I'll be implementing this project in 3 different phases:

Phase A: Core Runner and Foundational Assertions setup (Weeks 1-4)

Goal: In this phase, I'll be implementing the fundamental architecture and execution logic of the test runner. This includes implementing the core harness class, the test queue management, sequential execution, the text context object (t), and the most frequently used assertion methods (ok, strictEqual, equal, end). Basic TAP version and assertion line reporting will be implemented. Emphasis will be on creating

Phase B: Full Assertion Set, String Interpolation and CLI setup (Weeks 7-9)

Goal: In this phase, I'll be completing the runner's feature set by implementing the rest of assertion methods (throws, notEqual, notOk, fail, deepEqual, pass, notDeepEqual) used by stdlib. Focus will be on implementing accurate tape-like behavior, especially for throws and deepEqual. Then I'll be integrating @stdlib/string/format for built-in message interpolation in assertions and comments. Develop the stdlib-test CLI for test discovery and execution. Refine TAP output, particularly the diagnostic information for failures(YAML block), add final summary reporting.

Phase C: Migrating Tooling, Execution and Verification (Weeks 10-11)

Goal: In this phase, I'll be creating an automated script to handle the transitioning of code from tape to the newly implemented runner. The script will be executed, and the primary focus will shift to running the entire stdlib test suite with the new harness, rigorously identifying, debugging and resolving any discrepancies or failures to ensure test suite integrity and CI passage.