

From Idea To Launch

Agentic Programming with Spec-Driven Development

One-day immersive practical workshop



Learning Objectives

- Understand the difference between agentic programming and traditional vibe coding.
- Practise end-to-end agentic programming with spec-driven development.
- Master the four core pillars of SDD: rules, skills, workflows, and knowledge.
- Explore future directions for enterprise-scale IT organisations and professionals.

Vibe Coding Experience:
Hello World



Adding new features
using Vibe Coding



Limitations of Vibe Coding
at enterprise scale



SDD Introduction:
Plan - Spec - Code - Verify
Learn & Iteration



Deliver a complete product
from idea to product



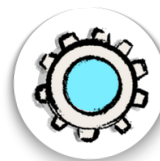
Learnings & Future
outlook



SDD Pillar 4: Build
technical and
procedural expertise
through “Knowledge”



SDD Pillar 3: Define a
software craftsmanship
‘Workflow’ for you and your
AI



SDD pillar 2: Use “Skills” to
transform AI into a full-stack
team.



SDD pillar 1: Use “Rules” to
shape principles and AI’s
personality

Outline

Vibe Coding Experience: Hello World

- No software background? No problem - Vibe Coding is for everyone.
- Set up the tools and environment, then run your first project: Hello World.

Limitations of Vibe Coding at Enterprise scale

- Exercise: Use Vibe Coding to develop new features based on requirements written in natural language.
- Discussion: What challenges would you face if you were to take over this work from the AI?

Introduction to Spec Driven Development

- AI-assisted programming, fully controlled and seamlessly integrated with agentic coding.
- Plan – Spec – Code – Verify.
- Learn & Iterate: You and AI learn from each other before moving to the next feature.
- Four pillars: skills, rules, workflow, knowledge.

SDD Pillar 1: Use rules to shape AI's personality and align it with your core principles

- Exercise: Teaching AI the “incremental delivery” practice
- Discussion: How to avoid reteaching AI each time
- Exercise: Writing a new rule, such as DoD, Testing Strategy, UI/UX principles, etc

SDD pillar 2: Use “Skills” to transform AI into a full-stack team.

- Exercise: Train AI to master ATDD and help you write high-quality user stories and acceptance criteria
- Exercise: My dream team – what does your ideal full-stack team look like?
- Exercise: Leverage your skills to transform your AI into a full-stack team

SDD Pillar 3: Define a software craftsmanship ‘workflow’ for you and your AI

- Exercise: Complete the development of the first feature
- Discussion: How can AI operate autonomously?
- Exercise: Design your own workflow
- Discussion: How can AI be made to enforce rules strictly?

SDD Pillar 4: Build technical and procedural expertise through “Knowledge”

- Discussion: Have you experienced frustration with AI's limited short-term memory?
- Introduction: ADR mechanism
- Exercise: Write a Retrospective skill and have people and AI hold a retrospective meeting together

Final exercise: 30 minutes, complete delivery of an application software

- Deliver the first MVP of your own application using the current SDD framework
- introduce openspec framework if needed

Learning review and future prospects

- What have we learned today?
- What is the key takeaway?
- What are your insights into the future direction of IT organisations and professionals?