

# From Vibe to Vision



The New Creator's Toolkit for Documenting and Communicating AI-Built Prototypes.

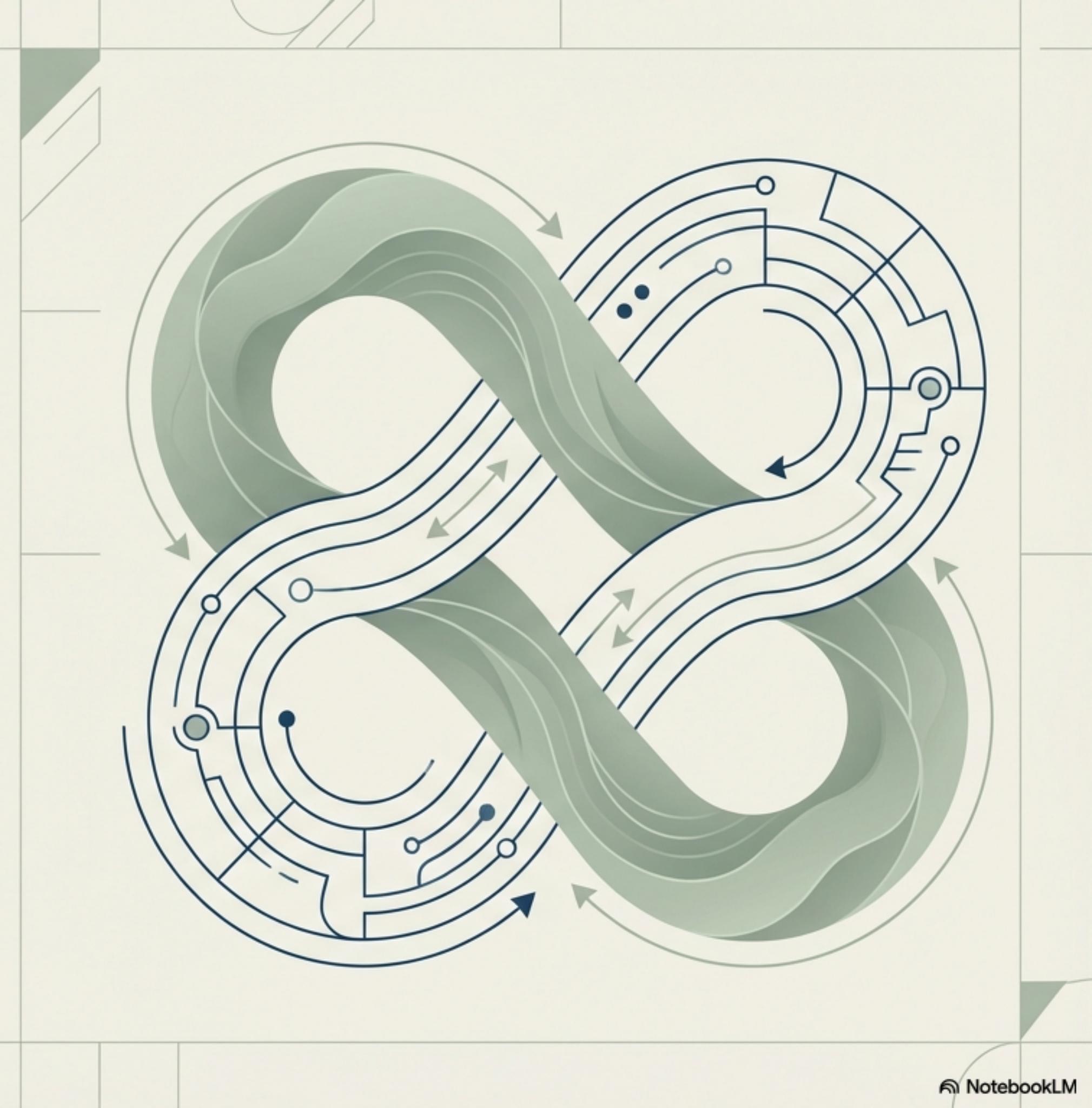
# The Rise of the Vibe Coder

“The hottest new programming language is English.”

— Andrey Karpathy

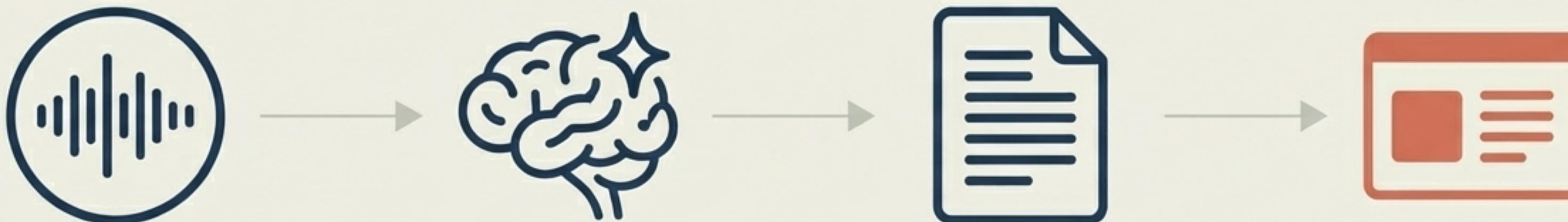
We are in a new era of software creation. Innovators are “vibe coding”—using natural language to guide AI assistants and build prototypes. You’re not writing syntax; you are “steering the thing until it feels right.”

This has democratised development, allowing product thinkers and entrepreneurs to bring their ideas to life in hours, not weeks.



# We Used This Workflow to Build This Guide

The source white paper for this presentation is a product of the AI-augmented workflow we are about to detail. It was co-created by Dinis Cruz and an advanced AI assistant (ChatGPT Deep Research), transformed from an initial “brain dump” voice memo into a structured, formal paper.



# From Prototype to Product: The Great Divide

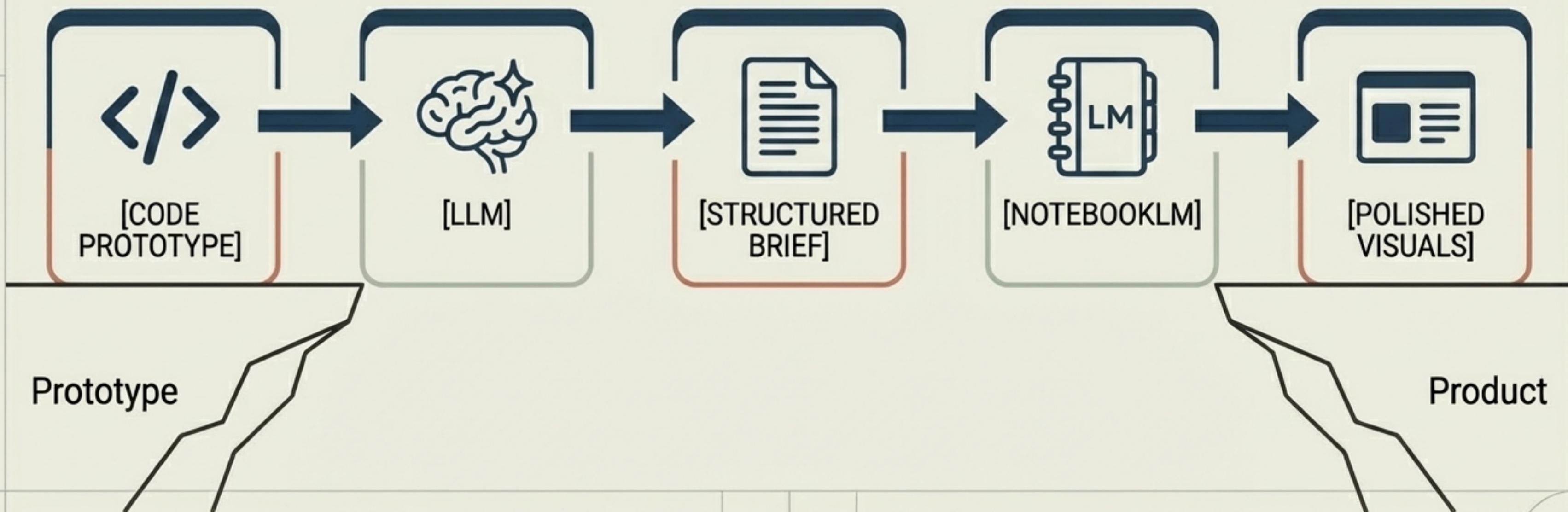


## THE COMMUNICATION GAP

The ease of vibe coding a prototype can mask the complexity of scaling it. As projects mature, creators face a gap. They need to manage non-functional requirements (performance, security, maintainability), handle version control, and clearly communicate the system's design. As one observer noted, for more control, you need to “*clearly and unambiguously specify what you want it to do. The best way to do that is coding.*” This is the communication challenge we solve.

# The Bridge: AI Debriefs AI's Work

We can bridge the gap by leveraging a two-step AI workflow. First, we use a Large Language Model (LLM) to debrief the work done by your AI coding assistant. Then, we use Google's NotebookLM to instantly visualise that debrief.



# Step 1: Generate a Technical Debrief with an LLM

After a coding session, ask your AI assistant to summarise what you've built. This creates a structured document explaining the app's features, architecture, and status. The AI infers the purpose of the code it wrote and articulates it for you.

...  
**Example Prompt:**  
*"Explain what we just built. Provide a technical overview of the application, including its key features, the tech stack used, how the main functionalities are implemented, and any important assumptions or future work."*



## Pro Tip:

You can start with a voice note. Record your thoughts, transcribe the audio (e.g., with Otter.ai), and then use an LLM to clean it up and turn your spoken ideas into a polished, structured document.



## Step 2: Instantly Visualise Your Brief with NotebookLM

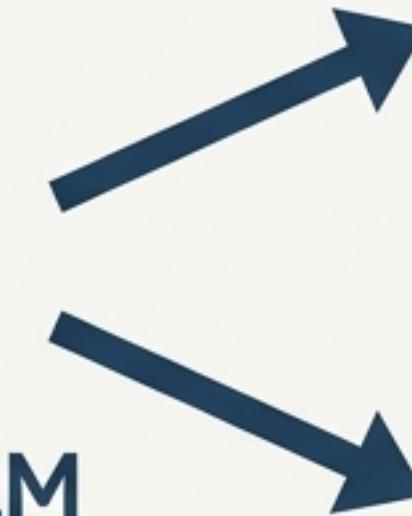
Google's NotebookLM is an AI notebook that analyses your documents and generates outputs from them. Its twin superpowers are:

- 1. Analysis:** It finds and understands information within your documents.
- 2. Visualisation:** It instantly transforms that information into infographics and slide decks.

 **Key Feature:** NotebookLM grounds its output *only in the content you provide*, not general web data. This ensures the visuals accurately reflect your project's specific details.



**NotebookLM**



Infographic



Slide deck

# The Transformation: From Raw Ideas to Polished Assets

```
// unformatted code  
def func():  
    print('data init')  
    # needs error handling  
    x = load_db()  
    return x  
  
# temp notes  
# brain dump:  
# what if we use a new model for this?  
# client mentioned data synthesis but vague  
# deadline in 2 weeks, focus on core api  
# this part is tricky, need to revisit later  
# no documentation yet  
  
// more code  
if x: process(x)
```

BEFORE

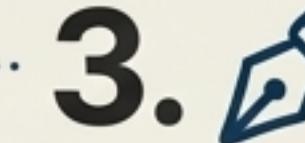
Raw Prototype & Notes



AFTER

Professional Infographic & Pitch Deck

# The Full Workflow: From Code to Slides in Minutes

1. 	2. 	3. 	4. 	5. 	6. 	7. 
<b>Build</b> Use your AI coding assistant to create your prototype.	<b>Summarise</b> Ask an LLM to generate a technical brief.	<b>Refine</b> Edit the brief and save as a PDF or Markdown file.	<b>Upload</b> Add your brief as a source in NotebookLM.	<b>Generate</b> Create an infographic and a slide deck with one click.	<b>Iterate</b> Regenerate as needed to get different visual styles or content focus.	<b>Export &amp; Share</b> Download your assets and share with stakeholders.

# Best Practices for Exceptional Results



## Write Rich, Narrative Briefs

The quality of the output depends on the richness of your source document. Don't just list facts; tell the story of your project. NotebookLM thrives on well-structured, story-like input.



## Use Clear Sections

Structure your brief with headings like "Introduction," "Problem," "Features," and "Future Work." Your slide deck will likely mirror this logical structure.



## Regenerate and Remix

Don't accept the first output. Generate 2-3 versions of a deck. You can then export them and cherry-pick the best slides from each, as direct editing is not yet available in the tool.

# Your New Creative Superpowers

This workflow doesn't just produce documents; it empowers you.



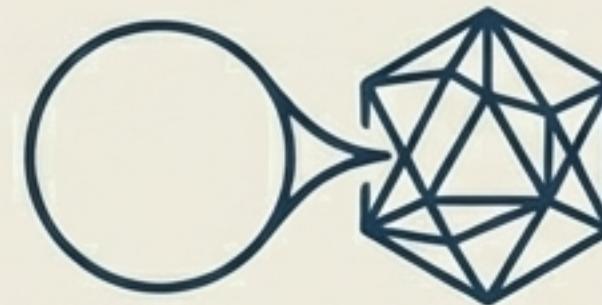
## Achieve Total Clarity

Asking an AI to explain your project helps you understand your own creation on a deeper level.



## Bridge to Collaboration

Provide developers with clear briefs and visual aids, creating the technical onboarding materials for your own project.



## Communicate Like a Pro

Go from a working app to a polished pitch deck in minutes. Speak the language of developers and investors.



## Spark New Ideas

AI-generated visuals can synthesise and reframe your input in insightful ways, revealing new connections and possibilities.

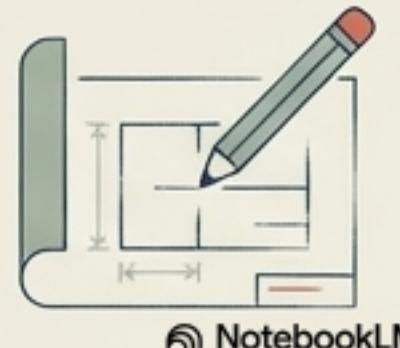
# A Realistic Look: Current Limitations (as of Dec 2025)

NotebookLM is a powerful new tool, and it's evolving rapidly. Be aware of the current state:

- **Usage Limits:** Free tiers may have caps on generating infographics or slides.
- [REDACTED]
- **No In-Tool Editing:** Generated slides cannot be edited directly within NotebookLM yet. You must export to Google Slides or PowerPoint.
- **'Black Box' Generation:** The tool does not show the intermediate prompts it used to create the visuals.

**“Today’s NotebookLM is the weakest it will ever be.”**

Future versions will be more powerful and flexible.

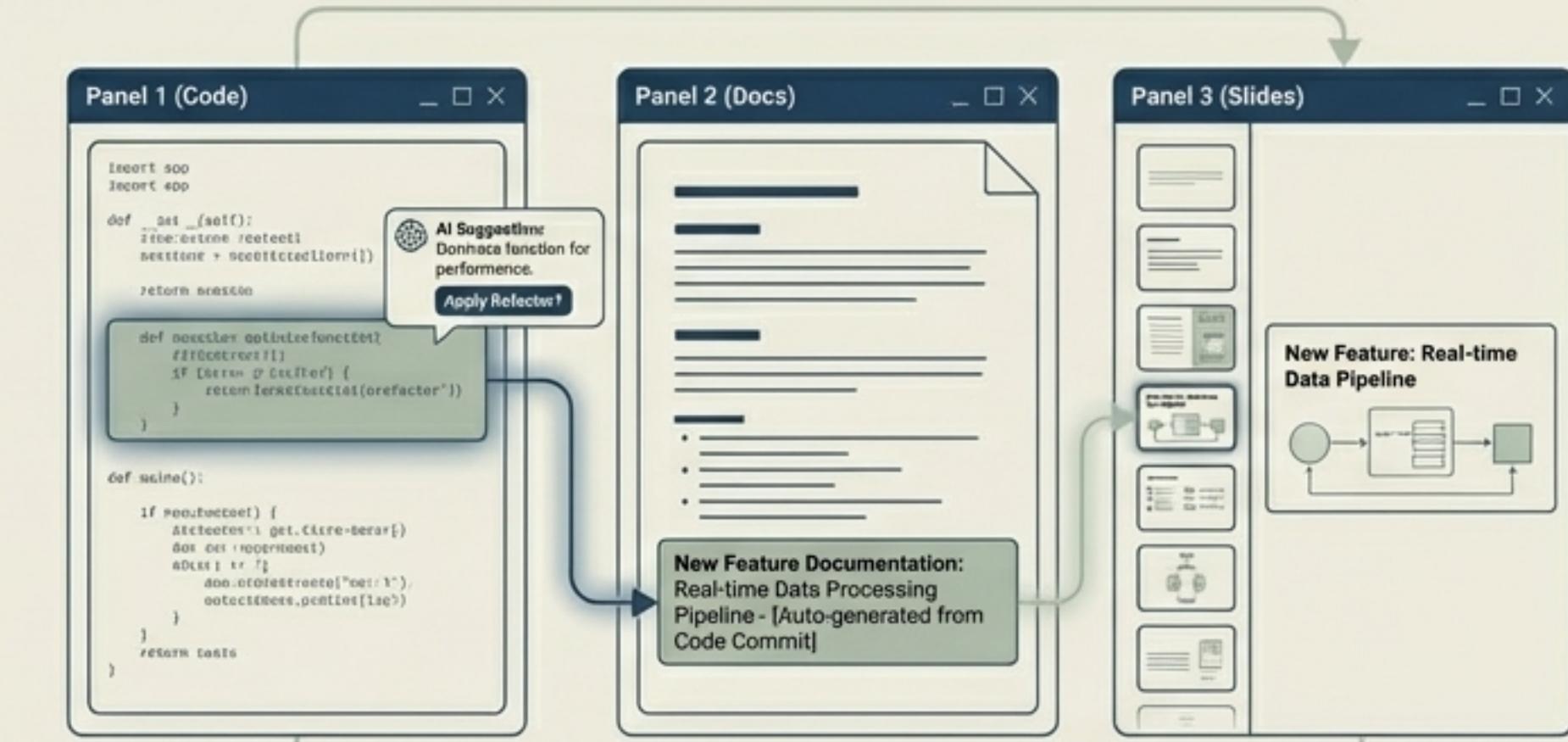


# The Future is an Integrated Creative Suite

This workflow is just the beginning. Soon, we can expect end-to-end integration where documentation and visuals are generated in real-time as you code.

Imagine an AI observing your progress and bring an updated slide deck ready after every major feature commit.

The role of creators will be to provide the vision, while AI handles the boilerplate drafts and documentation.



# You Now Have Three AI Co-Pilots

You are never working alone. By embracing this workflow, you assemble a powerful team to bring your vision to life.



## Your Coding Co-Pilot

To build the application.



## Your Writing Co-Pilot

To create the technical brief.



## Your Design Co-Pilot

To generate polished visuals.