented + prioritized in workflow = 2, documented only = 1, none = 0

Test Verification & Quality Assurance (6 points)

Item	Points	Criteria
Test generation/verification commands	2	Commands or skills for testing are defined
Test guidelines reference	2	Testing guidelines are referenced
Quality assurance guardrails	2	Pre-commit, local verification, or AI auto-checks run **during implementation**

Scoring:

- Guardrails: CI only = 1, automated verification during implementation = 2

Value Stream Integration (4 points)

Item	Points	Criteria
Integrated review mechanism	2	A mechanism to synthesize multiple review results is defined
Conflict resolution rules	2	Priority and adjudication rules for conflicting perspectives are documented

Scoring:

- Integrated review: Explicit integration output format = 2, partial = 1, none = 0
- Conflict resolution: Documented priority/adjudication rules = 2, implicit/ad-hoc = 0

When scoring: Test verification workflows with low-quality tests (low behavioral coverage, high mock density) indicate the workflow is ineffective regardless of its existence. High test quality maintained through workflows is evidence that guardrails are functioning.

Perspective 4: Context Quality (30 points)

Quality criteria for all context artifacts: AGENTS.md, commands, agents, skills, rules, and design templates.

4-1. Single Responsibility (5 points)

Each context artifact serves one purpose/domain.

Points	Criteria
0	Multiple unrelated responsibilities mixed in a single file
3	Generally separated, but some responsibility mixing remains
5	Each artifact clearly serves a single purpose

Detection indicators:

- Single command/skill handles multiple unrelated tasks
- AGENTS.md contains detailed procedures beyond entry-point role
- Agents split by review perspective, not by context-control purpose

4-2. Consistency (5 points)

No contradictions between artifacts; all references match the current codebase.

Points	Criteria
0	Contradictory instructions between files, or stale descriptions remain
3	Main artifacts are consistent, but some inconsistencies exist
5	All artifacts are coherent and match the current codebase state

Detection indicators:

- AGENTS.md instructions contradict command procedures
- References to deleted files or APIs remain
- Conflicting rules between rule files

4-3. Explicitness (5 points)

No ambiguity; LLM can judge and execute without hesitation. Stated in positive form.

Points	Criteria
0	Many ambiguous instructions; relies on negative-form instructions
3	Main instructions are clear, but some ambiguity or negative forms remain
5	All instructions are specific, positive-form, with no room for LLM hesitation

Detection indicators:

- Vague instructions like "write good code"
- Negative-form instructions like "don't do X"
- Success criteria or output formats undefined

- Skills/commands with ambiguous purpose, usage conditions, or I/O

4-4. Target Clarity (5 points)

Written for LLMs. No human-oriented preambles, decoration, or explanatory prose.

Points	Criteria
0	Human-oriented documentation repurposed as LLM context
3	Generally LLM-targeted, but human-oriented decorative text remains
5	All artifacts designed with LLMs as the target audience

Detection indicators:

- Preambles like "This document aims to..."
- Decorative headings or emoji
- Verbose explanations assuming human readers
- README content copied verbatim

4-5. Project Specificity (5 points)

Focused on project-specific judgment, not generic advice. Not verbose.

Points	Criteria
0	Mostly transcribed generic best practices or copied from other projects
3	Contains project-specific content, but generic advice is mixed in
5	All statements are project-specific judgments; nothing that should be handled by tooling

Detection indicators:

- Content like "use camelCase for variables" that linters/formatters should handle
- Transcribed generic coding principles
- Settings copied from another project without understanding

4-6. Quality Assurance Mechanisms (5 points)

Each task/principle has clear purpose and criteria, with verification mechanisms.

Points	Criteria
0	Procedures only; no purpose, criteria, or verification steps
3	Purpose is clear, but verification means (checklists and similar) are partial
5	Each task/principle has purpose, criteria, and quality checklists enabling artifact verification

Detection indicators:

- Lists of procedures without stated purpose
- Execution instructions without verification steps
- Undefined behavior for uncertain situations