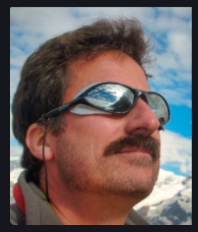
Lean, Agile and Large-Scale Product Development (LeSS)





Jon Kern

2+hour Hands-on Workshop

[ideate · collaborate · build · ship · iterate]

WHY WE'RE HERE

Today's Goal



Not just hear about Vibe Coding — feel it.

By the end of this workshop, you'll have experienced vibe coding inside real agile practices — not as a demo, but as something you built with your own hands.

Collaborate

Work together like a real agile team

Build

Ship working software in 2 hours

Reflect

Walk away with a repeatable approach

THE STAKES

The Promise & The Trap

THE PROMISE

Explosive productivity with AI agents.

THE TRAP

Without agile guardrails: hallucinations, mess, security holes, demos that crumble.

The Agile Manifesto still applies.

Individuals, working software, collaboration, and responding to change.
And all the underpinnings of technical, professional, practices as demanded by your context

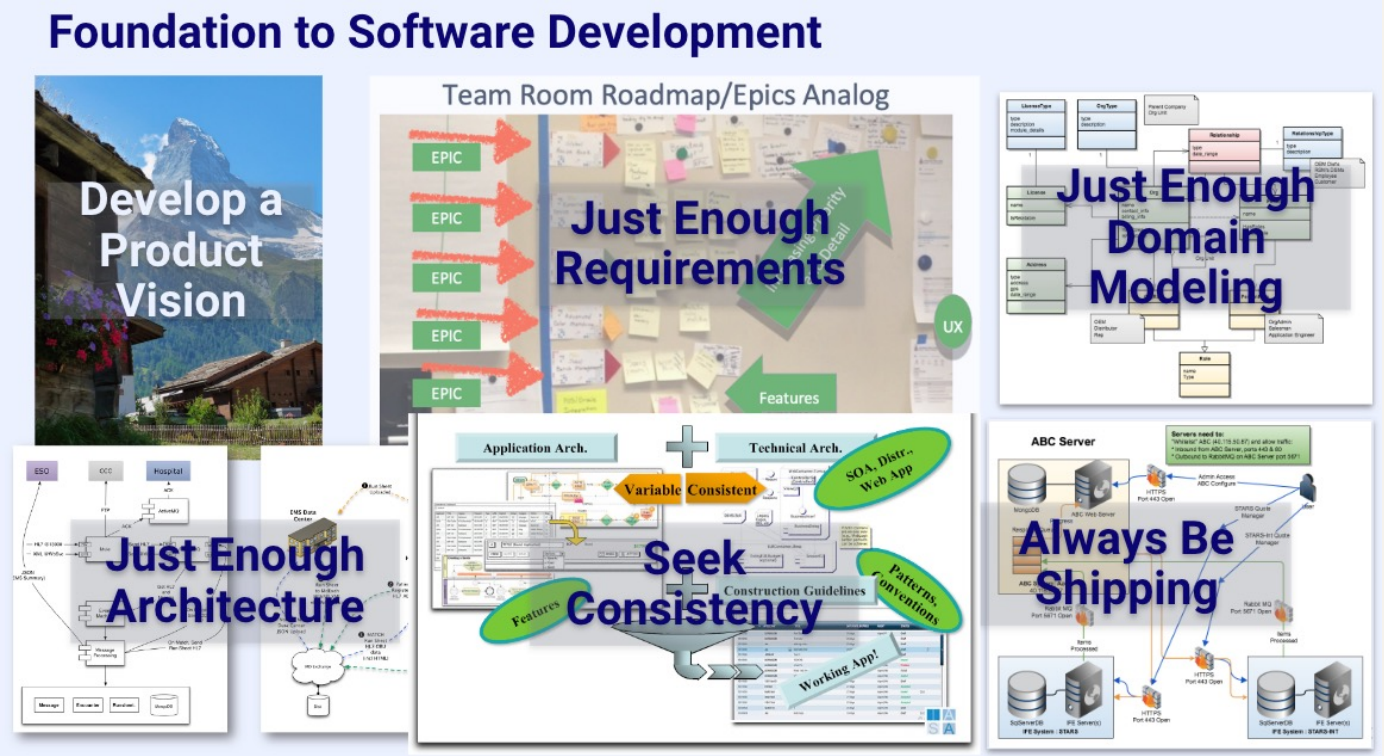
FOUNDATION

Foundational Principles

Applies for the whole product and down to the individual feature...

- Purpose
- Requirements
  - Pains/Gains
- Domain model
- Architecture
- Design
- Deploy

In *small steps* using Behavior-driven Development (BDD)



01 / EXERCISE

GROUP EXERCISE

Product Vision

Write your one-sentence vision — and name the value it delivers to the end user.

A TEMPLATE TO START

For the [user type] who needs [pain relief/gain], our product delivers [real value].

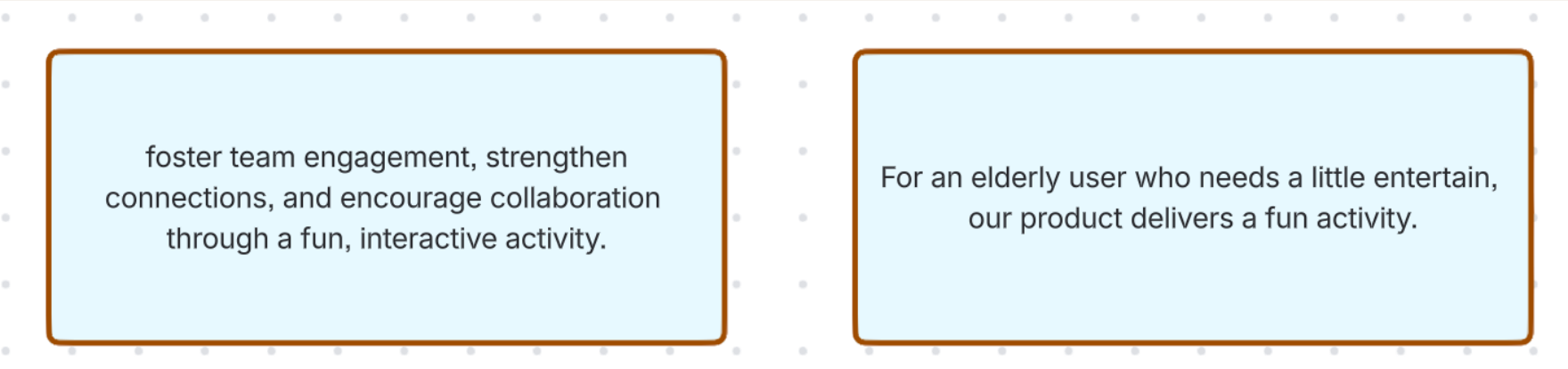
FOUNDATION

Business Purpose

In 25 words or less...

For the marketer who needs a way to stay awake during presentations, our product delivers an entertaining way to keep track of marketing fluff.

As meeting attendees, who need relief from buzzwords used by the CxOs and Industrial Agile Consultants, we need some fun on the side!



02 / EXERCISE

GROUP EXERCISE

Story Mapping

A la Jeff Patton & Feature-Driven Development.
Map a user journey or two to complete a useful task.

THE BACKBONE

Set Up Game	Join Game	Play a Round	Win & Share

FIRST STORIES

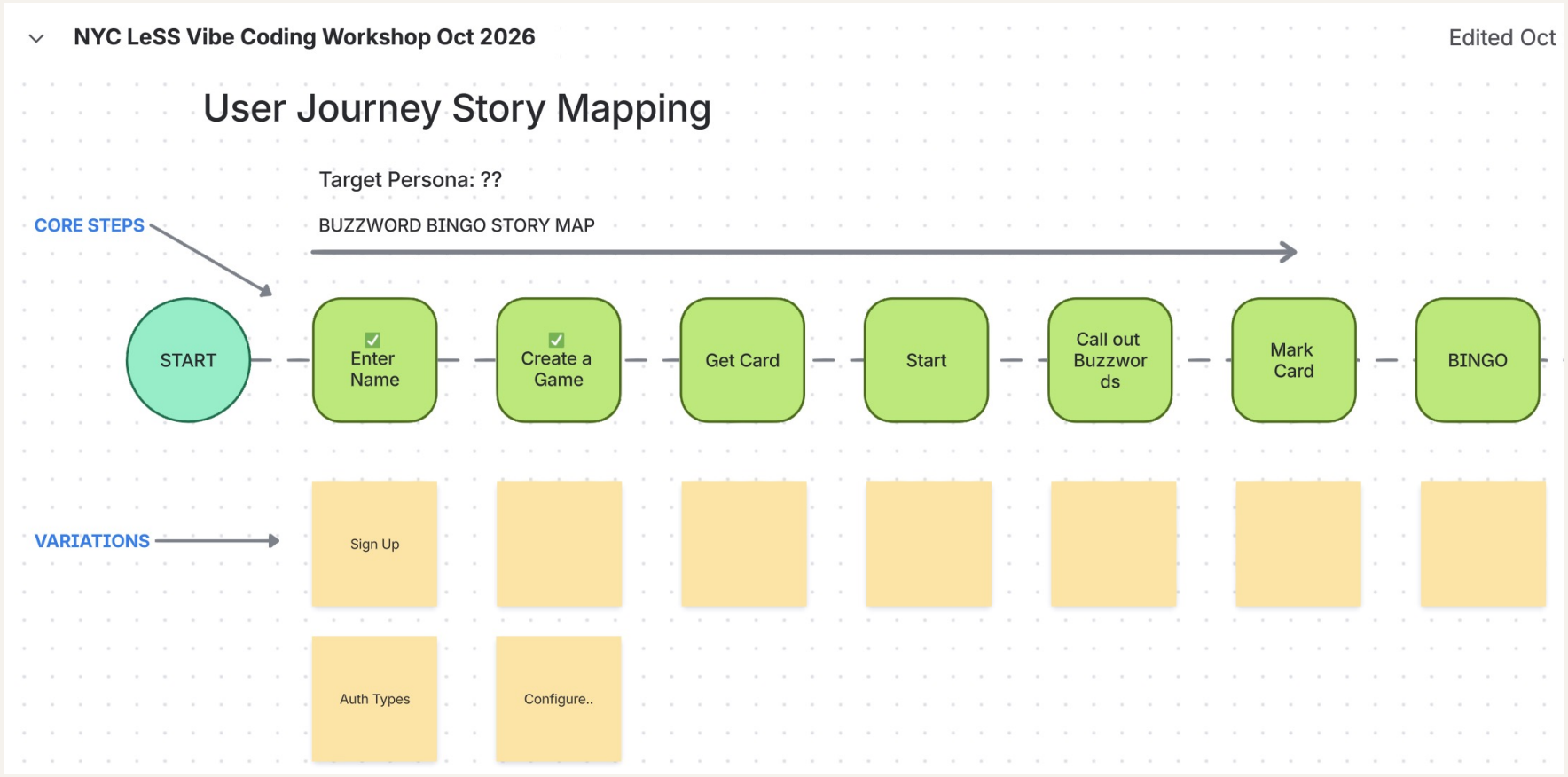
Generate a bingo card	Enter a game code	Tap a buzzword to mark it	Detect & call BINGO

Show the map → pull the first 4–5 stories onto the Kanban board.

FOUNDATION

Story Map(s)

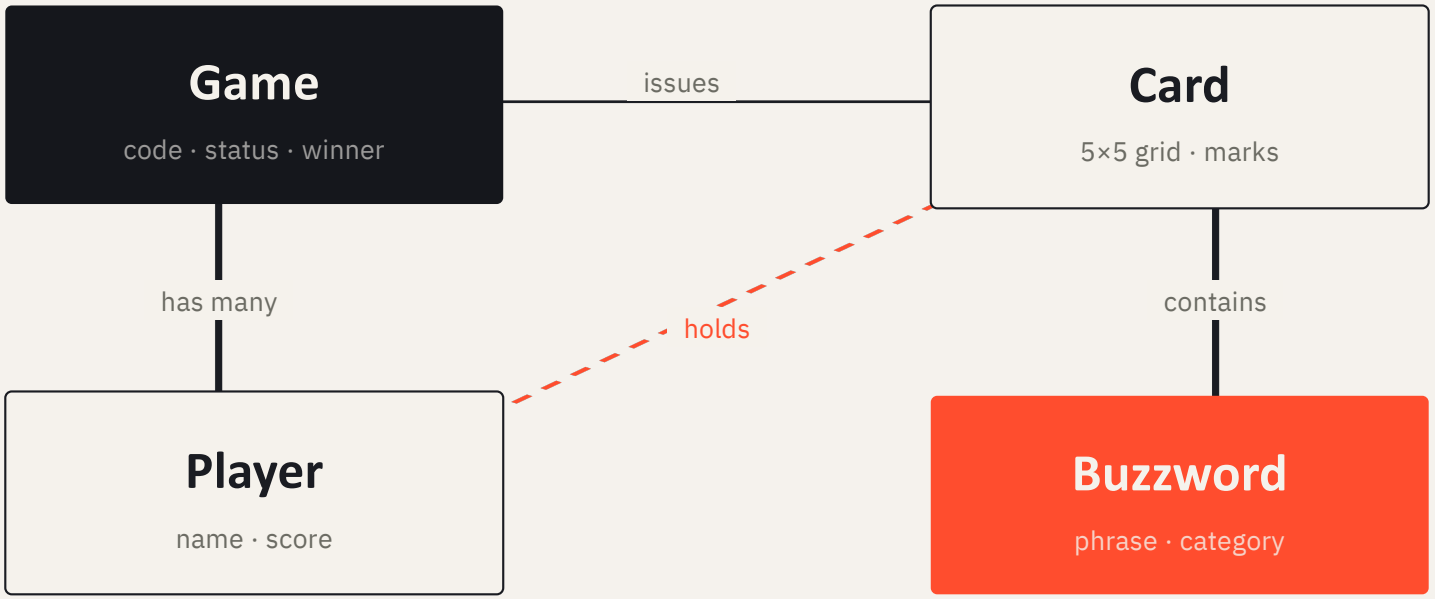
The user journey...



DOMAIN

Domain Model Sketch

Agile Buzzword Bingo — just enough modeling.



FOUNDATION

Domain Model

The core elements

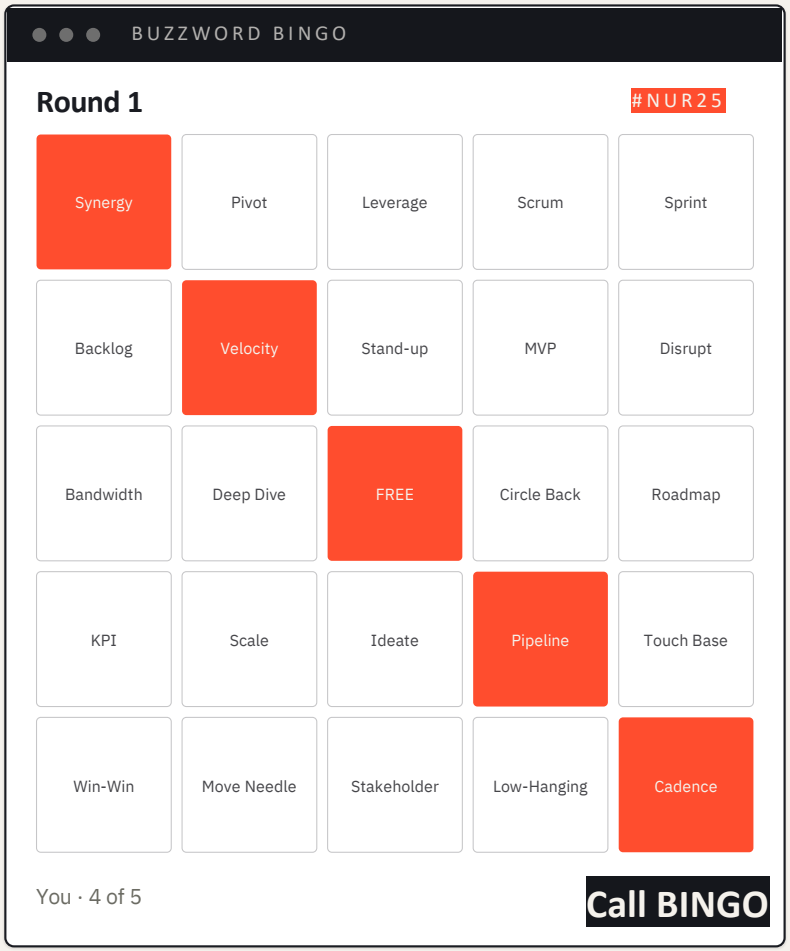
UI SKETCH

Interface Sketch

Agile Buzzword Bingo — just enough to start building.

- A 5×5 card of buzzwords, FREE in the center.
- Tap a square to mark it when you hear it.
- Five in a row → call BINGO and share the win.

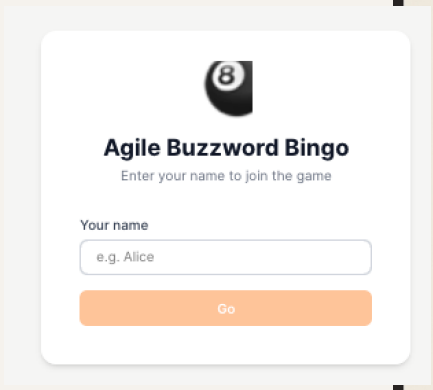
LOW-FI ON PURPOSE — REFINE IN CODE.

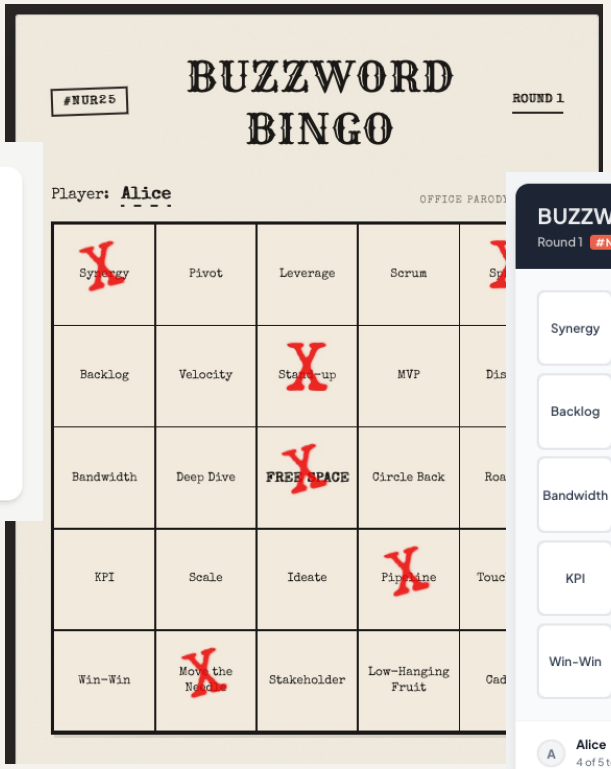


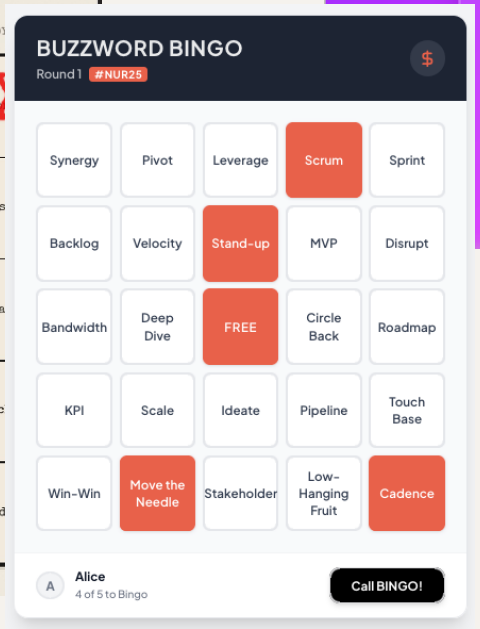
FOUNDATION

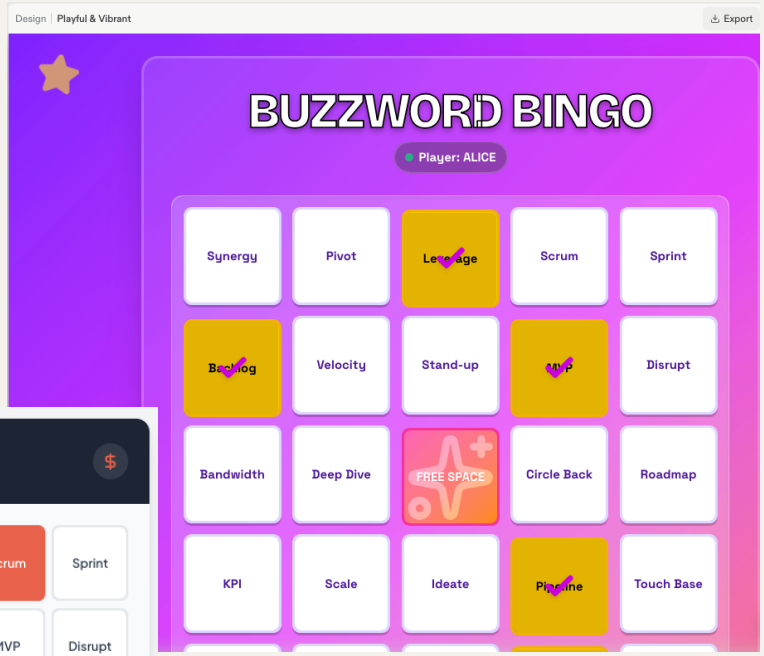
UI Sketches

The big ideas in a consistent wrapper









NOW WE BUILD

Live Coding

Less talking. More building.

>_ agile buzzword bingo · on replit

AI Pair

Prompt clearly, iterate fast, question the output.

Agile Rhythm

Small stories, frequent check-ins, working software first.

Test It

Automated & BDD where it counts. Manual to feel it. AI built-in.

Ship It

If it runs, it counts. Deploy now, refine after.

UI SKETCH

Cheat Sheet v1

Something we can feed to Replit

Replit built a Bingo Game
(At least the 5x5 card)
GRRRRRRRRR

```
You are an expert pragmatic agile developer. Focus on simplicity,
maintainability, and delivering value fast. Avoid over-engineering.

Project context: We are building a fun Agile Buzzword Bingo web app for a
business workshop.

Tech: Set up a complete, simple mobile-ready web app for Agile Buzzword Bingo.
Use [Node/Express + EJS, for example]. Include Tailwind CSS for quick nice
styling. Have a homepage that generates a 5x5 card. Make it deployable
publicly. Keep it minimal and easy to extend. Include a BDD testing framework
like playwright-bdd so we can demonstrate writing BDD style tests first.

Just take the project to the point where you have a "Hello World" generated
from a Cucumber style test:

Given I open the web app
When I enter my user "name"
Then I should see "Hello <name>"
```

UI SKETCH

Cheat Sheet v2

Something we can feed to Replit

You are an expert pragmatic agile developer. Focus on simplicity, maintainability, and delivering value fast. Avoid over-engineering.
Project context: We are building a fun Agile Buzzword Bingo web app for a business workshop.
Tech: Set up a complete, simple mobile-ready web app for Agile Buzzword Bingo. We will want to have nice styling. Make it deployable publicly. Keep it minimal and easy to extend. Include a BDD testing framework like playwright-bdd so we can demonstrate writing BDD style tests first.
Set this up so we have an EMPTY shell of a working project that we can then vibe code together as a workshop project. Just take the project to the point where you have a "Hello World" Cucumber style test that is FAILING:

```
Scenario: Users must enter their name
Given I open the web app
When I enter my user "name"
Then I should see "Hello <name>"
```

DO NOT BUILD THE "Hello World" feature AT ALL.
DO NOT BUILD THE BINGO GAME AT ALL.
Success is a failing test in the Console (or Shell)
That will leave us poised and READY to start building.

VIBE CODING

First Pass

The proverbial Hello Alice!



Feature: User greeting

Scenario: Users must enter their name
Given I open the web app
When I enter my user name "Alice"
Then I should see "Hello Alice"

VIBE CODING

Next Step

Some features...



Scenario: Create a Game
  Given I open the web app
  And I enter my user name "Alice"
  When I Start a Game called "LeSS NYC Bingo"
  Then I should see the "LeSS NYC Bingo" Game has been started

VIBE CODING

BDD Test results

Running the test harness...

```
Feature: User greeting # features/hello.feature:1

  Scenario: Users must enter their name # features/hello.feature:3
    Given I open the web app
    When I enter my user name "Alice"
    Then I should see "Hello Alice"

  Scenario: Create a Game # features/hello.feature:8
    Given I open the web app
    And I enter my user name "Alice"
    When I Start a Game called "LeSS NYC Bingo"
    Then I should see the "LeSS NYC Bingo" Game has been started

2 scenarios (2 passed)
7 steps (7 passed)
0m02.232s (executing steps: 0m02.206s)
```

VIBE CODING

Initial Deploy

Two features...

- Enter Name
- Create Game



VIBE CODING

And so on

Continue walking through the Story Map features

Feature: Basic Bingo

Background:
Given I open the web app

Scenario: Users must enter their name
When I enter my user name "Bob"
Then I should see "Hello Bob"

Scenario: Create a card
Given I enter my user name "Alice"
When I generate a card
Then I should see an empty Bingo card

Scenario: Start a game
Given Alice has generated a Bingo Card

```
| 1  | 2  |  3   | 4  | 5  |
+----+----+------+----+----+
| 6  | 7  |  8   | 9  | 10 |
| 11 | 12 | FREE | 13 | 14 |
| 15 | 16 |  17  | 18 | 19 |
| 20 | 21 |  22  | 23 | 24 |
```

When I start the game
Then the game should be started
And I should see a random number to see if I have a matching square

Scenario: Play a turn and match a tile
Given Alice has generated a Bingo Card

```
| 1  | 2  |  3   | 4  | 5  |
+----+----+------+----+----+
| 6  | 7  |  8   | 9  | 10 |
| 11 | 12 | FREE | 42 | 14 |
| 15 | 16 |  17  | 18 | 19 |
| 20 | 21 |  22  | 23 | 24 |
```

And the Bingo Card has the value "42" on one of the tiles
And I start the game
And the game should be started
And the number 42 is drawn by the system
When I click on the square with 42
Then it shows that it is matched

UI SKETCH

Advanced Concepts

Something we can challenge Replit to build when it makes sense

- Conduct a compliance/security audit and store the findings as part of the Super Admin portion of the web app. Employ a Kanban Board.
- Wire in code coverage and display results in the Super Admin view. Track quality over time and highlight trouble spots
- Implement smoke tests to be run all the time when making code changes
- Wire in code test checks to the deployment pipeline
- Build in other agents to monitor specific concerns

I have done most of these on production apps...

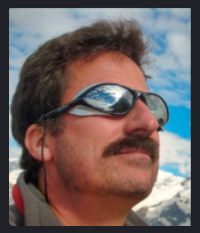
WRAP-UP

KEY LESSONS · RETRO

Vision first. Small stories. Ship early, ship often.

THE TAKEAWAY

Try this with your team next week.



Jon Kern

SCAN TO CONNECT ON LINKEDIN

RESOURCES

Cheat sheet · Public repo & live link · Connect with me on LinkedIn