

Building and Prototyping Apps with AI

Learn How to Go From Idea to Interactive Prototype in 90 Minutes

Adrian Cole, Founder, 30 Days of Product

30+ Products, 10 Years, Hardware —> Software —> AI Agents

Welcome and Framing

- The fastest way to see your app idea in action
- Why AI changes how we prototype
- Why thinking still matters more than tools

📌 **Key Point:** Today is not about replacing engineers. It is about compressing the time between idea and interface. We are going to learn how to use AI as leverage. The real skill you are building today is structured thinking.

Workshop Objectives

By the end of this session, you will:

Understand what AI powered prototyping is and is not

Understand the technology shift enabling this moment

Know when to use AI prototypes vs traditional development

Use a structured App Blueprint before building

Build or outline a working prototype

Understand risks and limitations

- ❏ **Key Point:** Clarity, not just speed, is the goal. If you leave here with one structured idea and one working prototype, this workshop worked.





The Old Way vs The New Way

1

Old Model

Idea → Spec → Developer → Weeks or Months → First Interface

2

New Model

Idea → Structured Thinking → AI Prototype → Same Day Feedback

📌 **Key Point:** Five years ago, you needed engineering just to see your idea. Now you can simulate workflows almost instantly. The bottleneck is no longer code. It is clarity.

What Is AI App Prototyping?

AI App Prototyping means using AI tools to generate:

- User interfaces
- Basic logic
- Workflows
- Mock data

It is not:

- FULL Production architecture (YET)
- Secure infrastructure
- Scalable backend systems

📌 **Key Point:** Think of AI prototyping as a visualization and experimentation engine. It allows you to simulate how something works before committing engineering resources.

Prototype vs Production

Prototype:

- Tests workflows
- Validates assumptions
- Demonstrates concept
- May break

Production:

- Secure
- Scalable
- Reliable
- Built with clean architecture

📌 **Key Point:** Most platforms claim you can ship to production. Technically you can. Practically, real products require engineers. The prototype is a learning tool.

The Technology Shift

What changed?

Large Language Models can interpret intent

AI can generate UI from descriptions

Backend services are abstracted

Hosting is automated

- ❏ **Key Point:** The breakthrough is intent translation. You describe what you want. The model converts that into interface and logic. That is a massive reduction in friction.

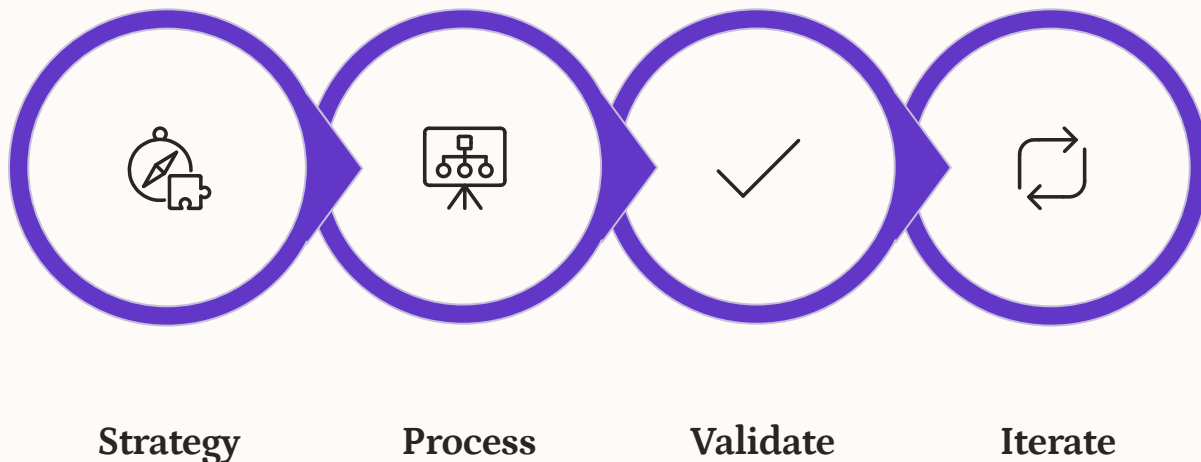
The New Builder Skill

Old Skill: Writing Code

New Skill: Structured Thinking, Prompt Engineering

You must be able to:

- Define the user
- Define the problem
- Define the outcome
- Define the workflow
- Use the AI Tools



❏ **Key Point:** AI amplifies clarity. If your thinking is messy, your prototype will be messy.

When to Use AI Prototypes

Best For:



Validating workflows



Testing user journeys



Investor demos



Early customer feedback



Internal tools



Personal dashboards

 **Key Point:** This is about learning fast. Before spending money on development, you can test assumptions.

Example Use Cases: Three Powerful Categories

AI prototyping is especially powerful in three categories:

01

Personal Products For Your Own Life

02

Internal Tools For Business Workflows

03

Product Exploration And New Venture Prototyping

- ❏ **Key Point:** Different contexts require different levels of risk tolerance. AI prototypes shine where speed and learning matter more than perfection.

Category 1: Personal Products

Example: Personal Fitness Dashboard

- Track workouts
- Track nutrition
- Monitor bloodwork
- Visualize progress over time

Why this works well:

- You are the user
- Low security risk
- Fast iteration
- Immediate feedback loop

📌 **Key Point:** Personal tools are ideal because failure is low risk and learning is high value.

Category 2: Internal Business Workflows

Example: Revenue Or Marketing Operations Dashboard

- Connect to sales or CRM data
- Track pipeline stages
- Monitor campaign performance
- Visualize revenue metrics in one place

This is powerful for:

- Founders
- Marketing teams
- Operations teams

Internal users do not need a polished consumer grade product. They need clarity and speed.

📌 **Key Point:** Internal tools are perfect for AI prototypes because they optimize workflows and can connect to data sources without needing public scale reliability.

Category 3: Product Exploration And New Ideas

Example: Travel App For Business Professionals

- Curated short trip recommendations
- Meeting friendly hotel suggestions
- Calendar based travel planning
- Expense visibility and trip summaries

In this stage, you are exploring:

- Is the workflow intuitive?
- Does this solve a real problem?
- What features actually matter?

📌 **Key Point:** AI prototypes are exceptional for product exploration because they let you test assumptions before investing in full scale development.

Risks and Limitations

Common Issues:

Fragile logic

Weak database
structure

Security gaps

Limited scalability

Technical debt

❏ **Key Point:** You cannot prompt your way into good architecture. AI can simulate behavior but not guarantee system integrity.

Responsible Usage Mindset

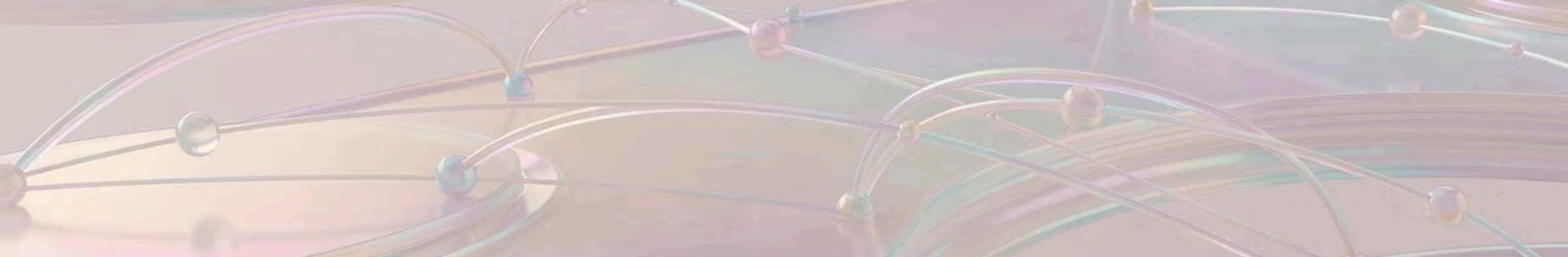
Use AI Prototypes To:

- Learn
- Validate
- Demonstrate

Do Not Use Them To:

- Skip engineering discipline
- Ignore security
- Avoid user research

📌 **Key Point:** The right mindset protects you from false confidence.



Landscape Overview

AI App Building Tools Overview

We will look at:

Claude

Lovable

Base44

VO.dev

HeyBoss

📌 **Key Point:** Each tool has strengths. No tool is perfect. Your job is to match tool to use case.

Tool Comparison Table

Tool	Strengths	Weaknesses	Best For
Claude	Excellent reasoning and prompt refinement, strong for logic design and workflow thinking, flexible for drafting specs and structured prompts	Not a visual builder by default, requires pairing with another tool for UI generation	Thinking through logic, refining prompts, designing workflows before building
Lovable	Fast UI generation, clean modern interface output, beginner friendly	Limited backend depth, less control over architecture, may abstract too much	Rapid visual prototypes and early concept demos
Base44	More structured app building, clearer feature modeling, stronger control over app structure	Slightly higher learning curve, less purely visual than some competitors	Builders who want more structure and control early
VO.dev	Strong frontend generation, excellent for component level UI work, good for React based projects	Frontend focused, requires backend pairing, more technical	Developers or semi technical founders building custom UI layers
HeyBoss	End to end generation including UI and logic, beginner accessible, fast iteration	Risk of hidden technical debt, limited transparency into architecture, scaling concerns	Non technical founders testing end to end app concepts quickly

📌 **Key Point:** Do not just ask, can it build my app? Ask, how much control do I need and what stage of the journey am I in?

Structured Process Before Prompting

1

Do Not Start With:

"Build me an app that..."

2

Start With:

- Clear problem
- Clear user
- Clear workflow

📌 **Key Point:** Most people skip this step and get mediocre results.

How To Actually Build With AI

When using AI app builders, do not ask for everything at once.

Current AI builders tend to:

- Cut corners
- Make hidden assumptions
- Skip edge cases
- Over simplify logic

Instead, build the way you would write real code.

One feature at a time. One workflow at a time.

Walk through the build the way a user would experience the product.

Ask yourself:

01

What is the first action the user takes?

02

What happens next?

03

What is the second task?

04

What is the third task?

Build in steps and layers.

Start with the primary workflow. Then add supporting features. Then refine logic.

❏ **Key Point:** AI performs dramatically better when you guide it step by step instead of dumping the entire vision into a single massive prompt.

Bad Example: The Loaded Prompt

Here is what not to do.

"Build me a full scale AI powered social fitness marketplace app with real time chat, video streaming, payments, subscription tiers, AI workout generation, calendar syncing, wearable integration, a scalable backend, enterprise level security, admin dashboard, analytics system, automated email marketing, push notifications, a referral program, and make it production ready and beautiful with modern UI. Also optimize it for global scale and make sure it can handle millions of users."

What is wrong here?

- It asks for everything at once
- It mixes prototype and production requirements
- It assumes architecture decisions
- It overloads the model with scope
- It gives no clear primary workflow

📌 **Key Point:** When you overload the prompt, the AI will guess, cut corners, and fabricate structure. This is how messy prototypes are born.

Check The Prompt

Let's break this down together.

What specifically makes this a bad prompt?

Discussion:

- Where is the clear primary workflow?
- What assumptions is the AI forced to make?
- Which parts are prototype scope and which parts are production scope?
- What would likely break first?



Key Point: If you cannot explain the first three user actions clearly, you are not ready to prompt.

Good Example: Step One Only

Instead of asking for everything, start small.

Example Step One Prompt:

"Build a simple web app prototype for busy professionals who want to find a gym partner.

For this step, only build the onboarding flow.

The user should:

1. Create a basic profile with name, fitness level, and availability.
2. Select their primary fitness goal.
3. Confirm their profile and land on a simple dashboard screen.

Do not include chat, payments, subscriptions, or advanced features yet. Focus only on the onboarding workflow and basic UI screens."

Why this works:

- Clear user
- Clear scope
- Single workflow
- Explicit constraints

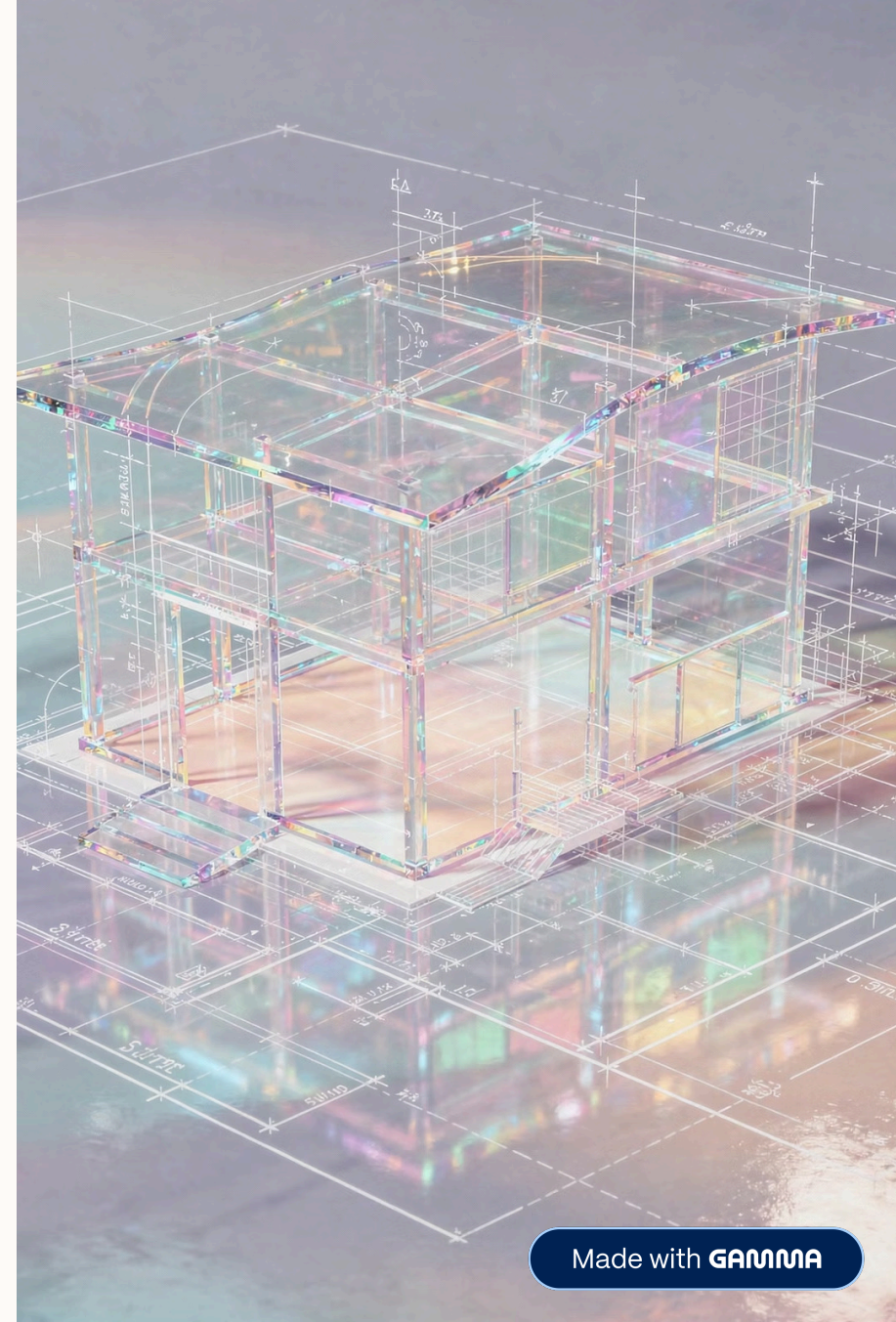
📌 **Key Point:** You are guiding the AI like you would guide a junior developer. Clear scope. Clear sequence. Clear boundaries.

Introducing the App Blueprint

The App Blueprint Includes:

- App Idea Name
- Problem Statement
- Solution Statement
- Ideal Customer
- Feature Brainstorm
- Final Reflection
- One Line Pitch

📄 **Key Point:** This worksheet is your thinking tool. The prototype comes after this.



Blueprint Deep Dive

Key Rules:

Make the problem concrete

Limit features to five

Define one primary workflow

Be specific about your user

📌 **Key Point:** If the problem is vague, the app will be vague.

Build Along Setup

Step 1: Choose a Simple App Idea

Examples:



Habit tracker



**Networking
matcher**



**Fitness
dashboard**

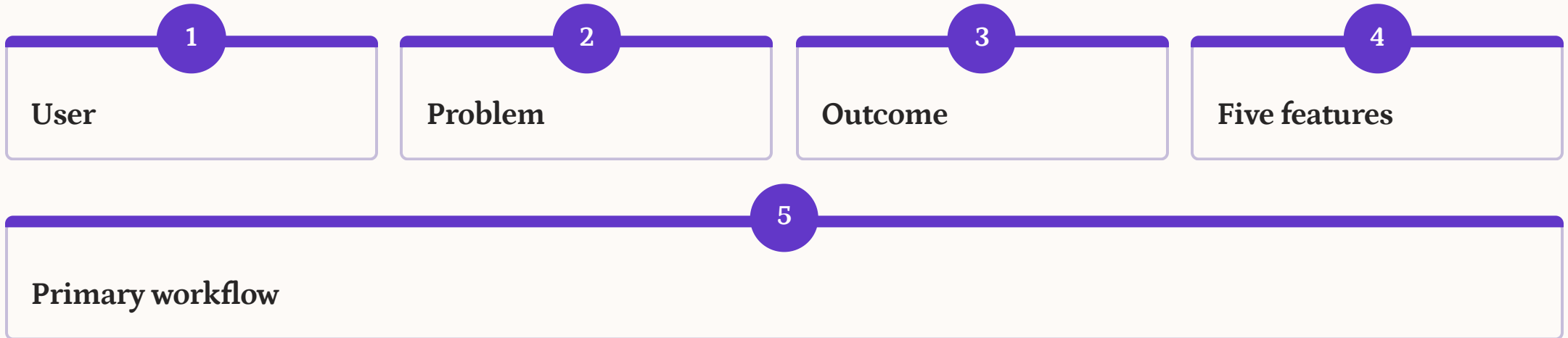


**Task
prioritization
tool**

📌 **Key Point:** We will build something simple but structured.

Fill Out Blueprint Together

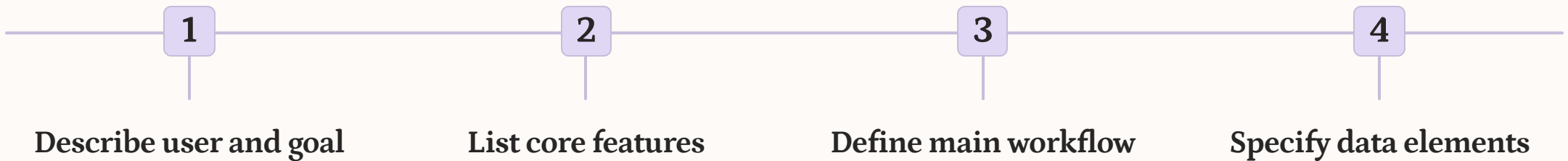
We will define:



📌 **Key Point:** Think clarity over complexity.

Translating Blueprint to Prompt

From Structure to Instructions:



📌 **Key Point:** AI performs better when instructions are structured.

Generate Prototype

Live Demo:

- 1 Generate UI
- 2 Walk through workflow
- 3 Identify gaps

📌 **Key Point:** We are evaluating the idea, not just admiring the interface.



Debrief

Ask:

| Does this solve the problem?

| What is confusing?

| What would break at scale?

| What needs engineering?

📌 **Key Point:** This is where learning happens.



Key Takeaways



**AI accelerates
visualization**



Structure beats speed



**Prototypes are learning
tools**



Engineers still matter

❏ **Key Point:** Your advantage is clarity plus speed.

Home Work

This Week:



Pick one idea



Use the App Blueprint



Build one prototype



Show it to a real person



Key Point: Momentum matters more than perfection.