

DRY, SOLID, SSOT, KISS Rubric

CRITERION	1	2	3	4	5
Code duplication	Same logic copy-pasted in 5+ places, no shared utilities	Some helpers exist but duplication still common in feature code	Shared hooks/utils for repeated patterns, most duplication removed	Extracted custom hooks, utility library, lint rule flags obvious duplication	Near-zero duplication, automated detection in CI, refactoring discipline
Single responsibility	God components doing fetch + render + validation + routing	Some separation but business logic mixed into UI components	Logic in hooks, UI in components, API layer separate	Clear module boundaries, each file has one reason to change	Dependency injection, interface-driven boundaries, independently testable modules
Source of truth	Same constant defined in 3 files, config scattered, types out of sync	Some centralization but derived values manually duplicated	Config in one place, types generated or co-located with schemas	Schema-driven types (Zod/OpenAPI), single config consumed everywhere	End-to-end SSOT: DB schema ' API types ' UI types, zero manual sync
Simplicity	Premature abstractions, 6-layer indirection for a simple CRUD	Over-engineered in spots, some dead code, unclear naming	Code readable at a glance, abstractions justified by actual reuse	Minimal indirection, no dead code, naming conveys intent clearly	New dev productive in hours, every abstraction earns its keep, docs optional

Code Consultant