

# **10 Experiments - 10 Learnings**

**Picking interesting examples from 200 prototypes created with ChatGPT, Claude, Gemini and Windsurf**

# Disclaimer

## No humans were harmed in this talk

- LLM Services used, not limited to: Claude, Gemini, ChatGPT, Lovable, Replit, Rosebud.ai, Udio, Suno, Veo 3, Sora.
- Price of Admission (24 Months) - 2000 Euro plus Tax
- 99% of this talk is my free time with my resources. I wanted ethics to be my limiting factor, not free tiers of services or other rules.
- Released content to: Github, NPM, Youtube, PyPi
- New Friends: 10
- Next: Unhindered and Unhinged Exploration and Experimentation

# Why so interested in that all?

- Coding with a LLM is in its core a form of generative art
  - As a VJ, I already was paying rent with it in 2005
- The Software/Value Discussion was part of my work as a Team Coach/Scrum Master
- I can do arts, coding and the requirements engineering part: LLM makes me 'a multi-modal-media one man army.
- It brought me places I was not supposed to go (FPGAs, DSPs) - i never was smart enough for this - so I thought!

# Timeline

- End of 2022: The beginning of the LLM craze
- Start of 2023: I start my position at Maiborn Wolff
- Middle 2023: I was like **holy sh\*\***
- Ending 2023: Prompting
- Ending: 2024 Writing Software with LLMs (learned about DSPs)
- **Now: Writing better Software with much greater scope than in 2021**
  - Learned FPGAs are hard. Basically step 1. ;)

# Why experiment and not 'release'

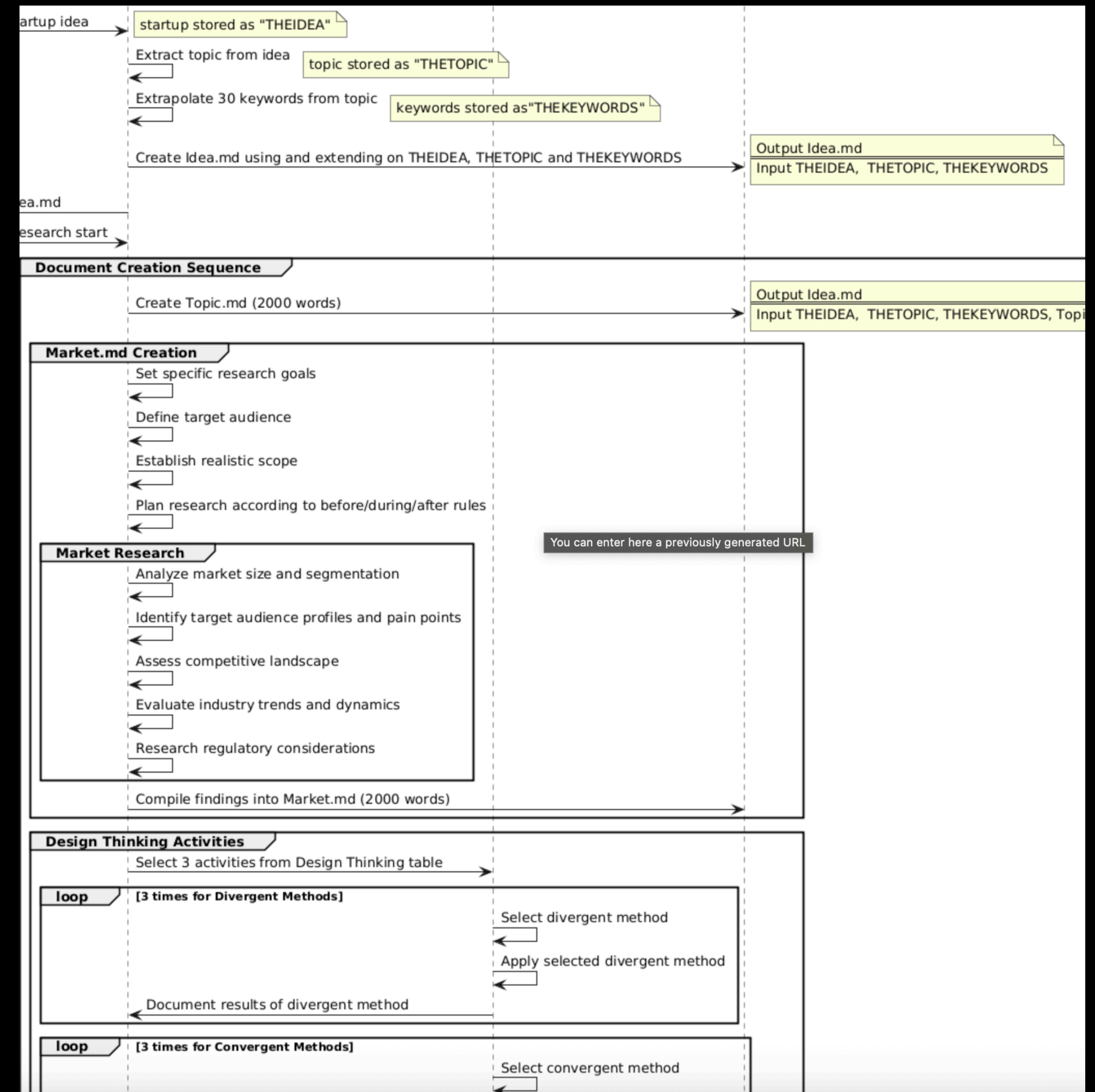
- Diminishing returns
- My time, my resources: I do what brings me joy.
- Assumptions: Things develop exponential - no need to move in!
- This is about broadening my horizon - not primarily being more productive

# Rapid Ideation

# Rapid Ideation

## The Prototype

- Not a Application, but a set of LLM instructions
- Validating ideas/requirements
- Question: How do we get from Idea to Backlog of > 100 Userstories in under 1 hour  
Refined and completed Userstories ca. 4 hours?



- **Topic.md:** Extract the underlying topic. Do a deep research into the topic itself (not its marketability). ca. 2000 Words
- **Market.md:** Based on the research in topic and the idea: outline of market conditions. Do a market analysis. 2000 words.
- **Design Thinking:** Pick 3 activities from the table in "Design Thinking Activities" and select at least 3 methods for diverging and 3 for converging. Simulate the activities using the '3 amigos team' and add the comprehensive results (quantitative and qualitative) for each activity to design-thinking.md
- **Vision.md:** Write a vision statement for the project. 200 words. What problem are we solving for whom?
- **Features.csv:** What user facing features do we need specifically for this project. columns: feature\_id, feature\_name\_, topic

You are my startup researcher/product designer. Help me, turn a idea into a set of documents with actionable information.

First: Ask me my startup idea.

When I responded with my idea, take the instructions in this document and the idea to help me produce the following files described below.

State the topic you extracted from my idea. List the file names you can produce and wait for my confirmation to start. Default is from top to bottom in the list.

project columns: feature\_id, feature\_name\_, topic

- **Architecture.mmd:** High level Architecture Diagram mermaid representing how the features interact
- **Prototype.md:** Generate prompts representing 3 distinct features. I will use the service rosebud.ai to generate prototypes.
- **Stories.csv:** Split features into smaller stories. Estimate each for value, penalty, effort, risk with fibonacci numbers. columns: id, feature\_id, topic, value, penalty, effort, risk, StoryText
- **Link.md:** List all content from google searches and other relevant sources to the topic.

# Rapid Ideation

## Learnings & Insights

- Evaluate your ideas and improve on them
- Ideation can be found at the heart of service design
- Diverging Processes and Methods: create more data
  - Example: Brainstorming - Collecting Ideas from everyone
- Converging Processes: Create a smaller amount of information out of data
  - Example: Cluster Ideas from Brainstorming

**Code, Learn, Delete, Repeat**

# Project Management Tool

## The Prototype(s)

- Support writing good user stories
- Help finding 'critical paths' in large sets of requirements
- Relative Weight Estimation (Benefit, Penalty, Effort, Risk)
- Question: How is estimation done in agentic development workflows

The screenshot displays two main components of the tool's interface:

### CSV Data Viewer

Current file: Stories.csv

Buttons: Import CSV, Export CSV

Search table...

Show/Hide Columns:  id  feature\_id  topic  value  penalty  effort  risk  UserStoryText

id	feature_id	topic	value	penalty	effort	risk	UserStoryText
S001	F01	Engine Core	13	21	8	5	As a framework user (developer), I want a reliable core game loop manager that consistently triggers game state updates based on elapsed time, so I can implement game logic without managing 'requestAnimationFrame' or 'setTimeout' timing complexities myself.
S002	F02	Game Mechanics	8	13	3	3	As a framework user (developer), I want a mechanism to define the resource types used in my game (e.g., 'LinesOfCode', 'Bugs'), specifying at least their unique identifier and starting amount, so I can easily declare the core currencies/materials. *(Note: Primary mechanism likely via F12 Data-Driven Config; API might be for runtime additions if needed, TBD)*

PREVIEW ASSETS CODE

### User Story Prioritization

Comparison 1 of 10 10%

#### Item A

##### User Authentication System

As a user, I want to securely log in to the application so that I can access my personal dashboard.

8 points

#### Item B

##### Search Functionality

As a user, I want to search for content so that I can quickly find relevant information.

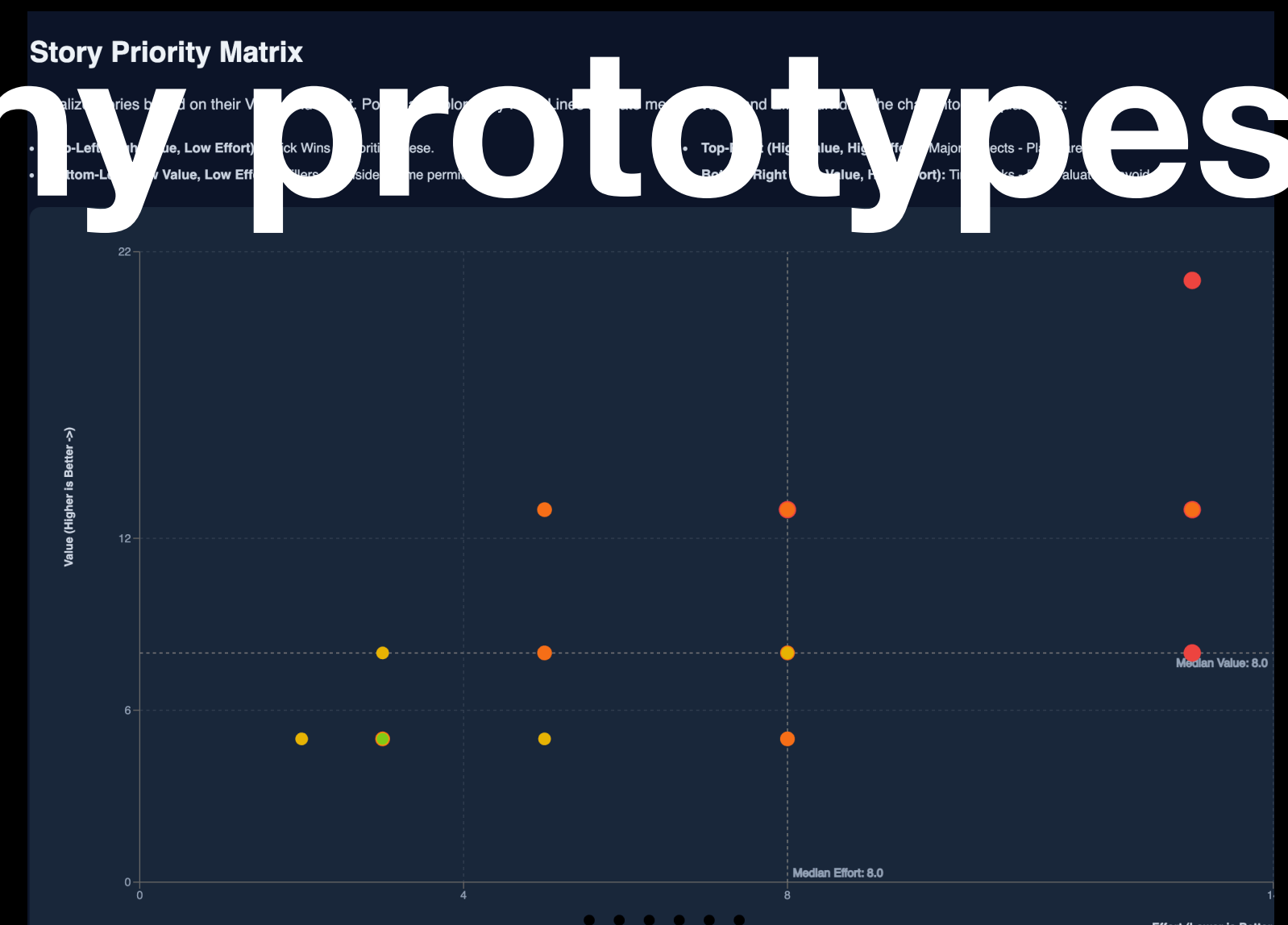
5 points

Item A is more important Equally important Item B is more important

# This problem needed many prototypes

The screenshot shows a project management interface. On the left, there's a 'Stories List' with three items, each with an effort value and an 'Enhanced' status. The main area shows the 'Details' of an ADR titled 'ADR-001: Task Decomposition AI Architecture'. It includes sections for 'Context', 'Decision', and 'Consequences'. The 'Decision' section states: 'We will implement a hybrid ML architecture combining NLP for text analysis and graph-based algorithms for dependency mapping. The system will use historical decomposition patterns and integrate with existing project data.' The 'Consequences' section lists: 'Positive: Improved decomposition quality, consistent planning, transferable knowledge. Negative: Initial training period required, need for feedback loop implementation.'

The dashboard for 'NexusFlow Project Orchestration' features a sidebar with navigation options like 'Dashboard', 'Kanban Board', 'User Stories', 'ADRs', 'Metrics', and 'Priority Matrix'. The main area displays three key metrics: 'Active Stories' (0, +3 from last week), 'Completed' (0, +8 this month), and 'Avg Cycle Time' (0d, -0.8d improvement). Below these are 'Recent User Stories' with descriptions and a 'Quick Actions' panel with options like 'New User Story' and 'Document ADR'. A 'Workflow Status' section shows 'Backlog' with 200 items.



# Code, Learn, Delete, Repeat

- take what you learned
- start from scratch
- reduce errors drastically
- keep: decision records, concepts, research results
- delete: most of the code and be happy about it. Then: Start again.
- extract core libraries and critical implementations

**A markdown document is fine!**

# markdown.md

## The Prototype(s)

- How do I get to a state of: well tested, well documented and hopefully not over engineered working software?
- **By holding back to start developing and make sense of what to build and why in Text form**
- Iterating concepts is cheap
- Less 'sunk cost fallacy'

### public-code-tools

#### adr

- ↓ ADR-003-use-commander-cli-framework.md
- ↓ ADR-004-offline-first-architecture.md
- ↓ ADR-005-table-based-output-multiple-formats.md
- ↓ ADR-006-retry-logic-exponential-backoff.md
- ↓ ADR-007-normalized-database-schema.md
- ↓ ADR-008-porter-stemming-fts.md
- ↓ ADR-009-content-hash-change-detection.md
- ↓ ADR-010-similarity-algorithm.md
- ↓ ADR-011-bulk-import-concurrent-queue.md
- ↓ ADR-012-store-raw-yaml.md
- ↓ ADR-013-configuration-strategy.md
- ↓ ADR-014-error-hierarchy.md
- ↓ ADR-015-database-triggers-fts-maintenance.md
- ↓ ADR-016-typescript-compiler-for-code-quality.md
- ↓ ADR-017-no-full-text-search.md

#### cli-app

#### docs

- ↓ additional-publiccode-yml-parser-features.md
- publiccode\_schema.sql
- ↓ publiccode-cli-command-interface-specification.md
- ↓ publiccode-cli-comprehensive-indexing-and-search-tool.md
- ↓ publiccode-cli-technical-architecture.md
- ↓ publiccode-yml-database-design-analysis.md
- ↓ publiccode-yml-parser-analysis.md
- ↓ publiccode-yml-research.md
- ↓ publiccode-yml-sqlite-database-implementation-guide.md
- ↓ publiccode-yml-sqlite-database-schema-design.md
- ↓ requirements.md

Code assistant

- types.ts
- data.ts
- character-logic.ts
- dungeon-generator.ts
- renderer.ts
- ui.ts
- input-handler.ts
- game.ts
- logic/player-actions.ts
- logic/turn-manager.ts
- logic/dungeon-logic.ts
- logic/item-logic.ts
- ui/ui-manager.ts
- ui/draw-character-screens.ts
- ui/draw-dungeon-screen.ts
- ui/draw-inventory-screens.ts
- ui/draw-spellbook-screen.ts
- logic/dungeon-generation/dungeon-types.ts
- logic/dungeon-generation/utils.ts
- logic/dungeon-generation/geometry.ts

Analyzed errors for 70 seconds

Thought for 611 seconds

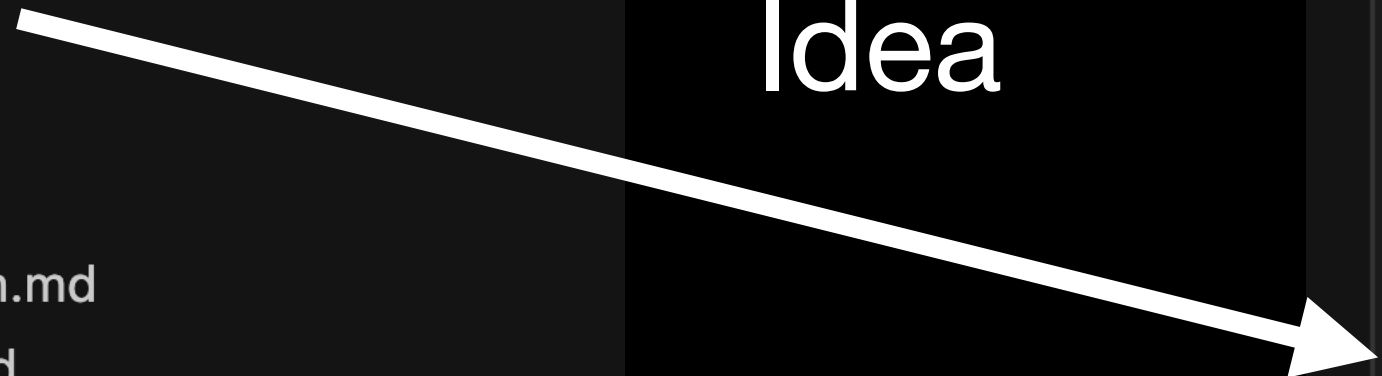
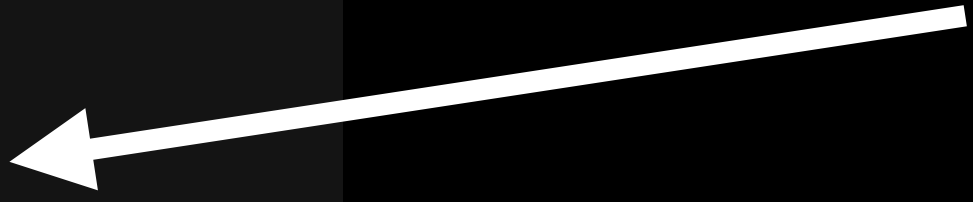
- > p5js-playground
- > procedural-3d-world-generator
- > quest-generator
- > random-dungeon-generator (2)
- > webgl-dungeon-crawler
- > WorldCraftAI

Prototype

- research
  - > Sangband
  - 01\_character\_systems.md
  - ~/private/dungeon-crawler/research/01\_character\_system
  - 03\_talent\_system.md
  - 04\_combat\_system.md
  - 05\_magic\_system.md
  - 06\_item\_system.md
  - 07\_monster\_system.md
  - 08\_dungeon\_generation.md
  - 09\_crafting\_systems.md
  - 10\_status\_effects.md
  - 11\_other\_systems.md
  - 12\_store-system.md
  - 13\_trap\_calculations.md
  - 13\_trap-system.md
  - systems.md
  - webgl-dungeon.md

Idea

Analysis



# Types of documents

## in your app directory

- `./research` - Results from different LLMs, web searches, Scientific Papers, very relevant documentation.
- `./adr` - Architecture Decision Records
- `./systems` or `./process` - A more structured way to look at how things work
- `./docs/` - concept, ui concept, what to build etc.
- `./requirements/` - multiple, linked files with brief requirements
- `./stories/` - Userstories aide the implementation steps and help us cross validate requirements and architecture decision records.

# Requirements Engineering FTW

# Robo Requirements Engineering

## The Prototype

- Help me find and document requirements.
- Worked: Gherkin, TCLR, User Stories, etc pp
- Question: How can we express requirements in a concise way for a LLM?

### Table of Contents

1. Introduction
2. Stakeholder Analysis
3. Functional Requirements
4. Non-Functional Requirements
5. Data Requirements
6. Interface Requirements
7. Architecture Requirements
8. Quality Attributes
9. Constraints and Dependencies
10. Risk Analysis
11. Requirements Traceability
12. Appendices

#### FR-3.2.2 Faceted Search [SD, GM]

The system shall support filtering by:

- Categories and tags
- Development status
- Software type
- Legal attributes (license, copyright)
- Geographic scope
- **Acceptance Criteria:** Apply multiple filters simultaneously with sub-second response
- **Priority:** High

#### FR-3.2.3 Similarity Search [SD]

The system shall find similar software based on features and descriptions.

- **Acceptance Criteria:** Return top 10 similar items using semantic analysis
- **Priority:** Medium

# Concept to Requirements

- There is a value in classical req. engineering
- Looking at activities is a good way to start
- Helps taking different viewpoints
- Shoutout: BMW Engineering with a Requirements DSL
- If you are missing a DSL: Self Assembly - Just build one

# Self Assembly

# Self Assembly

## The Prototype

- Self coined Term
- LLM tool use can execute thre whole workflow
- Plug in Systems or code gen for extensions. LLM in the application can propose extensions to itself.

### TRLC-AI: Requirements to TRLC Converter

Convert unstructured requirements documents into validated TRLC code using AI

 **Input Requirements**  Generated TRLC Code

Choose input method:

Text Input  File Upload

Enter your requirements:

Enter unstructured requirements, bullet points, meeting notes, etc.

Example:

- The system shall authenticate users within 2 seconds
- High priority: Password must be at least 8 characters
- User management should be handled by admin team

 Generate TRLC Code

TRLC-AI: Transforming unstructured requirements into validated TRLC code

Powered by Google Gemini AI • Built with Streamlit

# Self Assembly

## Learnings & Insights

- Self Coined term
- **Build the LLMs tool usage before Human features**
  - Helps avoiding building ui crust that you don't need afterwards.
- Feature Number 1: LLM integration
- **Later: Add ui for humans where needed and exactly as needed**
  - Not every process needs a human

# Hard Questions

- The code review has identified several critical security issues that need immediate attention:

#### **Critical Security Issues:**

1. **CRDT ID Generation** – Using `Math.random()` instead of cryptographically secure methods
2. **XSS Vulnerability** – Direct HTML injection in template rendering without sanitization
3. **Insecure Peer IDs** – Predictable peer IDs that could enable session hijacking

#### **Other Important Issues:**

- Missing input validation on remote operations (could allow malicious operations)
- Memory leaks in event listeners
- Race conditions in auto-save functionality

#### **Positive Findings:**

- Excellent architecture following vanilla JS requirements
- Sophisticated CRDT implementation for conflict-free collaboration
- Strong WebRTC mesh networking supporting multiple participants
- Comprehensive test coverage with TDD approach
- Good performance optimizations

The codebase demonstrates exceptional quality overall with an A- grade, but the security vulnerabilities must be fixed before production use.

The architecture successfully implements complex collaborative features while maintaining simplicity and avoiding frameworks.

# Hard Questions

- In order not to be constantly ashamed, humans have the capacity to avoid hard questions
- Let the LLM help you by asking exactly those questions and answer them.
- Code Review is a broad term - be more specific.
- Learn what to ask for. If you don't know what to ask for - you probably should not use a LLM for that.

# From Vibe Coding to Vibe Launching

BUS FACTOR

**1**

TOTAL FILES

**192**

TOTAL CONTRIBUTORS

**8**

CRITICAL CONTRIBUTORS

**1**

**Risk Level: CRITICAL**

**Analysis Message**  
Project has a bus factor of 1. Recent knowledge is concentrated in a single developer.

**Recommendation**  
Urgent action needed. Implement immediate knowledge transfer sessions and pair programming.

**Critical Contributors**

Darko Pervan

**Top 10 Contributors**

AUTHOR	WEIGHTED AUTHORSHIP	FILES OWNED	RECENT ACTIVITY
Darko Pervan	54.69%	105	23119



It supports two calculation methods:

### Authorship-Based Factor)

```
node lib/index.js abf <git-repo-url>
```

### Baylzade Bus Factor)

```
node lib/index.js jbf <git-repo-url>
```

### Options

- `--json` - Output results as JSON
- `--quiet` - Minimal output
- `--summary` - Summary only

### Examples

```
# Analyze with ABF method
node lib/index.js abf https://github.com/chalk/chalk.git

# Analyze with JBF method and JSON output
node lib/index.js jbf https://github.com/chalk/chalk.git --json

# Quiet mode - only show bus factor number
node lib/index.js abf https://github.com/chalk/chalk.git --quiet
```

More

Downlo

# Coding to Launching

- At some point, a launch makes sense to create feedback
- Start by creating mechanisms to process feedback automatically
  - mcp server with connection to logs
  - Post Mortem Documents about outages
  - Github Bugreports
- Decision making can be based on hard numbers from now on
  - provide tools to interpret the logs, not llm runtime

**Minimalism**

# Minimalism

## The Prototype

- I had examples, where  $> 50\%$  LLM time used for fixing lint and other formatting errors. About 400 rules.
- How many Wrappers/SDK to access a DB with a JSON interface?
- How much react to place a button?
- How many model files to describe what could be 1 Table, 3 Triggers and 2 Stored Procedures?
- Question: How can we do this with about 50% of the tokens?

```
{
  "name": "mindmap-app",
  "version": "1.0.0",
  "description": "",
  "type": "module",
  "main": "index.js",
  ▶ Debug
  "scripts": {
    "test": "NODE_ENV=test node --test tests/unit/**/*.test.js tests/integration/**/*.test.js tests/e2e/*.test.js",
    "test:unit": "NODE_ENV=test node --test tests/unit/**/*.test.js",
    "test:integration": "NODE_ENV=test node --test tests/integration/**/*.test.js",
    "test:e2e": "NODE_ENV=test node --test tests/e2e/*.test.js"
  },
  "author": "",
  "license": "ISC",
  "devDependencies": {
    "fake-indexeddb": "^6.0.1",
    "jsdom": "^26.1.0"
  }
}
```

# Embrace the 'Smol Stack'

## Solution: Minimal starter Templates

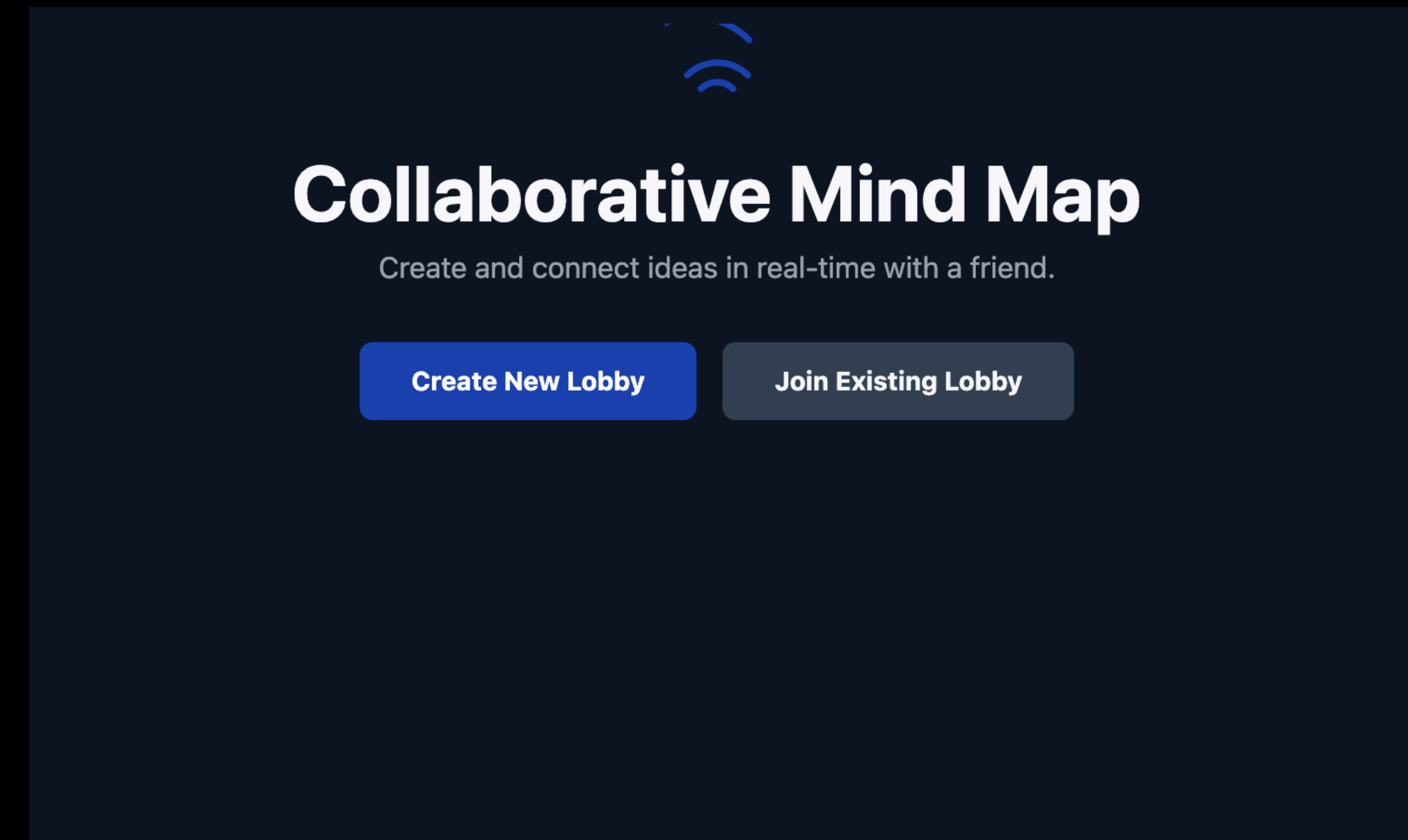
- No need for “Developer Experience” when the machine is the developer.
- Limit ‘rules’ to a amount that is humanly possible to understand
- Use strict
- no react
- html + js
- no build/no bundle
- no SPA

# Serious Vertical Slicing

# Vertical Slicing

## The Prototype

- Question: What if we started with all the hard technical challenges in Iteration 0?
- A collaborative mind map was a very challenging feature set
- 



# Collaborative Mindmap

Add Node

Save

Export

↶ Undo

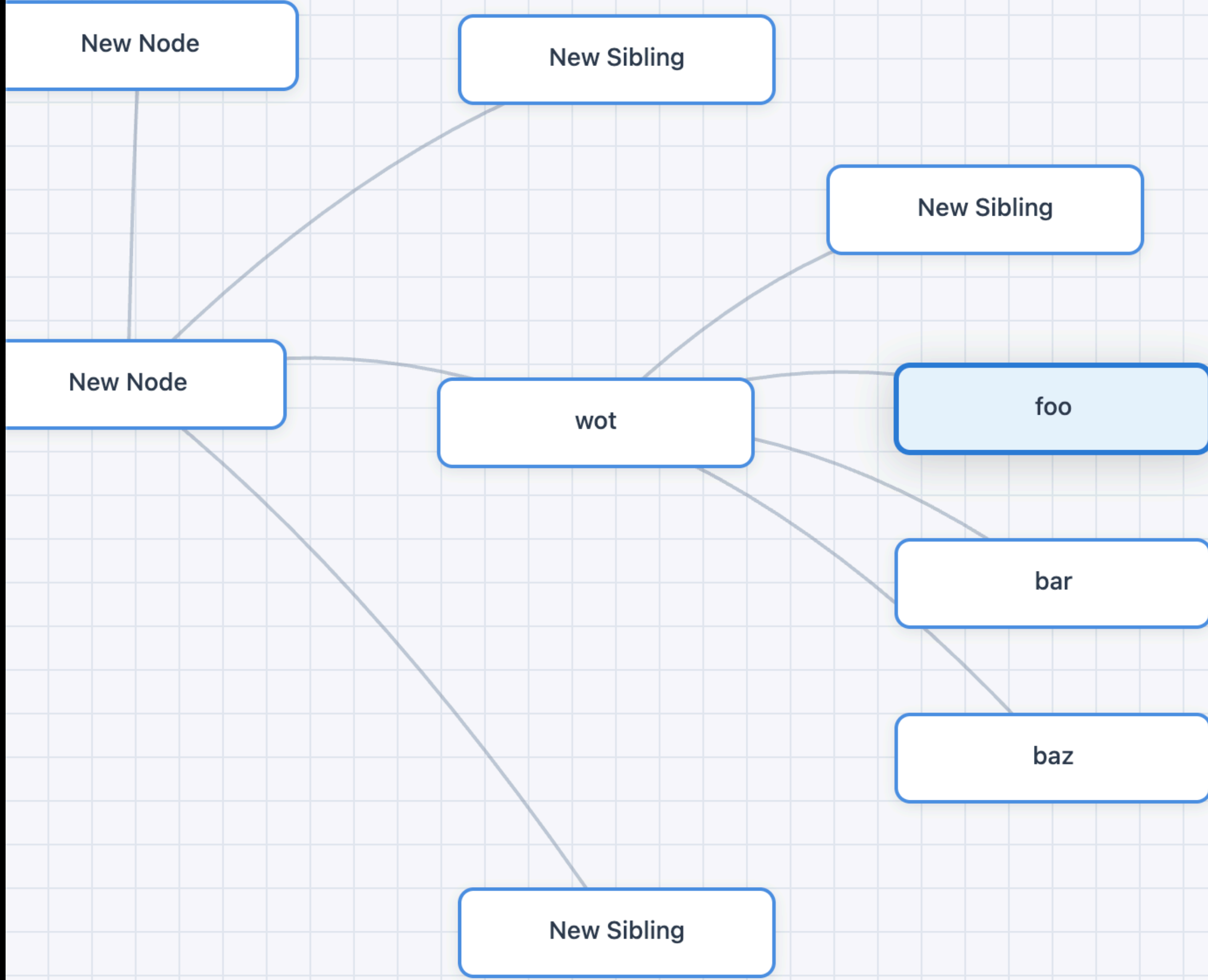
↷ Redo

⚡ Auto Layout

📄 Templates

👥 Collaborate

? Help



# Vertical Slicing as if you meant it

## Learnings & Insights

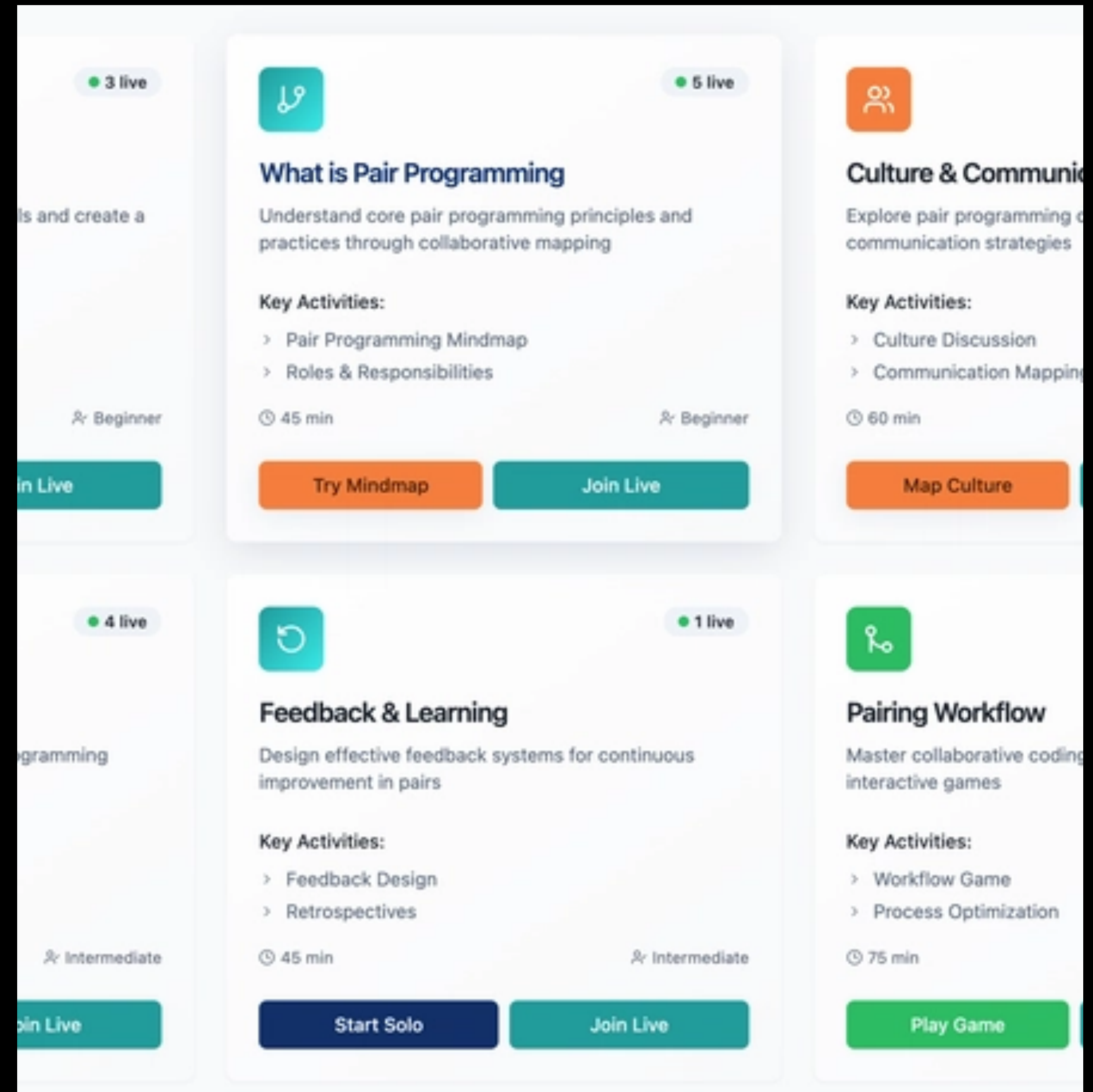
- It is not obvious what is complex and not complex for a LLM.
- Result: The implementation showed that I had missed a crucial part of the architecture: CRDT - Storage optimized for distributed apps
- Hint: Sometimes try 'will it Blend' and let the Agent Fail (or not)

**“Hallucinations” ftw**

# Hallucinations = FTW

## The Prototype!

- Hallucinations can be helpful
- After weeks of searching the right UI pattern: lovable does the job
- I was sticking a bit too close to my coding roots and did not ask UX questions
- I got a result for a programmer, not a User.
- Question: How to use hallucinations to my advantage?





3 live

### Introduction

Set your pair programming learning goals and create a personal roadmap

**Key Activities:**

- > Goal Setting Workshop
- > Learning Plan Creation

30 min

Beginner

Start Journey

Join Live



5 live

### What is Pair Programming

Understand core pair programming principles and practices through collaborative mapping

**Key Activities:**

- > Pair Programming Mindmap
- > Roles & Responsibilities

45 min

Beginner

Try Mindmap

Join Live



2 live

### Culture & Communication

Explore pair programming culture and effective communication strategies

**Key Activities:**

- > Culture Discussion
- > Communication Mapping

60 min

Intermediate

Map Culture

Join Live



4 live

### Driver-Navigator Roles

Master the fundamental roles in pair programming through practice

**Key Activities:**

- > Role Switching
- > Lab Exercises

90 min

Intermediate

Start Solo

Join Live



1 live

### Feedback & Learning

Design effective feedback systems for continuous improvement in pairs

**Key Activities:**

- > Feedback Design
- > Retrospectives

45 min

Intermediate

Start Solo

Join Live



7 live

### Pairing Workflow

Master collaborative coding workflows through interactive games

**Key Activities:**

- > Workflow Game
- > Process Optimization

75 min

Advanced

Play Game

Join Live

# Hallucination = Good

## Insights & Learnings

- After weeks of searching the right UI pattern: lovable does the job
- I was sticking a bit too close to my coding roots and did not ask UX questions
- The new idea: 'Free ad hoc collaboration on the document.'
- couple prompt and a validator to reduce the amount of hallucinations
- Different tools have totally different ways to act on the same set of instructions. One tool's answer = other tool's hallucination.
- Asking the same question twice is interesting too

# Conclusions

- The cost of software has changed. Less on production time. Maybe much less.
- This changes the economics of Prototyping and Project Organisation at large.
- working with IImS has a lot in common with generative art. Good for you, if you did not skip art class.



**questions?**

**contact!**

**sebastian.schuermann**

**<https://github.com/sebs>**

