# **Pre-Coding Essentials (Component: Workspace Manifest Cargo.toml, Version/FormulaID: VM-ENGINE v0)**

## **1) Goal & Success**

Goal: Define a reproducible Rust workspace (members, profiles, features).

Success: cargo build --locked and cargo test --locked pass on Win/Linux/macOS; no network at runtime; deterministic release outputs.

## **2) Scope**

In scope: [workspace] members/default-members, shared features, profiles, resolver.

Out of scope: Per-crate deps/logic (crates/\*), .cargo/config.toml network flags, Tauri config.

## **3) Inputs → Outputs (with schemas/IDs)**

Inputs: crates/\* paths; rust-toolchain.toml; Cargo.lock; .cargo/config.toml.

Outputs: target/ build artifacts; stable cargo metadata; reproducible profile settings.

## **4) Entities/Tables (minimal)**

## **5) Variables (only ones used here)**

## **6) Functions (signatures only)**

(Manifest has no functions.)

## **7) Algorithm Outline (bullet steps)**

Declare [workspace] members for all crates (vm\_core, vm\_io, vm\_algo, vm\_pipeline, vm\_report, vm\_cli, vm\_app folders).

Set [workspace].default-members to CLI/core crates (exclude vm\_app/Tauri by default).

Enable resolver = "2".

Define [profile.dev] and [profile.release] with lto, codegen-units=1, panic="abort", strip=true (where supported).

Define shared [workspace.dependencies] or [patch] only if pinning exact versions is needed (optional).

Define [workspace.metadata] block for engine metadata (optional, non-normative).

## **8) State Flow (very short)**

Steps: resolve → compile → test.

Stop/continue: stop on missing member path, feature resolution failure, or lockfile mismatch.

## **9) Determinism & Numeric Rules**

Toolchain pinned via rust-toolchain.toml; lockfile required via --locked.

Profiles force codegen-units=1, lto=fat, panic=abort for release.

No numeric rules here.

## **10) Edge Cases & Failure Policy**

If Tauri toolchain absent → keep vm\_app out of default-members; build vm\_cli only.

Any crate adds a build script needing network → fail under --locked; vendor or pin deps.

Mixed arch builds (x86\_64/arm64) must still pass with same profiles.

## **11) Test Checklist (must pass)**

cargo metadata --no-deps OK.

cargo build --locked -p vm\_cli OK on Win/Linux/macOS.

cargo test --locked OK across core crates.

cargo tree --duplicates clean (or explained).

Re-run cargo build --locked with clean target/ → identical outputs (tooling hash checks handled outside Cargo).