Azul Arc — Node.js TDD + BDD Playbook (Workshop Edition)

# 0) What we’re using

- Node.js (LTS or latest) — runtime.

- Fastify — minimal API framework.

- Supertest — HTTP testing.

- Node’s built-in test runner — lean unit/integration tests.

- Nodemon — re-runs tests on save.

- Cucumber.js — BDD with Gherkin.

- Optional: Jest/Vitest + Test Explorer for GUI integration.

# 1) Machine prep (Windows quick-start)

Install Node.js:

choco install nodejs-lts -y

Fix npm.ps1 issue:

Set-ExecutionPolicy -Scope CurrentUser -ExecutionPolicy RemoteSigned -Force

Verify:

node -v && npm -v

# 2) Project scaffold

npm init -y

npm i fastify

npm i -D supertest nodemon @cucumber/cucumber

npm pkg set type="module"

npm pkg set scripts.test="node --test"

npm pkg set scripts["test:watch"]="nodemon -x \"node --test\" -w src -w test"

npm pkg set scripts.bdd="cucumber-js --require features/steps/\*\*/\*.js"

npm pkg set scripts["bdd:report"]="cucumber-js --require features/steps/\*\*/\*.js --format html:reports/cucumber.html --format json:reports/cucumber.json"

# 3) App code — src/app.js

Contains Fastify app with:

- POST /standups

- GET /standups

- GET /standups/blockers

- GET /standups/export?format=csv

# 4) TDD tests — test/app.test.js

Tests API endpoints with node:test + supertest.

Run in watch mode:

npm run test:watch

# 5) BDD — features + steps

Feature file in features/standups.feature.

Steps in features/steps/standups.steps.js.

Run:

npm run bdd

Generate reports:

npm run bdd:report

# 6) Teaching flow (60–90 minutes)

1. Kickoff

2. TDD loop (red-green-refactor)

3. Add blocker scenario

4. BDD demo

5. CSV export scenario

6. Retro

Optional: use LLM prompts for pairing.

# 7) Common issues & fixes

- Chocolatey permission: run as Admin.

- npm.ps1 blocked: Set-ExecutionPolicy RemoteSigned.

- ESM vs CJS errors: add "type":"module".

- Tests hang: ensure await app.ready() + app.close().

- Nodemon runs \node: fix script.

- Cucumber option unknown: remove --publish-quiet.

# 8) Optional: Test Explorer UI

For testers wanting GUI:

- Jest: npm i -D jest + Jest VS Code extension.

- Vitest: npm i -D vitest + Vitest Runner/Lab extension.

Keep node --test for devs, add jest/vitest scripts for testers.