

COMMUNITY TECH TALK • 45 MIN

# From Business Focus to Makers, *a boilerplate so you finally ship.*

How business analysts and product people can turn the requirements they already write into working software with agentic development, and the boilerplate that makes it repeatable.

## THE PROBLEM

How many hours last week went into explaining to developers what stakeholders want, waiting for the next sprint to add "one more filter", or rewriting the same Jira ticket three ways?

Great ideas die in backlogs waiting for developer time. The people who know best what to build are the furthest from building it.

## FOR

**Business analysts, product managers and product owners**

## STACK

Astro

Tailwind CSS

LangChain

AI coding agents

## WHAT THE TALK COVERS

### 01 Why, what, how

The BRD is the why, the PRD the what, the FRD the how. Business analysts already write the first two, and those are exactly what an AI coding agent needs.

### 02 Agentic development

Agents abstract the "how to": with a clear why and what, you can get the code running and validate an idea before it reaches a developer.

### 03 Agentic grooming

Using AI to refine stack choices: framework, styling, backend, auth, analytics and forms, decided in a table before a line of code.

### 04 Intent over syntax

Define the why and what, provide the context, let the agent handle the implementation. Your documentation becomes the prompt.

### 05 The proof

A database Q&A interface (LangChain, PostgreSQL, Vite) built with this pattern in about 40 minutes, one meeting's worth of time.

### 06 The architecture

Data layer, semantic layer and agentic layer: where the business logic lives, and where LangChain turns questions into SQL.

## KEY TAKEAWAYS

- The development team builds the engine; the business analyst provides the map. Agentic development removes the friction between the two.
- The question shifts from "How do I code this?" to "What should be built?"
- Choose one stack, master it, ship with it. A boilerplate turns one success into a repeatable one.
- Just start with a prompt: in this race you only lose if you don't try.

## Watch the companion video

[www.youtube.com/watch?v=4xQVUw5YirE](https://www.youtube.com/watch?v=4xQVUw5YirE)

## Read the write-up

[jalcocert.github.io/JAlcocerT/a-diy-boilerplate-to-ship](https://jalcocert.github.io/JAlcocerT/a-diy-boilerplate-to-ship)

## The landing boilerplate

[jalcocert.github.io/JAlcocerT/diy-landing-boilerplate](https://jalcocert.github.io/JAlcocerT/diy-landing-boilerplate)

## Want this in *your stack*?

A 30-minute decision call turns this pattern into a plan for your team.

[consulting.jalcocertech.com](https://consulting.jalcocertech.com)