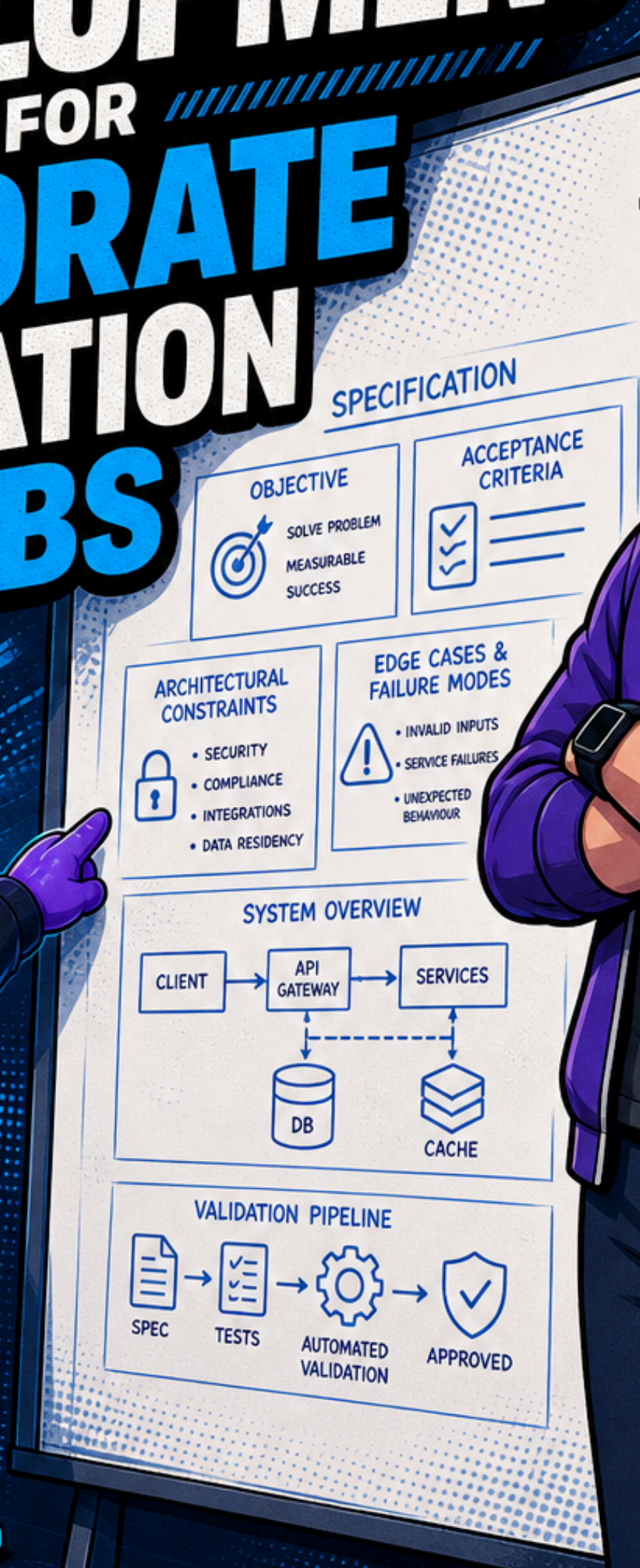


SPEC DRIVEN DEVELOPMENT

FOR CORPORATE INNOVATION LABS



WHAT IS SPEC DRIVEN DEVELOPMENT, AND WHY DOES YOUR INNOVATION LAB **NEED IT NOW?**



Most corporate innovation labs run fast. They're under pressure to produce a demo-ready prototype before the next board review, and the instinct is to start building immediately. Get a developer in the room, open a Figma file, start prompting an AI coding tool, and see what comes out.



The problem is that speed without structure produces the wrong thing faster. And in a corporate environment, building the wrong thing doesn't just waste sprint capacity. It creates rework, compliance exposure, and the kind of credibility damage that makes the next funding conversation harder.



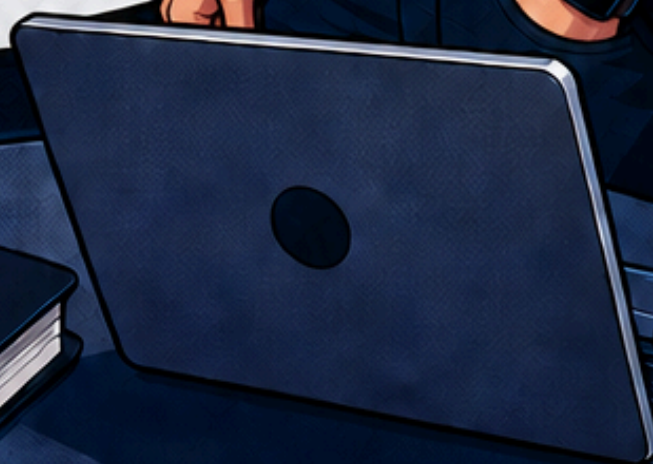
Spec driven development (SDD) is a methodology that flips this sequence.

“ It inverts the traditional relationship between intent and implementation. Instead of writing code and hoping the intent can be reconstructed from it later, SDD insists that intent is captured first, completely, unambiguously, and in a form that outlives any individual codebase.

That sounds simple. In practice, it changes almost everything about how innovation labs should run.

CLARITY
BEFORE
CODE

INTENT
FIRST



START WITH THE SPECIFICATION, NOT THE PROTOTYPE

SPEC FIRST.
ALIGN ALWAYS.
DELIVER
CONFIDENTLY.



THE SPEC: EXECUTABLE OPERATIONAL CONTRACT



OBJECTIVE & CONTEXT

What problem are we solving, for whom, and what does success look like?



ACCEPTANCE CRITERIA

What must be true for this feature or system to be considered done?
Written before a single line of code.



ARCHITECTURAL CONSTRAINTS

Security standards, integration patterns, data residency, compliance obligations – from day one.



EDGE CASES & FAILURE MODES

What happens when inputs are wrong, integrations fail, or users behave unexpectedly?

Confluence

Requirements.docx

- Feature overview
- Stakeholder inputs
- High level flows
- Assumptions
- Questions

OUTDATED

THE DIFFERENCE THAT DELIVERS

TRADITIONAL SPEC

- ✗ Written, approved, ignored
- ✗ Drifts by Sprint 3
- ✗ Outdated by Release 2
- ✗ Doesn't match the product



SDD SPEC

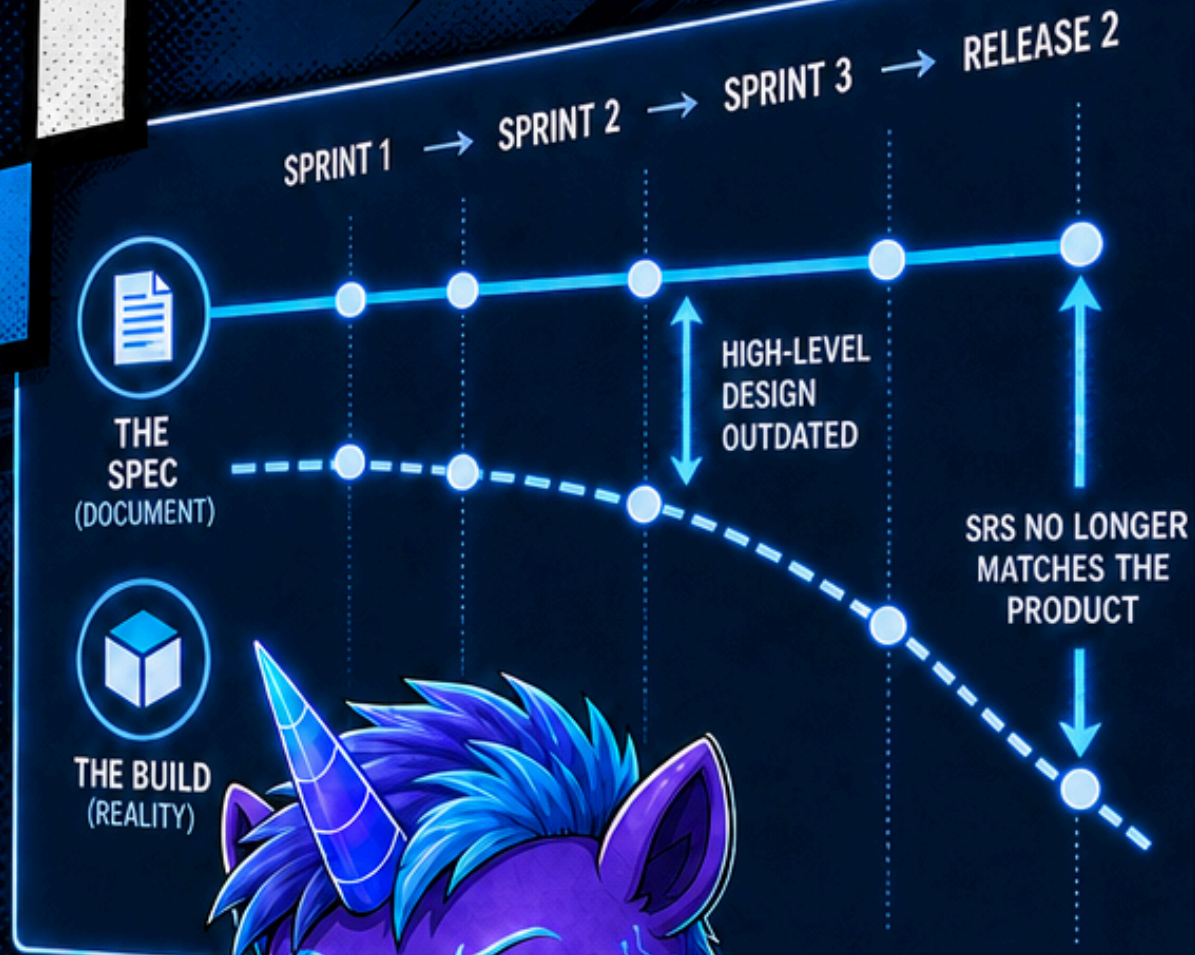
- ✓ Living, executable contract
- ✓ Aligned across humans & AI
- ✓ Stays current, always relevant
- ✓ Single source of truth



CLARITY BEFORE CODE IS THE MOST PERSISTENT CONSTRAINT
IN SOFTWARE DELIVERY.

STOP WRITING FICTION. START WRITING CONTRACTS.

THE DRIIFT PROBLEM



A traditional spec gets written, approved, then ignored as the build diverges.

The problem is not the absence of specs — it's that they **drift**.



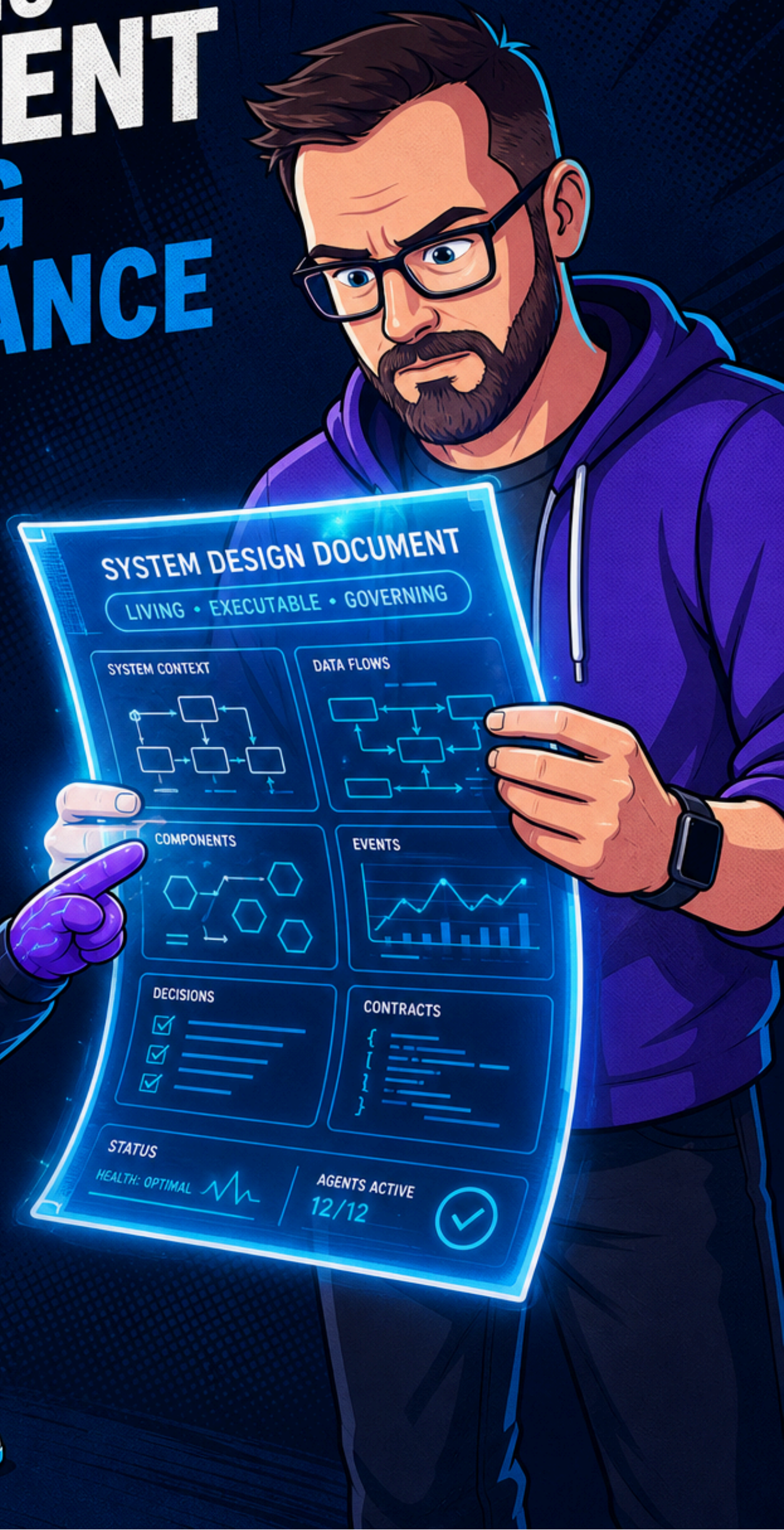
By **Sprint 3**, the high-level design is outdated.



By **release 2**, the SRS no longer matches the product.

FROM STATIC DOCUMENT TO LIVING GOVERNANCE

The specification has ceased to be a boring static PDF. In a mature SDD practice, it functions as an **executable operational contract** — actively governing the system and serving as the **persistent memory** both humans and AI agents need to maintain alignment.



FOUR ELEMENTS EVERY SDD SPEC INCLUDES



1. OBJECTIVE & BUSINESS CONTEXT

- problem
- audience
- measurable success



2. ACCEPTANCE CRITERIA

- written before any code exists



3. ARCHITECTURAL CONSTRAINTS

- security
- compliance
- integration patterns



4. EDGE CASES & FAILURE MODES

- wrong inputs
- failed integrations
- unexpected users

ALIGN YOUR STAKEHOLDERS

BEFORE THE FIRST SPRINT, NOT AFTER

SHARED SPECIFICATION



OBJECTIVE

Enable secure customer onboarding with automated identity verification.



BUSINESS CONTEXT

- Improve onboarding conversion
- Reduce manual review workload



CONSTRAINTS

- Comply with NZ Privacy Act
- Data residency in NZ
- Integrate with existing IdP



SUCCESS CRITERIA

- 90% verification within 60 seconds
- < 1% false reject rate
- Audit logs for all decisions

STAKEHOLDERS

- Product
- Engineering
- Risk & Compliance
- Security
- Legal

ALIGN INTENT.
REDUCE RISK.
DELIVER WITH CONFIDENCE.





THE SPEC AS SINGLE SOURCE OF TRUTH FOR GOVERNANCE AND ONBOARDING



GOVERNANCE BY DESIGN

Spec-anchored development adds governance layers, constitutional constraints, and supervision checkpoints.



EVIDENCE FROM DAY ONE

A well-structured specification becomes the evidence layer for audit trails, compliance, and decision-making.



BUILT-IN AUDITABILITY

Record decisions, constraints, criteria, and approvals. Not bureaucracy. Shorter audits. Stronger defense.



ONBOARDING THAT SCALES

The spec captures why, what, and what would break. Version control for your thinking.

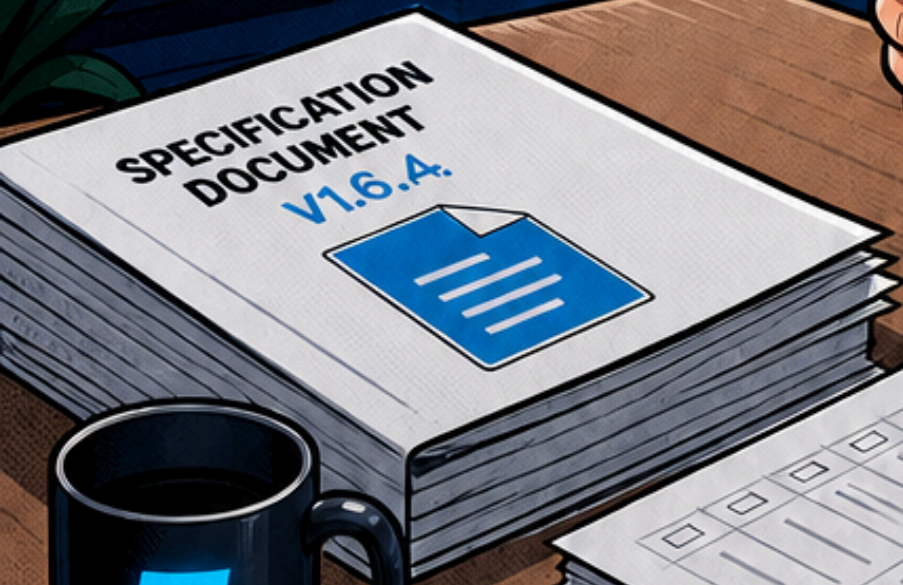


AUDIT TRAIL - SPEC EVIDENCE

DECISION	APPROVED BY	DATE	REFERENCE
<input checked="" type="checkbox"/>			
<input checked="" type="checkbox"/>			

CHANGE LOG & APPROVALS

DECISION	APPROVED BY	DATE	REFERENCE
<input type="checkbox"/>			
<input type="checkbox"/>			
<input type="checkbox"/>			



DERIVING AUTOMATED TESTS DIRECTLY FROM THE SPECIFICATION

SPECIFICATION

ACCEPTANCE CRITERIA

- ✓ User can sign up with email.
- ✓ Password must be at least 8 characters.
- ✓ Email must be unique.
- ✓ Account is locked after 5 failed login attempts.

Audit log is recorded for all authentication events.

AUTOMATED TEST CASES

- ✓ TC-001 Sign up with valid email → Success
- ✓ TC-002 Sign up with invalid email → Error
- ✓ TC-003 Password less than 8 chars → Error
- ✓ TC-004 Duplicate email → Error
- ✓ TC-005 Lock account after 5 failed attempts
- ✓ TC-006 Audit log recorded on login success
- ✓ TC-007 Audit log recorded on login failure



EXECUTABLE SPECIFICATIONS.
AUTOMATED TESTS.
QUALITY FROM DAY ONE.



ACCELERATING HANDOFFS TO PRODUCTION TEAMS



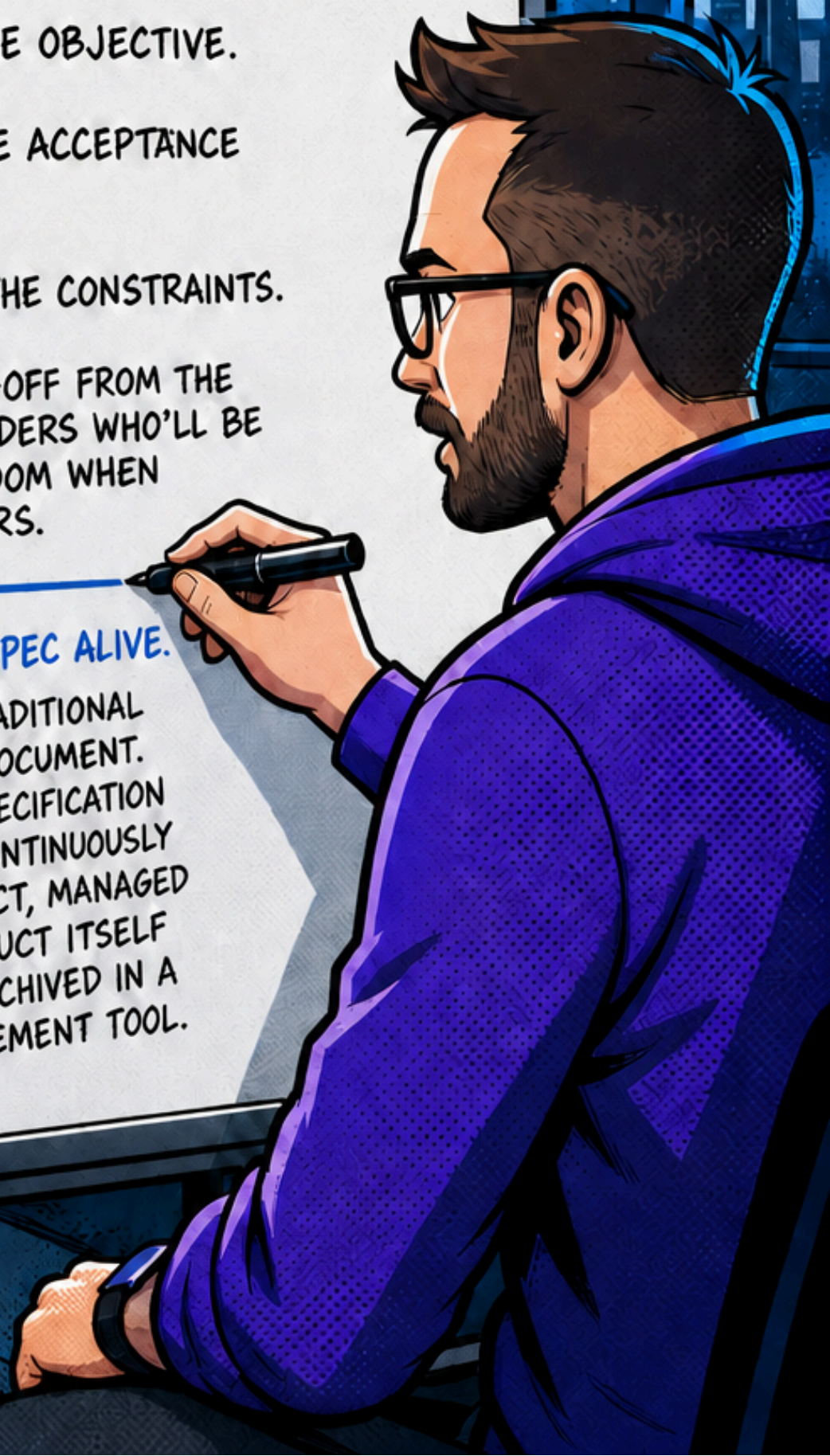
A PRACTICAL NOTE ON GETTING STARTED

BEFORE ANY CODE EDITOR OR AI CODING TOOL IS OPENED:

- ☑ DEFINE THE OBJECTIVE.
- ☑ WRITE THE ACCEPTANCE CRITERIA.
- ☑ RECORD THE CONSTRAINTS.
- ☑ GET SIGN-OFF FROM THE STAKEHOLDERS WHO'LL BE IN THE ROOM WHEN IT MATTERS.

THEN KEEP THE SPEC ALIVE.

THIS IS NOT A TRADITIONAL REQUIREMENTS DOCUMENT. IT IS A LIVING SPECIFICATION THAT EVOLVES CONTINUOUSLY WITH THE PRODUCT, MANAGED INSIDE THE PRODUCT ITSELF RATHER THAN ARCHIVED IN A PROJECT MANAGEMENT TOOL.



THE BOTTOM LINE

CORPORATE INNOVATION LABS ARE UNDER REAL PRESSURE IN 2026. SHORTER TIMELINES, HIGHER SCRUTINY, AND AN INCREASINGLY COMPLEX AI GOVERNANCE ENVIRONMENT MEAN THAT THE "BUILD FAST AND SEE" APPROACH CARRIES MORE RISK THAN IT USED TO.

SPEC DRIVEN DEVELOPMENT DOESN'T SLOW YOU DOWN. IT REMOVES THE MOST EXPENSIVE KIND OF REWORK: THE KIND THAT HAPPENS AFTER YOU'VE ALREADY SHOWN THE PROTOTYPE TO THE WRONG PERSON. ONE AGREED-UPON SPECIFICATION DOCUMENT, WRITTEN BEFORE THE FIRST SPRINT, ALIGNS YOUR STAKEHOLDERS, GOVERNS YOUR BUILD, GENERATES YOUR TESTS, AND MAKES YOUR HANDOFFS CREDIBLE.

THAT'S NOT A PROCESS OVERHEAD. THAT'S HOW YOU IMPROVE YOUR CHANCES OF PUTTING SOMETHING IN FRONT OF THE BOARD IN SIX WEEKS AND HAVING IT HOLD UP UNDER SCROTINY.



SPECIFICATION DOCUMENT

- ✓ STAKEHOLDER ALIGNMENT
- ✓ BUILD GOVERNANCE
- ✓ TEST GENERATION
- ✓ CREDIBLE HANDOFFS

E EVOTRON STUDIO

SPRINT PLAN

DISCOVERY
↓
DESIGN
↓
BUILD
↓
TEST
↓
LAUNCH

STAKEHOLDER SIGN-OFF

- CEO ✓
- CFO ✓
- CTO ✓
- HEAD OF RISK ✓
- LEGAL ✓
- PRODUCT ✓

TIMELINE: 6 WEEKS

BOARD DEMO ★

SUCCESS CRITERIA

- VIABILITY
- COMPLIANCE
- USABILITY
- IMPACT

RISKS & MITIGATIONS

- AI GOVERNANCE
- DATA PRIVACY
- MODEL RISK
- BIAS & FAIRNESS



EVOTRON STUDIO PAIRS SENIOR KIWI OPERATORS WITH AN AGENTIC PLATFORM TO STAND UP PRODUCTS, BRANDS, AND GO-TO-MARKET IN WEEKS. IF YOUR LAB IS LOOKING FOR A SENIOR OPERATOR WHO BUILDS TO SPEC FROM SPRINT ONE, [LET'S TALK.](#)

EVOTRON STUDIO
AGENCY AGENT SYSTEM

```
vo = {  
  "agentic_workhorse",  
  "always_on",  
  ts: 12,  
  ise: "invisible tech",  
  it: "visible results"
```

```
ship(value) {  
  ders.focus();  
  execute();  
  n results.delivered();
```

TYPE. JUST OUTCOMES.
EVOTRON STUDIO



<https://evotronstudio.co.nz>