

DELIVERY SYSTEM · CLI + CONFIGURATION

From empty project to working WordPress

An owned CLI turns the start of a WordPress site into a reviewable flow: environment, database, core, security, and first artifact activation.

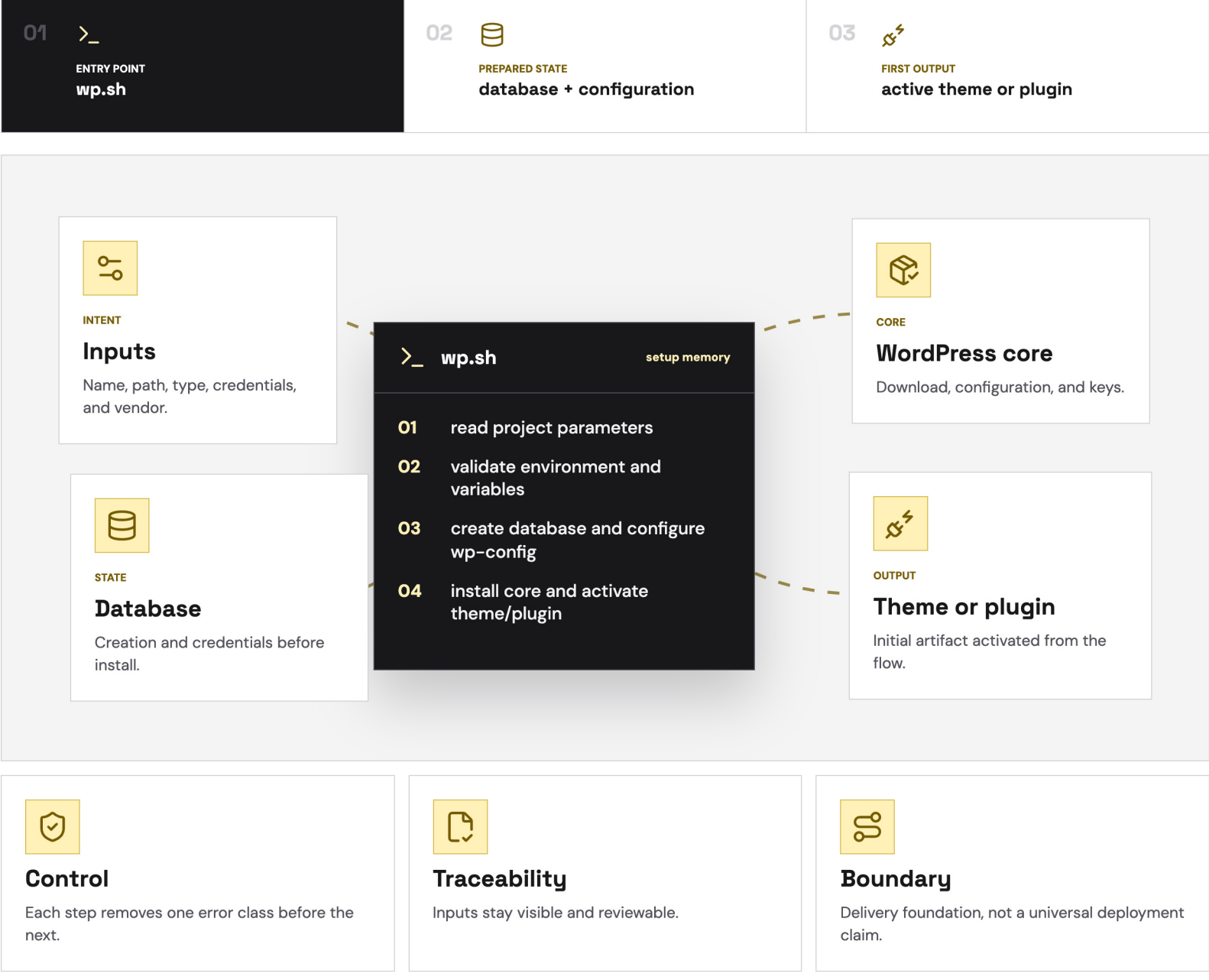
DELIVERY

WP-CLI · Bash 4+ · MySQL/MariaDB · Composer

SYSTEM MAP

A sequence you can read at a glance.

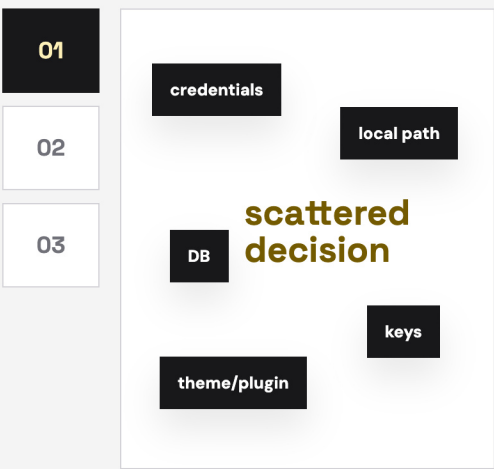
The CLI turns choices once scattered across terminal, notes, and memory into a chain of actions with identifiable input, state, and output.



01 · Situation

The hard part was not installing; it was remembering the right sequence.

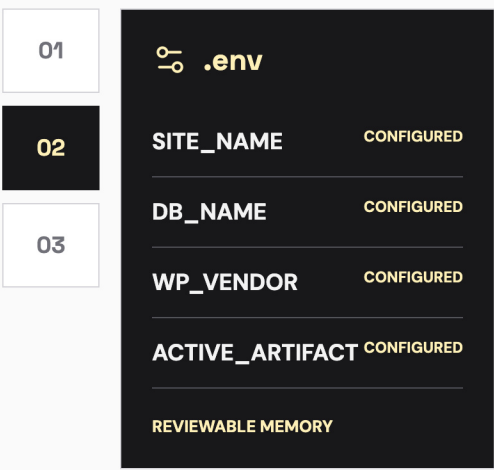
Every site repeated environment choices, credentials, database setup, security keys, install path, and the first active artifact. When those decisions live in memory, the cost appears later: project drift, setup that is hard to review, and technical time spent on steps that do not improve the product.



02 · System

The CLI makes explicit what used to be a loose conversation.

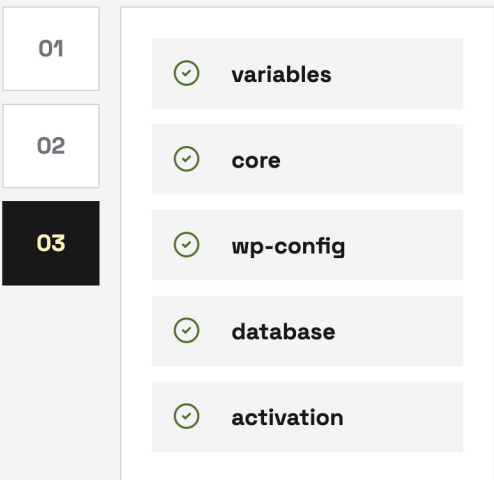
The wordpress-cli repository documents requirements such as PHP, WP-CLI, and Bash 4+. Its script receives a site name, theme or plugin, script paths, and vendor paths; when data is missing, it can ask for it and persist it in environment files so the start has reviewable memory.



03 · Evidence

The demonstrated value is a delivery foundation, not an inflated promise.

The evidence supports an owned system for preparing WordPress with less friction: checking variables, downloading core, generating configuration, creating the database, installing, and activating. It does not claim public adoption or universal deployment for every project.



Tooling does not replace judgment. It prevents spending that judgment again on the same setup.