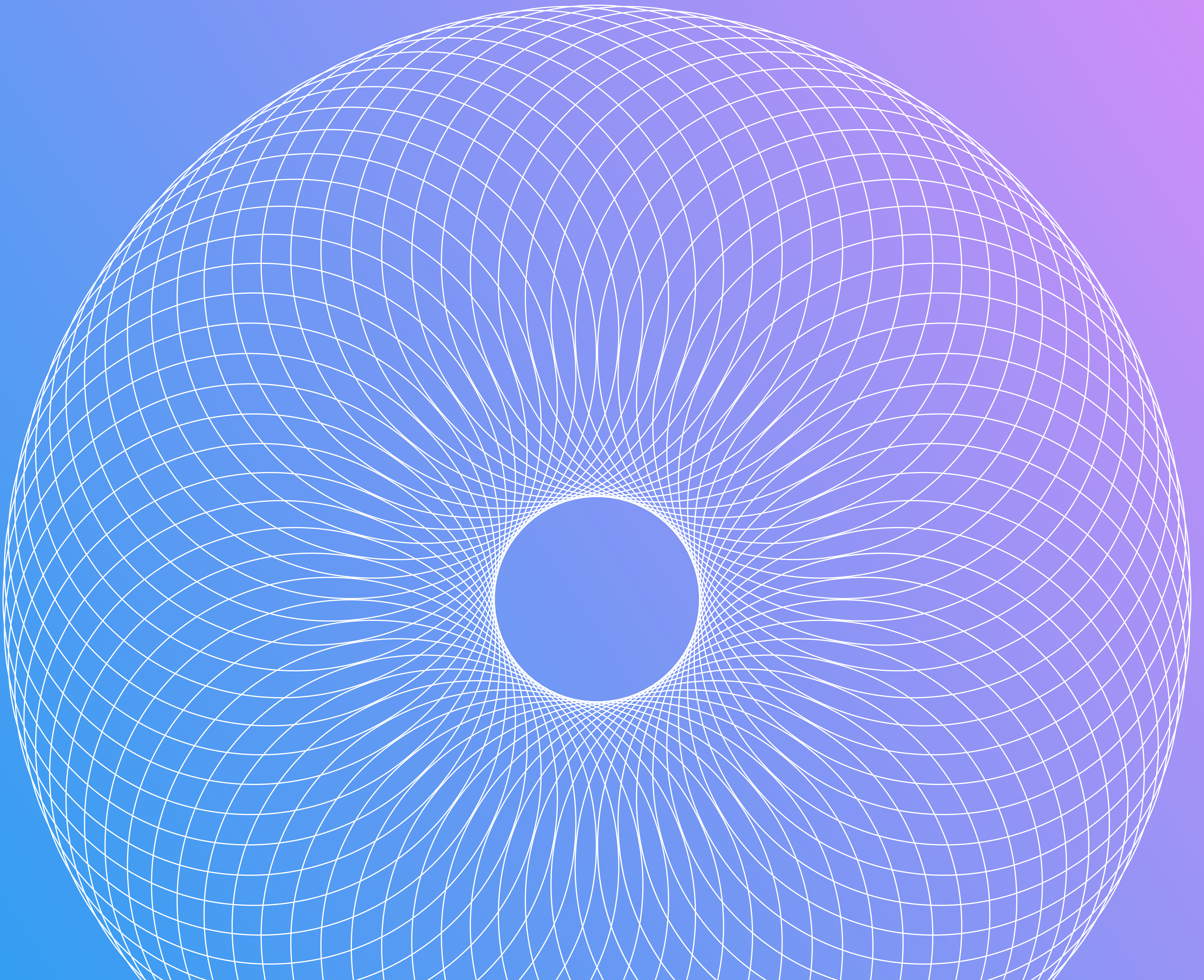




A guide to **AI-First** Product Design

Practical strategies for designers and product teams to transform AI experimentation into systematic competitive advantage through proven practices.



Contents

Exec Summary

AI-First Principles

The AI-First Operating System

Intelligent Systems

Designing With Intent

New Ways of Working

Redefining Teams and Roles

The ROI of AI-First Product Teams

Final Thoughts



Exec Summary

Product design is at a clear turning point. AI is no longer just an interesting experiment — it’s becoming a practical driver of how companies design, build, and compete. Work that once took weeks of iteration, documentation, and handoffs can now happen in hours, sometimes minutes. And it’s not just about moving faster — it’s about changing how product teams think, collaborate, and deliver value.

This shift isn’t only about technology. It’s about how organizations operate. Teams that succeed with AI don’t simply add new tools to old processes — they rethink the operating model from the ground up. That’s why we’re seeing a new breed of AI-native companies emerge — built from day one with these practices in place — moving faster, learning quicker, and outpacing competitors.

The teams leading this shift are:

- **Building AI-ready design systems** that encode intent, not just visual appearance, so AI can generate context-aware outputs.
- **Running on a new AI-driven operating system** where context, memory, and tools work together to drive learning velocity.
- **Redefining roles** — from AI Operations to Design Engineering — to orchestrate human-AI collaboration at scale.
- **Embracing new product lifecycle methods** to harness AI’s speed for deeper craft and bigger problem spaces.

Early adopters are already seeing **30–50% efficiency gains for designers and engineers**, and **25% + for product managers** — but the real win is qualitative: more time spent on strategic, high-value work and a compounding edge over time.

This guide is your roadmap: the principles, infrastructure shifts, and maturity milestones needed to turn scattered AI experiments into a sustainable advantage.

The real question for product leaders isn’t if AI will reshape your practice — it’s whether your team is ready to compete with the new generation of AI-augmented and AI-native players.

Jon Sukarangsang
Principal & Founder
Summer Friday & Partners

Introduction: Understanding the Transformation

Traditional product design operates within well-established constraints: manual iteration cycles, documentation overhead, and the inherent friction between design intent and technical implementation. These constraints have shaped how teams organize, how they measure progress, and what they consider achievable within sprint cycles.

AI introduces new possibilities that challenge these assumptions. When design systems can generate contextually appropriate components, when documentation updates automatically, and when design-to-code translation becomes seamless, teams can redirect effort toward higher-value problem-solving.

The productivity improvements are significant—teams report 30-50% efficiency gains for designers and engineers, with product managers seeing 25% improvements in their core workflows. But the more important shift is qualitative: teams spend less time managing design debt and more time addressing user needs and business objectives.

This reallocation of effort creates a compounding effect. Better-researched features, more thoughtful interactions, and deeper attention to user experience details become achievable within existing resource constraints.

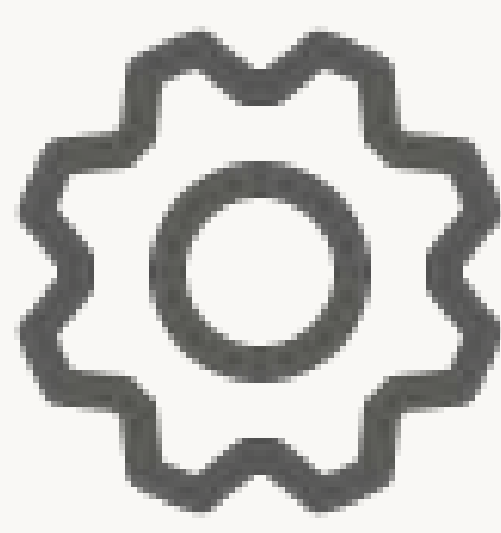


Product x AI Opportunity Wins



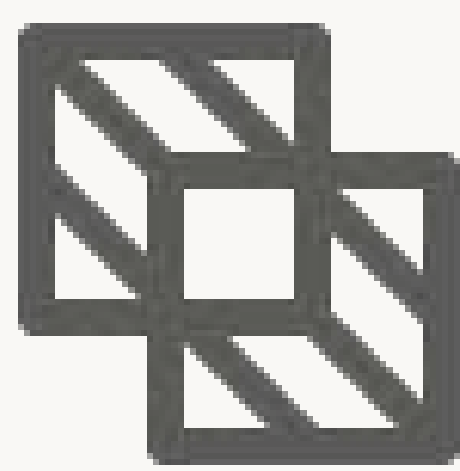
Intelligent Documentation

When designers modify components in Figma, AI automatically updates documentation in Confluence, generates changelogs, and syncs metadata across connected systems.



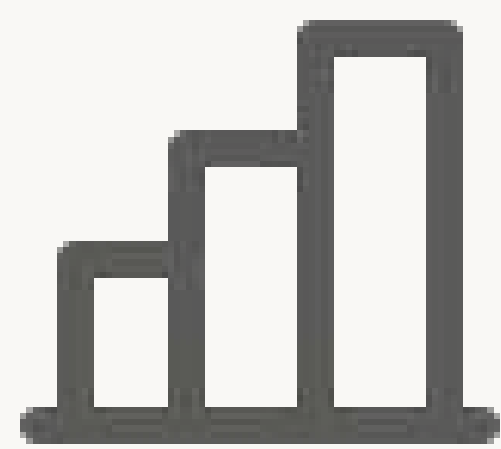
Knowledge Discovery

AI acts as an intelligent librarian, instantly answering questions like "What are the accessibility guidelines for form inputs?" by retrieving information from internal wikis and project archives.



Design-Development Pipeline Automation

AI generates production-ready code components upon design finalization, complete with styling and functionality, while creating pre-populated development tickets.



Continuous Quality Assurance

AI continuously scans design files for brand guideline deviations, identifies inconsistent component usage, and flags accessibility issues before they become costly to fix.



Contextual Design Generation

AI produces brand-compliant outputs by leveraging design system rules and user data, automatically generating landing page variations optimized for specific segments while maintaining brand compliance.

The AI-First Operating System

AI's impact isn't just in isolated wins—it comes from a foundation where context, memory, and tools work together. That foundation is the AI-First Operating System.

The AI-First Operating System is our framework that aligns your technology, design assets, and ways of working so AI can deliver real, repeatable business value. It's the playbook for transforming scattered AI experiments into an integrated, strategic capability. Built on three interdependent pillars, it connects the technical foundation, the design leverage, and the operational practices that allow teams to adapt and improve continuously.

1

Intelligent Systems

The foundational systems layer where context, memory, and tools are built, connected, and configured to support AI. This foundation ensures AI outputs are relevant, accurate, and actionable, enabling better decisions and more efficient execution across the product lifecycle.

2

Designing with Intent

AI-ready design systems that encode meaning, not just appearance. These systems give AI the rules, rationale, and constraints it needs to generate brand-consistent, context-aware work at scale, turning your design system into a true force multiplier.

3

New Ways of Working

Workflow and process redesign for the AI era. Non-linear, adaptive, and experimentation-driven approaches that turn AI's speed into deeper craft, faster learning, and better outcomes. This is where teams rethink how they move from problem definition to delivery.

Principles for AI-First Product Teams

The transition to AI-first product design requires more than adopting new tools—it demands a fundamental shift in how teams think about their work, their processes, and their relationship with technology. These five principles form the foundation of this transformation, guiding organizations from traditional design practices toward a more dynamic, intelligent, and human-centered approach. Rather than simply layering AI onto existing workflows, these principles reshape how product teams operate, collaborate, and deliver value.

Iteration over Artifacts

Traditional product design emphasizes the creation of polished deliverables—detailed specifications, pixel-perfect wireframes, and comprehensive documentation that teams spend weeks perfecting before any testing begins. AI-first design flips this approach, prioritizing rapid iteration and treating deliverables as temporary stepping stones rather than precious final products. Teams can now generate multiple design variations in minutes, test concepts quickly, and learn what resonates with users before investing significant time in detailed implementation. This shift allows designers to explore more possibilities, fail faster, and ultimately arrive at better solutions through continuous refinement rather than upfront perfection.

Elevate Human Thinking

In traditional workflows, designers and product managers often find themselves consumed by repetitive tasks—resizing assets, writing documentation, creating handoff specifications, and managing version control across multiple tools. AI-first design liberates human talent from these routine activities, enabling teams to focus on higher-order thinking: strategic problem-solving, creative exploration, empathetic user understanding, and complex interaction design. When AI automatically generates design specifications, updates documentation as components change, and handles the mechanical aspects of design systems maintenance, teams can dedicate their uniquely human capabilities to the work that truly matters—understanding user needs, solving complex problems, and innovating meaningful solutions.

Experiment and Learn Quickly

Traditional organizations often approach new methodologies with comprehensive pilot programs, extensive planning phases, and requirements for perfect use cases before implementation. AI-first design embraces a culture of safe experimentation, encouraging teams to start small, test frequently, and learn from both successes and failures. A product team might begin by using AI to generate initial user story drafts for a single sprint, learn from that experience, then gradually expand to automated testing scenarios and design reviews. This approach recognizes that AI capabilities evolve rapidly, and the organizations that adapt fastest are those that make experimentation a core competency rather than a special initiative.

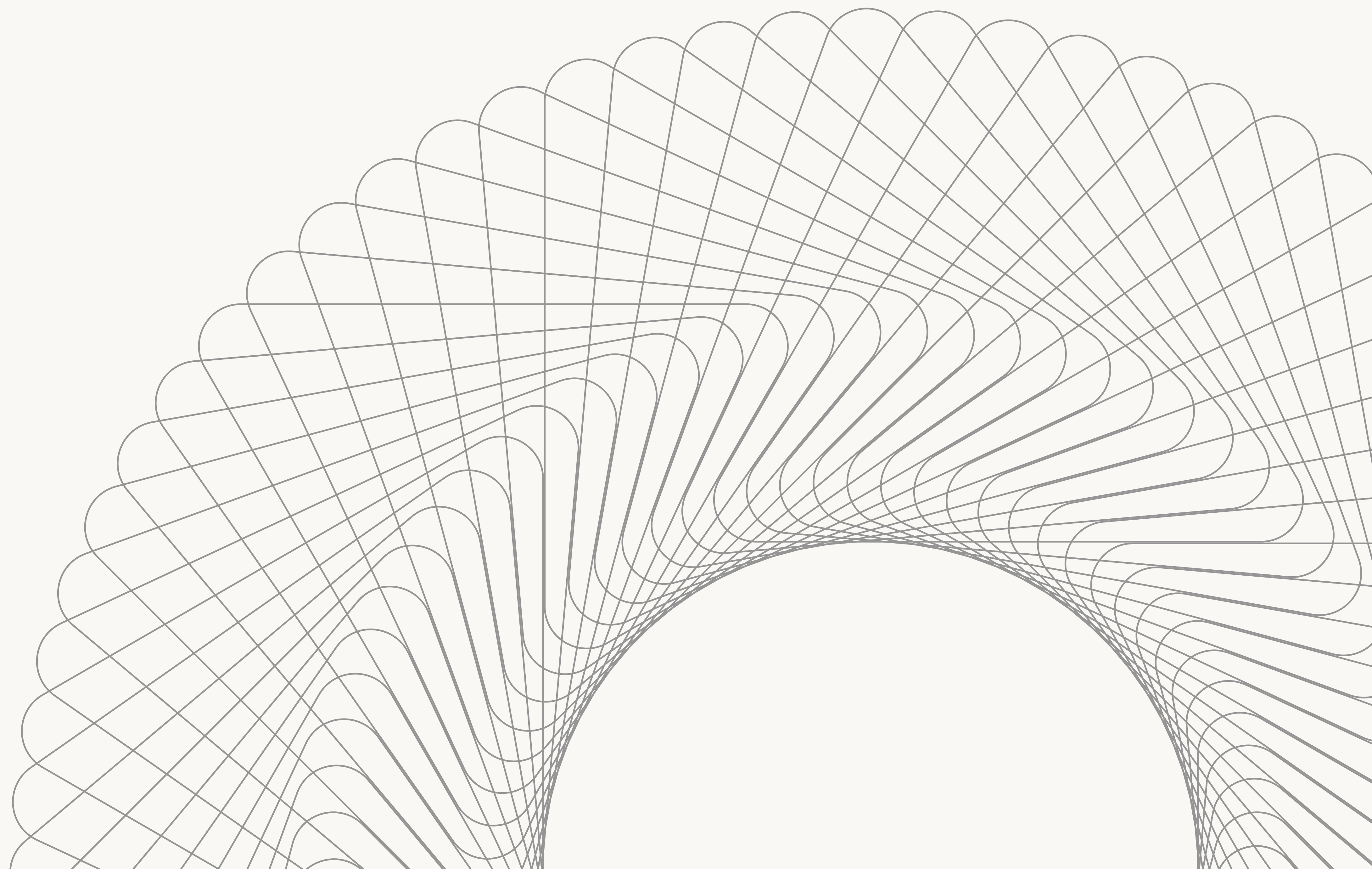
Principles for AI-First Product Teams (continued)

Rethink the Workflow

Many organizations fall into the trap of adding AI features to existing processes without questioning whether those processes remain optimal. AI-first design goes deeper, fundamentally redesigning how teams collaborate, make decisions, and move from concept to launch. Instead of using AI to create better design handoffs, teams might create dynamic design systems where changes automatically propagate through design files, documentation, and code repositories simultaneously. This principle requires teams to step back from familiar workflows and ask not "How can AI improve what we do?" but "What becomes possible when AI handles the connective tissue between our tools and processes?" progress, and what they consider achievable within sprint cycles.

Lead with Trust and Accountability

The power of AI comes with the responsibility to use it transparently and ethically. Traditional approaches often implement AI tools without clear governance, hoping teams will organically develop best practices. AI-first design establishes transparent guidelines, human oversight mechanisms, and clear accountability structures from the outset. Teams create review checkpoints for AI-generated content, maintain audit trails for AI-assisted decisions, and ensure human designers take ultimate responsibility for design choices. This principle recognizes that sustainable AI adoption requires not just technical implementation but cultural practices that maintain quality, ethics, and human agency in the design process.



Traditional vs. AI-First: A Practical Comparison

These principles matter most in the day-to-day. They change how fast ideas move from concept to code, how quickly insights shape decisions, and how many options teams can test before choosing a direction. Side by side, the difference between traditional and AI-first work is unmistakable.

Type of Task	Traditional	AI-First
Design System Updates	Designer manually updates 47 components across Figma files, then documents changes in Confluence, creates JIRA tickets for engineers	AI detects component changes and automatically updates documentation, generates development tickets with specs, and notifies relevant team members
User Research Synthesis	Product manager spends 2 days reading interview transcripts and manually identifying patterns across 20 user sessions	AI analyzes transcripts to surface key themes and pain points, while human focuses on strategic implications and solution ideation
A/B Test Design	Team creates one variation, debates alternatives in meetings, then manually designs 2-3 options over several days	AI generates 10 test variations based on conversion best practices, team selects most promising concepts for validation
Design Handoffs	Designer creates detailed specification document, schedules handoff meeting, answers follow-up questions via Slack	Dynamic design system automatically generates code snippets and specifications when design is marked "ready for development"
Content Creation	Marketing team spends weeks writing copy variations for different user segments and contexts	AI generates personalized content variations while team focuses on brand strategy and messaging frameworks

Intelligent Systems

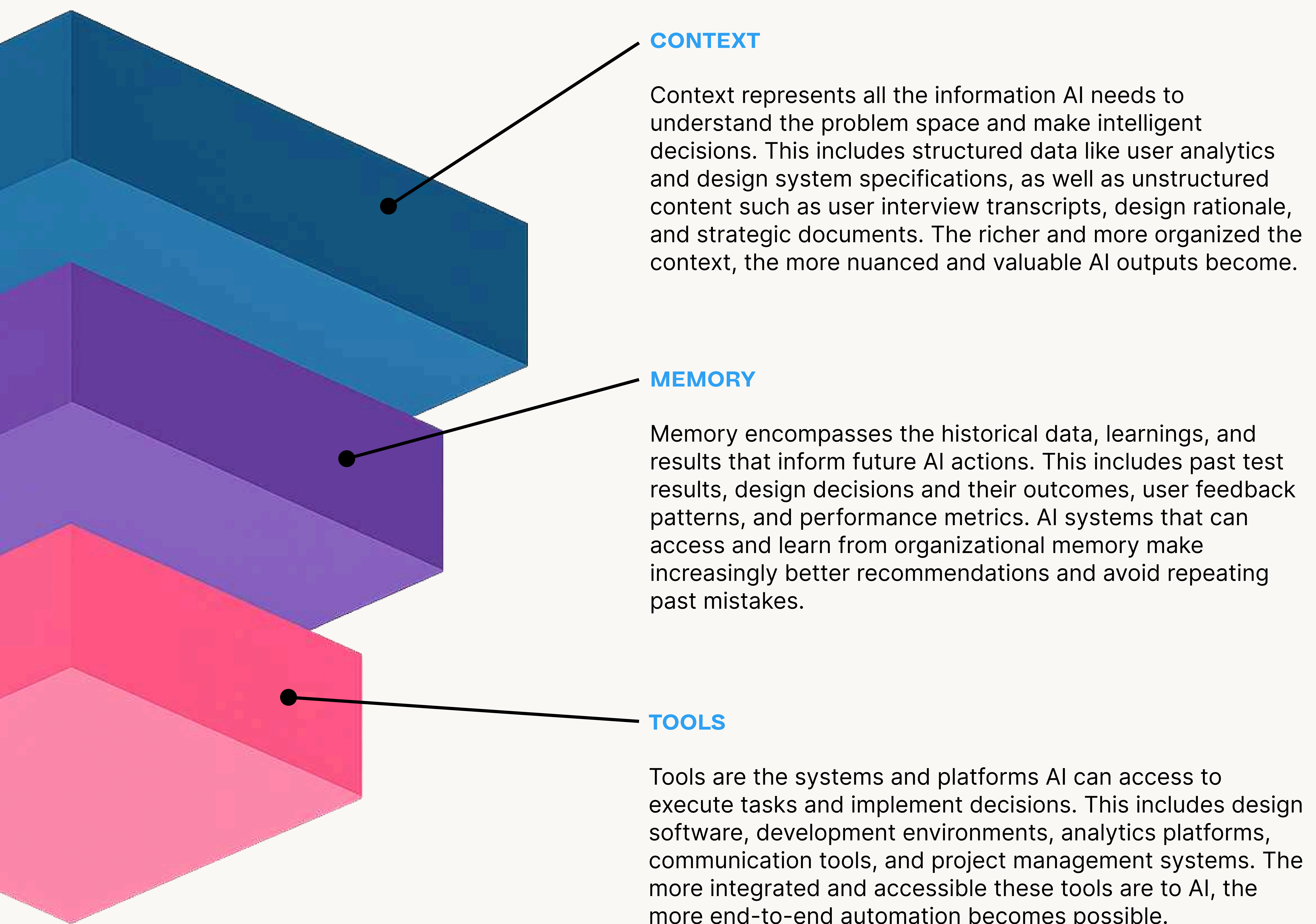
Most product teams are already using AI in some form, and we're going to assume your team is already utilizing in their daily work —whether it's ChatGPT for quick answers, a CustomGPT for project-specific tasks, or an embedded assistant in their tools. But AI-First teams go further: they manage systems built to make AI work faster, smarter, and more reliably across the product lifecycle.

These systems link context (the live data and constraints), memory (what's been tried, what worked, and why), and tools (where the work happens) in ways that AI can navigate without handholding. That requires more than tool familiarity—it demands a working knowledge of AI concepts so teams can shape how the system learns, adapts, and executes.

Done well, Intelligent Systems turn everyday AI use into a competitive advantage—moving from ad-hoc wins to a repeatable, continuously improving engine for product delivery.

The AI Solution Framework

Every successful AI application in product design relies on (at least) three interconnected elements: rich context that enables intelligent decision-making, accumulated memory that improves performance over time, and integrated tools that execute tasks seamlessly. By mapping these components across your workflows, teams can identify the highest-impact opportunities for AI integration and build the foundation for sustainable competitive advantage.



Context, Memory, and Tools: In practice

Great outputs don’t come from tools alone—they come from how well teams connect context, memory, and tools in their workflows. The examples below show how these three elements come together in everyday product work, from A/B testing to content creation, and how richer connections between them unlock better, faster outcomes.

	<i>The data you give it</i>	<i>A way to structure data over time</i>	<i>What it can use to shape its output - via API, webhooks, etc.</i>
	↓	↓	↓
Use Case	Context	Memory	Tools
A/B Testing	User research data, behavioral analytics, conversion metrics, brand guidelines	Historical test results, winning patterns, seasonal trends, audience segment performance	UserTesting, Optimizely, Figma, Analytics platforms, Email marketing tools
Design System Maintenance	Component specifications, usage guidelines, brand standards, accessibility requirements	Past component performance, adoption patterns, maintenance issues, user feedback	Figma, GitHub, Documentation platforms, Design token tools, CI/CD pipelines
User Research Synthesis	Interview transcripts, survey responses, behavioral data, user journey maps	Previous research findings, validated insights, persona evolution, pain point trends	Dovetail, Miro, Notion, Video platforms, Typeform
Feature Prioritization	User feedback, market research, technical constraints, business objectives	Feature performance history, development velocity, user adoption patterns	JIRA, ProductBoard, Analytics tools, Customer support platforms
Content Creation	Brand voice guidelines, user personas, performance data, competitive analysis	Content performance metrics, messaging effectiveness, audience engagement patterns	CMS platforms, Social media tools, Email marketing, Analytics, A/B testing tools

Agentic Patterns

Think of agentic patterns as the “playbooks” for putting AI to work in product teams. You can deploy a single sharp-shooting agent for quick wins, orchestrate multiple tools into a seamless end-to-end flow, or build a coordinated system of agents that think and act together. The right pattern isn’t about being more advanced—it’s about picking the one that delivers the most value for the job in front of you.

Simple Agent (Single-Task Automation)

A focused AI agent that automates one well-defined task with clear inputs and outputs.

- Generate mobile checkout design variations using the design system
- Summarizing user research sessions into bullet points
- Drafting UX copy for form fields and error states

Best For:

- Rapid ideation and prototyping
- Low-risk experimentation
- Individual productivity boosts

Orchestration (Chaining Tools & Processes)

An AI agent or workflow that coordinates multiple tools and steps across design, development, and handoff, with human checkpoints.

Scenario example: Lint Figma file → auto-fix safe changes → generate validated variant pack → create Jira tickets with specs → produce Storybook components → update docs → Slack summary.

Best for:

- Design-to-code automation
- Embedding AI into standard workflows
- End-to-end asset creation

Multi-Agent

A network of AI agents, each specializing in part of the workflow, sharing context, memory, and tools for more complex reasoning and execution.

Research agent analyzes analytics + past tests for recommendations → Design agent generates aligned flows with annotated rationale → Dev agent produces component code, updates DS, and runs visual regression tests

Good for:

- High-context, strategic decisions
- Complex, multi-step initiatives dependent on org knowledge
- Continuous optimization loops across design, product, and engineering

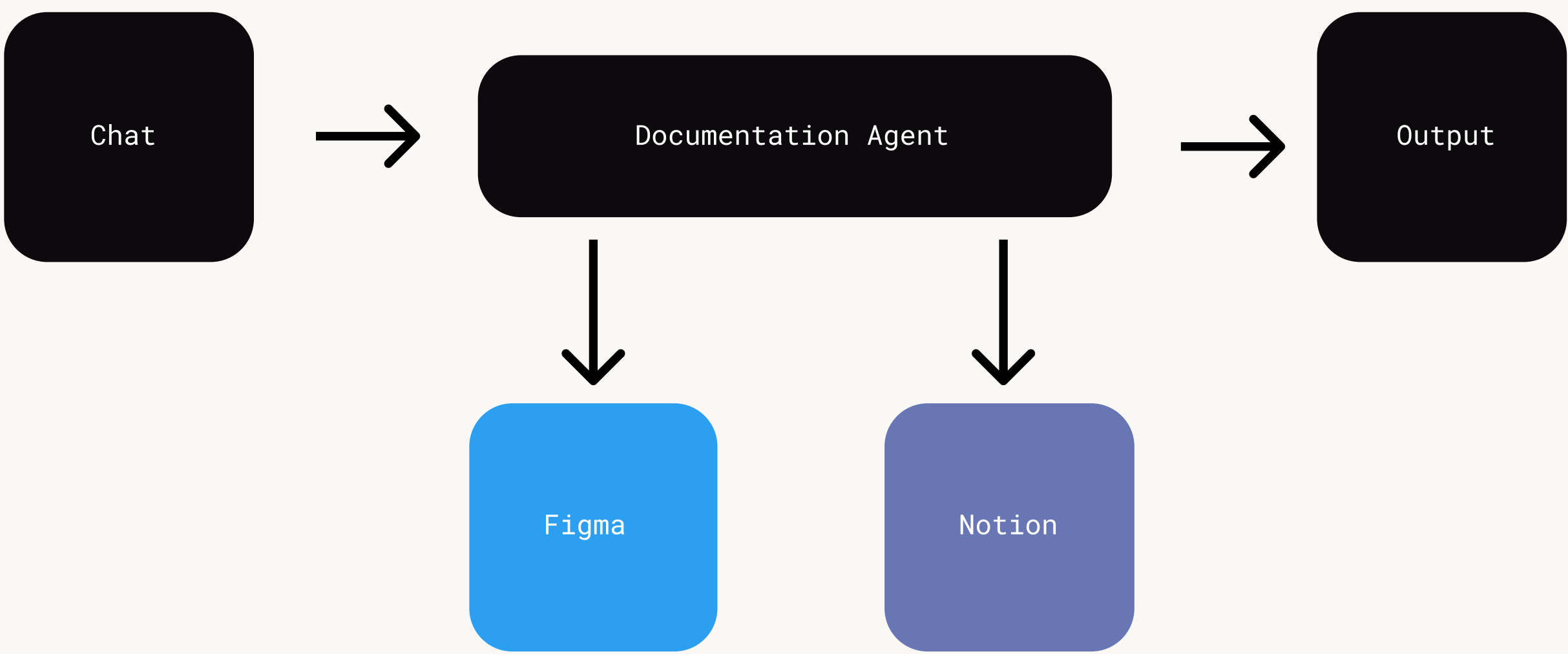
Agentic Patterns (continued)

Context

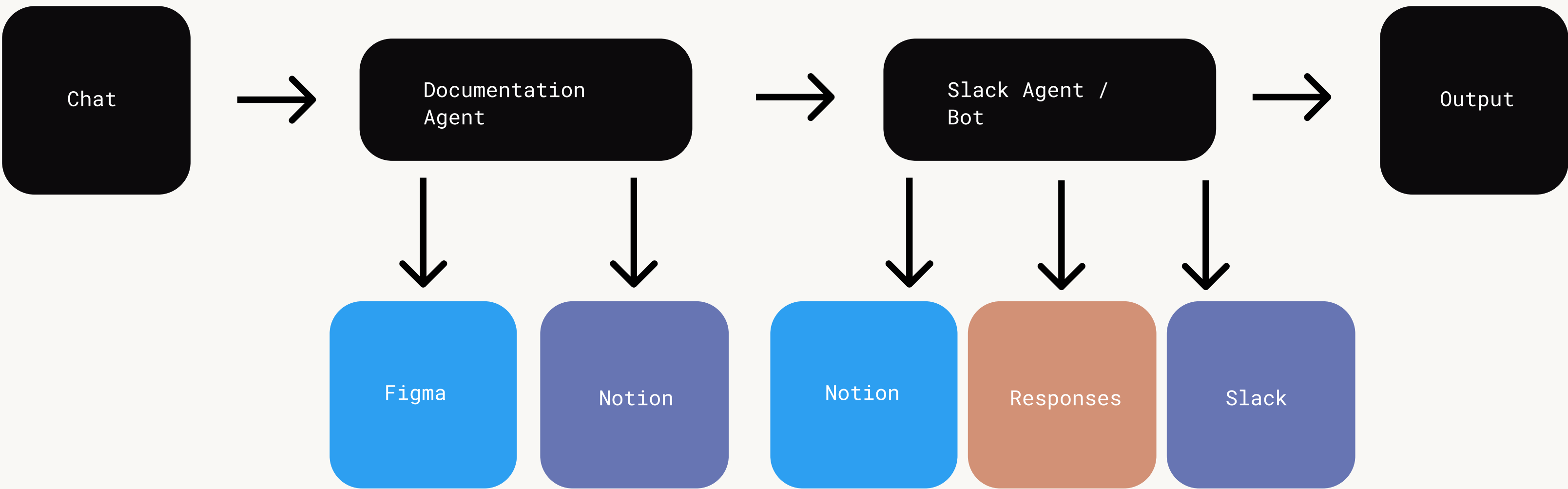
Tools

Memory

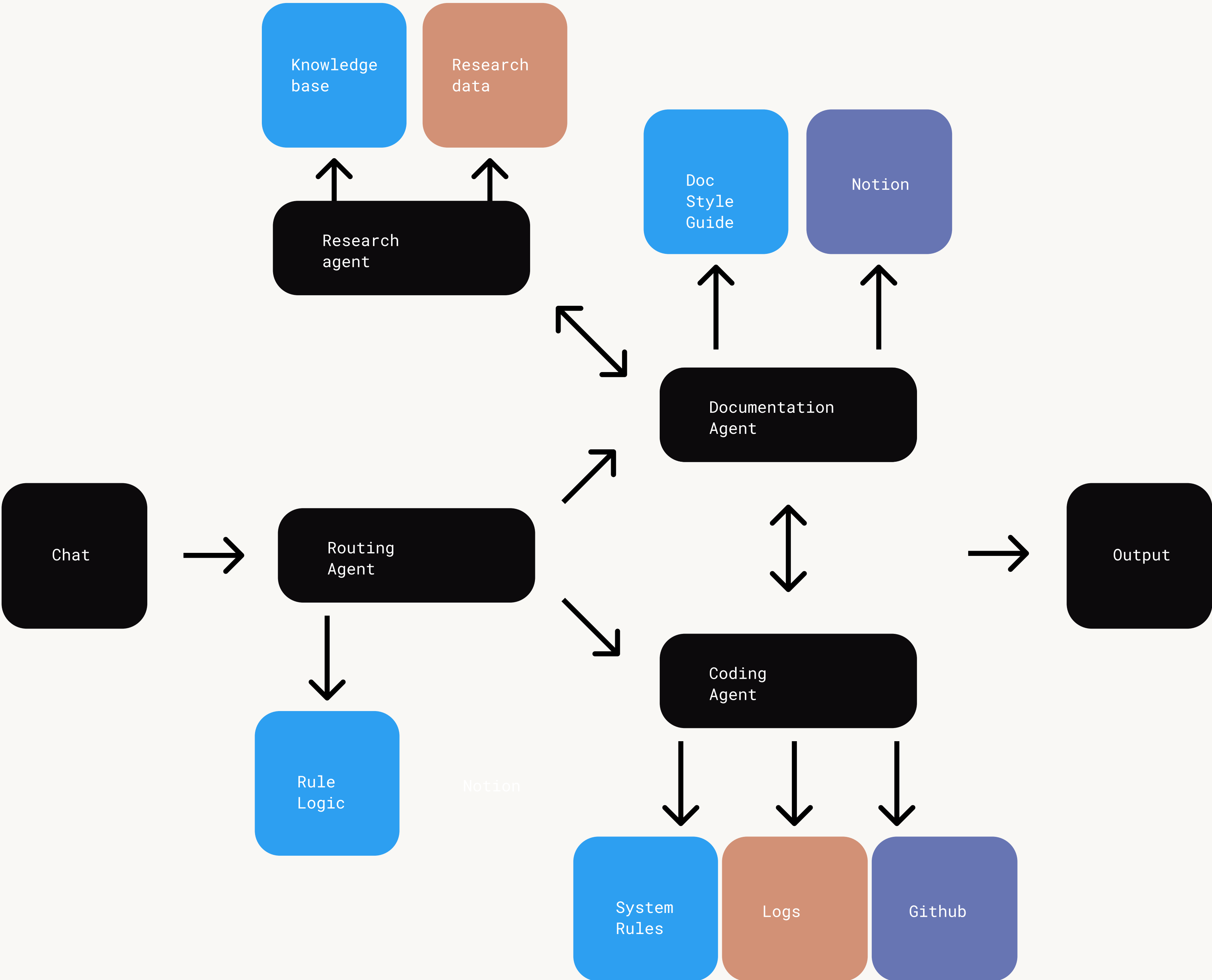
A single agent takes an input, processes it through one well-defined action, and produces a clear output. For example, the agent may draw from a design system to create the output directly, optionally updating connected tools like Figma or Notion along the way.



Orchestration - Here, multiple specialized agents and tools are chained in sequence. One action triggers two events: documentation updates and Slack updates sent. Human checkpoints ensure quality and strategic alignment at each stage.



Multi-Agent - A network of agents shares context, memory, and tools to solve more complex, high-context problems. A routing or director agent may coordinate the work based on inputs and context. Here, a research agent analyzes data and feeds insights to documentation and dev agents, which update design systems, generate code, and produce outputs. Each agent specializes, but together they can make decisions based on defined constraints, and produce results of a team of humans working together.



Good context and good instruction

Every effective AI interaction relies on good context and instruction.

- **Prompt engineering** is about how you tell the AI what to do—crafting clear, unambiguous instructions that guide it toward your desired outcome.
- **Context engineering** is about what the AI knows before it starts working—how you gather, structure, and deliver relevant information so it can make informed, accurate decisions.

When done well, these two things work together to turn AI from a generic assistant into a highly specialized collaborator that “gets” your product, your customers, and your goals.

Context engineering: giving AI the right world to work in

What it entails:

- Identifying all relevant data sources: structured (design tokens, analytics) and unstructured (user interviews, decision logs).
- Organizing and encoding that information so it’s accessible to AI—via embeddings, metadata, or direct tool integrations.
- Maintaining freshness so the AI’s “mental model” matches reality.

Who does it:

- AI Operations teams (maintaining prompt libraries, connecting data sources).
- Design Systems and Product Ops teams (embedding intent, rationale, and standards into components and documentation).
- Engineers (integrating APIs, setting up retrieval pipelines).

Prompt engineering: telling AI exactly what you want

What it entails:

- Defining the task in plain, unambiguous language.
- Providing examples, constraints, and success criteria.
- Breaking down complex requests into step-by-step interactions.

Who does it:

- Everyone!
- Product Managers (framing business objectives and decision criteria).
- Designers (providing examples, style guidance, and acceptance standards).
- Engineers (translating prompts into repeatable, automated instructions in code).

Why this all matters in AI-first teams

Without good context, AI is guessing. Without good instruction, AI is wandering. Together, context engineering and prompt engineering determine **accuracy**, **relevance**, and **efficiency**.

High-performing AI-first teams treat these as design disciplines, not side chores. They maintain living systems for context and instruction so that every AI interaction—whether a quick content draft or a complex multi-agent workflow—produces work that’s on-brand, on-target, and production-ready.

Context Engineering in practice: User Research Knowledge Graph

Goal: Organize interviews, surveys, and tests into a structured, searchable layer AI can tap instantly.

Capture: personas, pain points, needs, context (device, region), sentiment, severity, and relationships (Pain point → impacts → Persona).

AI can answer: “What frustrates first-time mobile users?” “Which pain points hurt NPS for premium customers?”

Setup: central repository with tagging, NLP for theme extraction, vector database for semantic + structured search, regular ingestion to keep insights fresh.

Selecting your tool stack

We generally don’t like to talk about AI tools too early. In fact, we think AI “influencers” have placed too much focus on tools over foundational AI concepts. It leads to general news fatigue when you’re trying to stay updated on the latest releases.

There is no universal “best” tool stack. The right choice depends on your context. What’s the job at hand? Does the tool actually solve the problem you have, or are you forcing your workflow around it? Is it intuitive enough for your team to adopt without endless training?

Not to mention - Can it meet your organization’s data, governance, and accessibility standards without creating new risks? Can it connect seamlessly to your existing environment, or will it become an island?

The most valuable toolkit is one that’s **easy** for your team to use, **safe** for your business, and deeply **embedded** in your workflows—not the one with the most features on the comparison chart.

That being said, here are just a few of the tools we see, use, and recommend when it comes to AI-First Product Teams.

Frontier LLMs:

Core large models that power most AI-first workflows. Today’s top models* include **ChatGPT-5, Claude Sonnet and Opus, Gemini 2.5 Pro, and ManusAI** for specialized reasoning. Many teams run multiple models in parallel to compare outputs or route different tasks to the best performer.

Communication & Knowledge:

We are fans of **NotionAI** if you’re already using Notion for documentation and quick planning. **Granola** and **Fathom** for note-taking, and **WisprFlow** for dictating - a game changer for stream of thought prompting.

Orchestration

Frameworks and platforms for chaining tools, data, and models together. **LangChain, LangFlow, Zapier,** and **n8n** are go-tos for building agentic workflows. For quick no-code setups, **Cassidy** makes multi-tool AI agents accessible to non-technical teams.

Prototyping

Rapidly generate interactive designs or production-ready UIs from prompts. Tools like **Lovable, Bolt, V0.dev, Figma Make** and more recently, **Subframe** let you move from idea to interactive mockup in minutes.

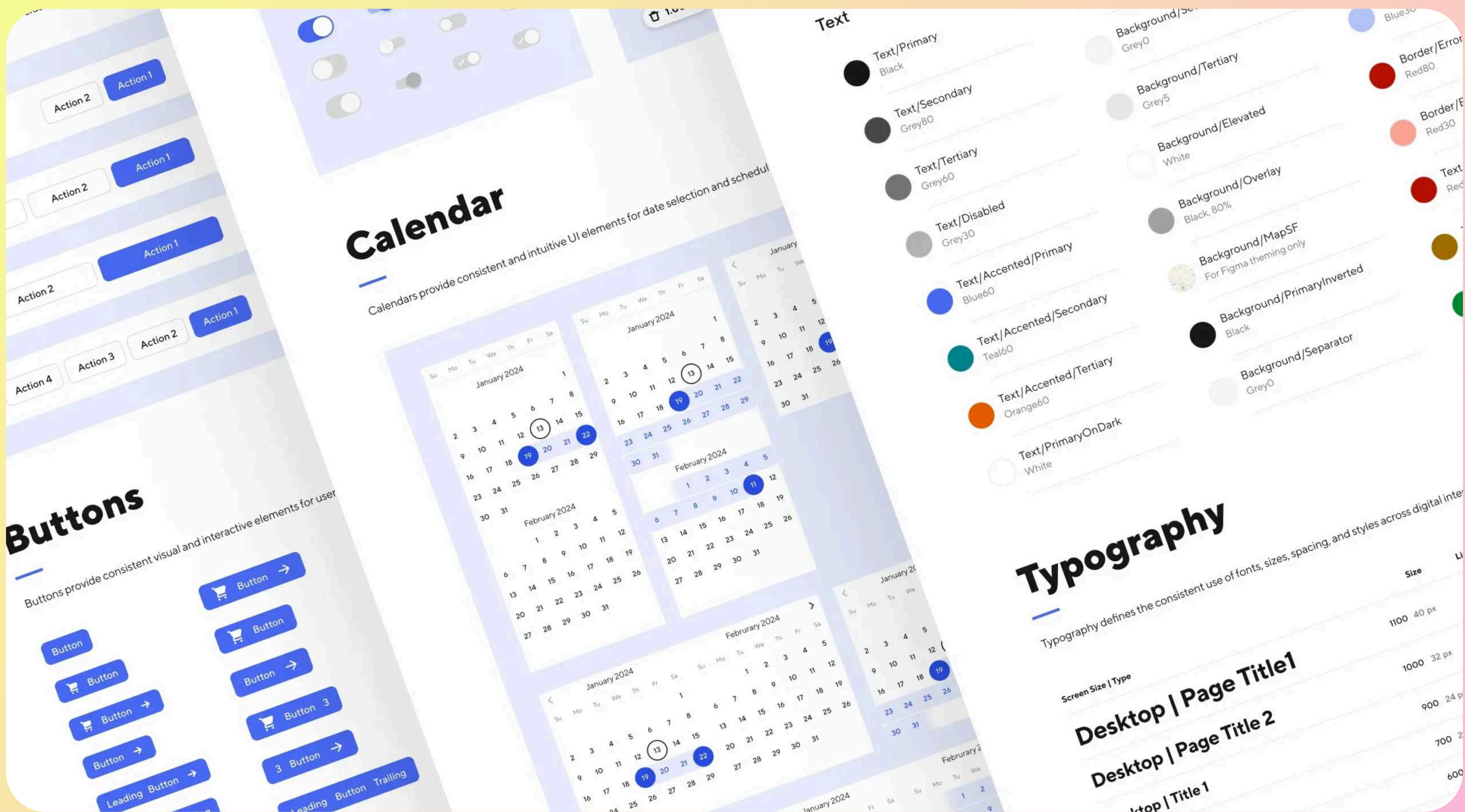
IDEs & Coding

AI-native development environments and copilots boost engineering velocity. **Replit, Cursor,** and **Claude Code** (via Anthropic) assist with scaffolding, debugging, and refactoring. GitHub **Copilot** remains a staple in VS Code

Product & Planning

Planning and prioritization tools with AI embedded. **Jira** with Atlassian AI, **Linear** with AI issue summarization, and **Productboard** with AI feedback clustering all help PMs focus on decisions, not admin. Many PMs also keep **ChatPRD** in their arsenal for generating requirements

*as of August 2025



Designing with Intent: Design Systems as a Force Multiplier

Design systems have emerged as one of the highest-leverage investment for organizations pursuing AI-first product design. Design systems represent the most robust set of structured data and context that many product organizations have, making them a true force multiplier for product teams when properly configured for AI consumption.

But here's the thing. Most design systems today are built for humans—they help designers and developers align, but they don't speak well to intelligent systems or agents. When we say agents, we mean AI tools that can generate, evaluate, and act, whether that's a design assistant, coding copilot, or documentation generator.

For AI tools to work well, we need something more than just visual structure—they need design intent.

Being AI-ready means your system communicates intent, not just appearance. It's not just a collection of buttons and cards—it's a structured interface that AI can reason with. This requires a fundamental shift from describing how components look to encoding why they exist and how they behave.

Core Components of an AI-Ready Design System

Building an AI-ready design system isn't about adding a few new tools—it's about structuring your system so both humans and AI can understand, maintain, and extend it. That means organizing components in a way AI can reason, encoding intent alongside appearance, and keeping everything connected from design through to code. The principles below outline the key building blocks to make that possible.

Semantic Structure

Components are named for their function and purpose, not just their appearance. Instead of 'Button-Blue-Large,' a semantic name might be 'Call-to-Action-Primary' or 'Navigation-Link-External.' This allows AI to understand what a component is and what it does within the user journey, rather than just how it looks.

Token Architecture

A relational token architecture defines the hierarchy, themes, and behaviors of design elements. This provides a structured way for AI to understand the design language and apply it consistently. Tokens can represent anything from color values and typography scales to spacing units and animation durations, all linked to their semantic meaning. This allows for dynamic adjustments by AI while maintaining design integrity.

Centralized Documentation

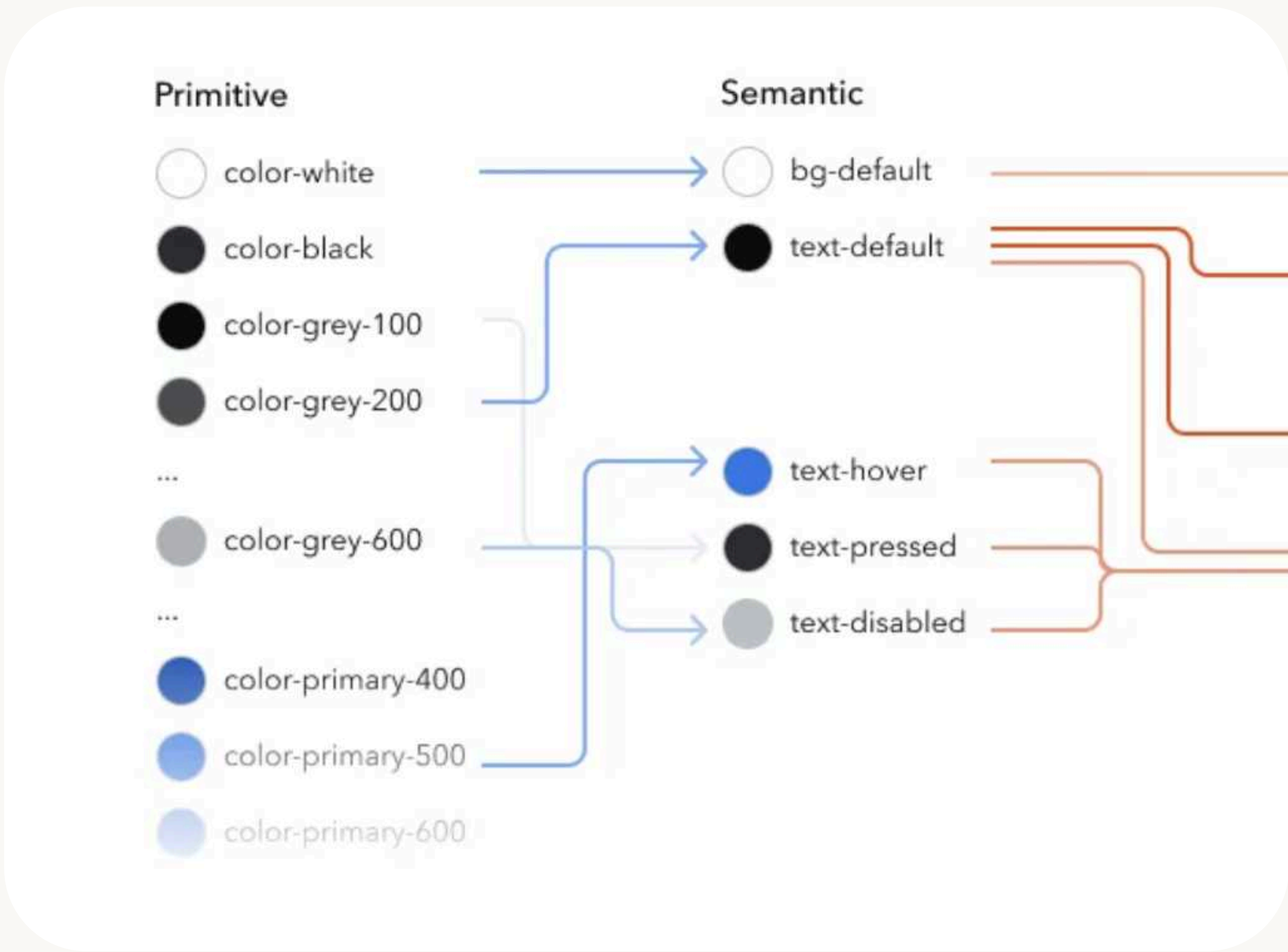
Usage rules, rationale, and standards are embedded directly into the system, rather than being scattered across various documents. This provides a single source of truth for both humans and AI. An AI can query the system to understand when a specific component should be used, its accessibility requirements, or its performance implications.

Connected Design-Code Parity

A shared truth between design and code is established through Multi-Component Platforms and agentic tooling. This ensures that what is designed is accurately reflected in the final code, and vice versa. This continuous synchronization is vital for preventing design drift and ensuring consistency across the product lifecycle.

Encoded Design Intent

This represents perhaps the most critical aspect of an AI-ready design system. It involves embedding functional, contextual, behavioral, business, and accessibility metadata into design components. This rich metadata provides the necessary context for AI to make intelligent decisions, allowing it to move beyond simple pattern matching and generate truly context-aware and meaningful designs.



An example of semantic structure in your design system. While this practice has been around for a while, it's a foundational step towards adding meaning to your design system

Practical Example: AI-Ready vs Traditional Design Intent

Consider the difference in how a button component might be defined:

Traditional approach: `color: blue, size: large, font: Arial`

While this is useful for human designers, an AI would struggle to understand its purpose or how it should behave in different scenarios.

AI-ready approach enriches this with intent-based metadata:

- Functional Intent: `action: primary, purpose: submit_form`
- Behavioral Intent: `on_click: navigate_to_confirmation_page, on_hover: subtle_glow_effect`
- Business Intent: `goal: increase_conversion_rate, metric: click_through_rate`
- Accessibility Intent: `aria_label: Confirm Order, keyboard_navigable: true`

This level of detail allows an AI agent to not only replicate the button's appearance but also understand its role in the user interface, its impact on business goals, and its adherence to accessibility standards.

Here's another example with full code context:

Not AI-Ready: Presentation without Purpose

This example focuses only on how the UI looks. It lacks metadata, intent, or contextual cues—making it difficult for AI to reason about or extend the component meaningfully.

```
<SectionCard variant="elevated" spacing="lg">
  <TextBlock>
    <TitleText level={2}>Welcome to Acme</TitleText>
    <BodyText>
      Set up your account in just a few minutes.
      We'll walk you through everything.
    </BodyText>
    <PrimaryButton variant="solid" color="blue">
      Let's Go
    </PrimaryButton>
  </TextBlock>
</SectionCard>
```

AI-Ready: Intent-Rich and Context-Aware

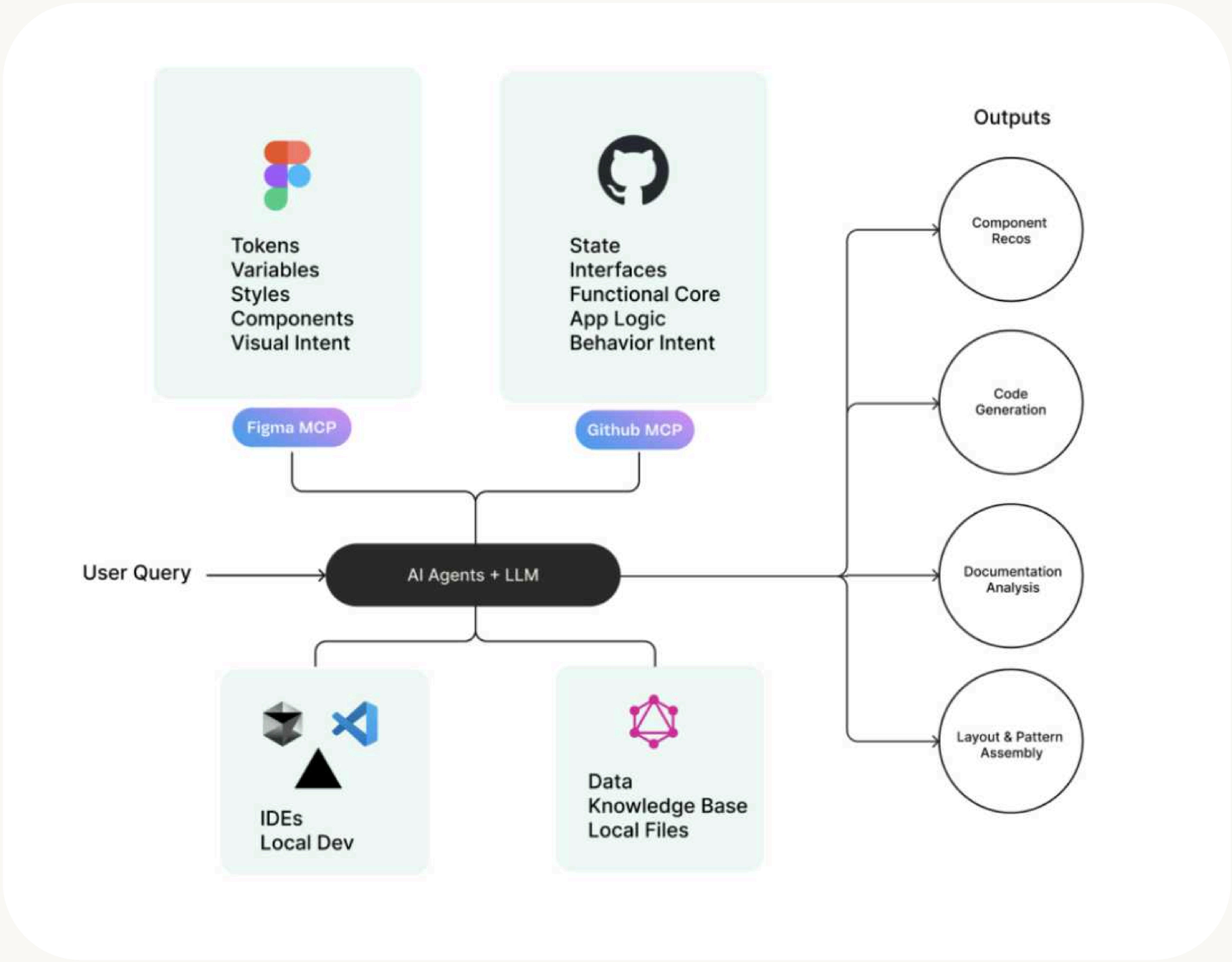
This version embeds functional, behavioral, business, and accessibility intent—creating a structure that both humans and intelligent agents can understand and act on.

```
<OnboardingStep id="account-setup" step={1}
  aria-labelledby="account-setup-heading">
  <StepIntroduction
    headingId="account-setup-heading"
    title="Set Up Your Acme Account"
    description="We'll guide you through creating
    your profile, preferences, and team access."
  />

  <PrimaryActionButton
    type="submit"
    appearance="primary"
    data-intent="begin-onboarding"
    aria-label="Begin setup process"
  >
    Begin Setup
  </PrimaryActionButton>
</OnboardingStep>
```


AI Agent Architecture: AI as a Central Broker

A sophisticated AI agent architecture positions AI as a central broker within the product development ecosystem. This agent can access and synthesize information from disparate sources, such as Figma for design files, GitHub for code repositories, user feedback platforms, and analytics tools. By acting as an intelligent intermediary, the AI agent can facilitate seamless communication and data flow between different tools and teams, generating outputs like code snippets, design documentation, or test cases.



New Ways of Working: The AI-First Product Lifecycle

We're all familiar with the typical product development cycle: extensive planning sessions, detailed requirements documents, lengthy design phases, and careful risk mitigation at every step. These methodologies made perfect sense when design and development took months, when changing direction was expensive, and when teams had to reduce risk through comprehensive upfront planning. But that world no longer exists.

Today, when AI can generate ten design variations in the time it used to take to schedule a planning meeting, when prototypes can be built and tested in hours rather than weeks, these traditional approaches become obstacles rather than enablers. We're clinging to methodologies designed for scarcity in an age of abundance—scarcity of options, scarcity of iteration speed, scarcity of experimentation capacity.

The Limitations of Traditional Approaches

Waterfall methodologies, with their emphasis on comprehensive upfront planning and sequential phases, become counterproductive when AI can generate multiple solution variations in minutes. Agile and Scrum, despite their iterative nature, often devolve into artifact-heavy processes in enterprise environments. Teams get trapped in endless sprints with two-week cycles that prioritize process compliance over meaningful shipping. The focus shifts from solving user problems to managing backlogs, estimating story points, and maintaining ceremony-heavy workflows that slow rather than accelerate innovation.

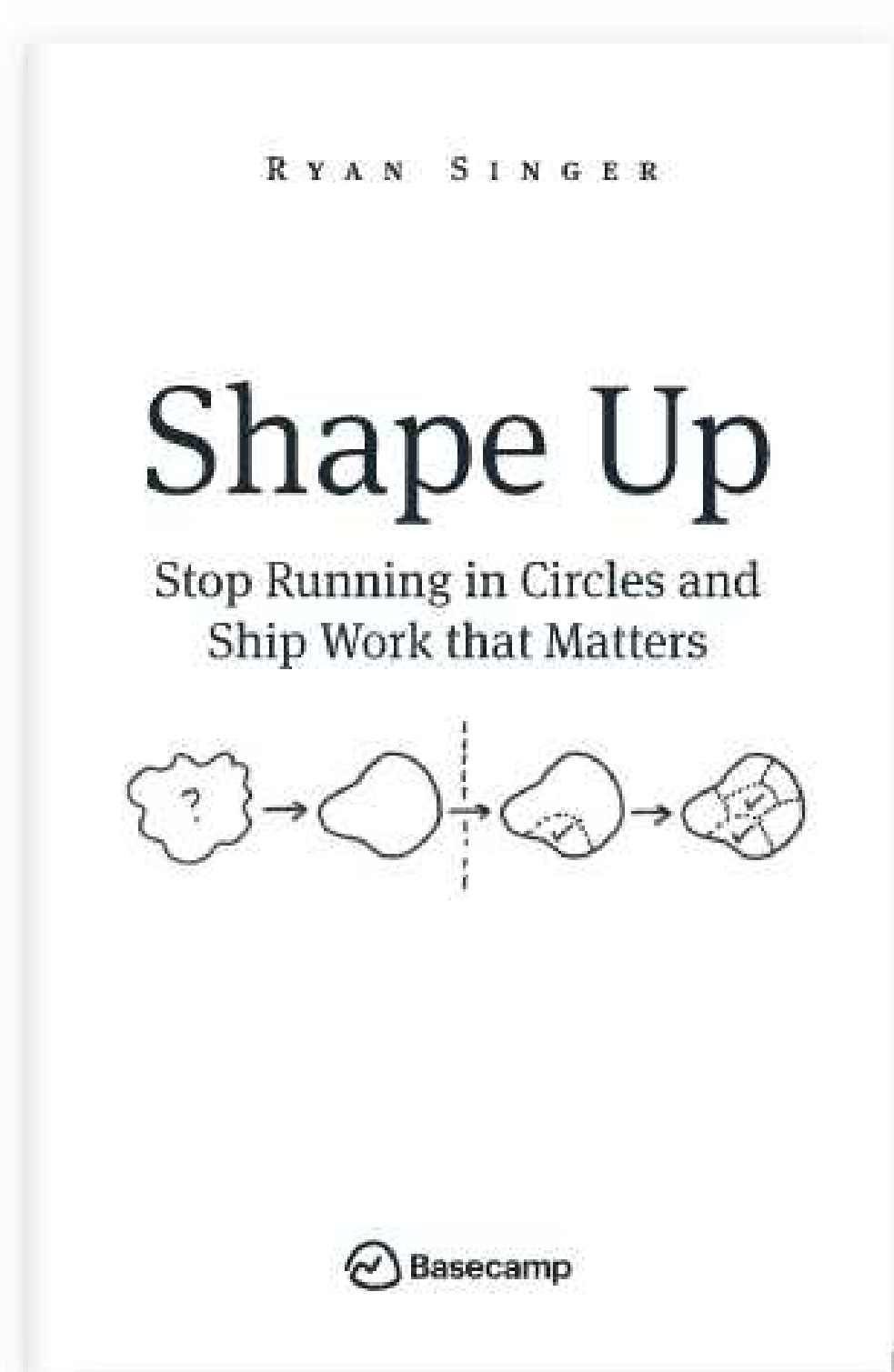
These traditional approaches optimize for predictability and control, but AI-first product development requires optimizing for learning velocity and adaptation speed. When teams can prototype and test ideas in hours rather than weeks, the bottleneck shifts from implementation to problem definition and strategic direction.

The Prototype & Prune approach

AI-first product development embraces what we call the "prototype and prune" approach. This flips traditional thinking, instead of spending extensive time planning what to build, teams rapidly generate multiple prototypes and ruthlessly eliminate what doesn't work. This methodology treats prototypes as disposable learning tools rather than precious deliverables, allowing teams to explore solution spaces that would be impossible to investigate manually.

The power lies in volume and speed of experimentation. When AI can generate ten landing page variations, five user flow options, and multiple content approaches simultaneously, teams can test more hypotheses in a single day than traditional methods allow in weeks. The focus shifts from perfecting individual artifacts to identifying patterns across multiple experiments and iterating toward optimal solutions.

This approach requires comfort with ambiguity and a willingness to discard work quickly. Teams must develop judgment about which prototypes merit further development while maintaining momentum through rapid iteration cycles. The methodology succeeds when human creativity focuses on problem definition and solution direction while AI handles the mechanical work of variation generation and testing.



The Shape Up methodology, published in 2019 by Ryan Singer of Basecamp, provides an ideal framework for AI-first product development. Unlike traditional agile approaches that fragment work into small tasks, Shape Up emphasizes collaborative, senior-led "shaping" sessions that define problems and solution paths before execution begins.

Shape Up's core principle is "appetite-driven" development—teams set fixed timeboxes (usually six weeks maximum) and vary scope instead of expanding timelines. Although we've seen teams move quicker than this.

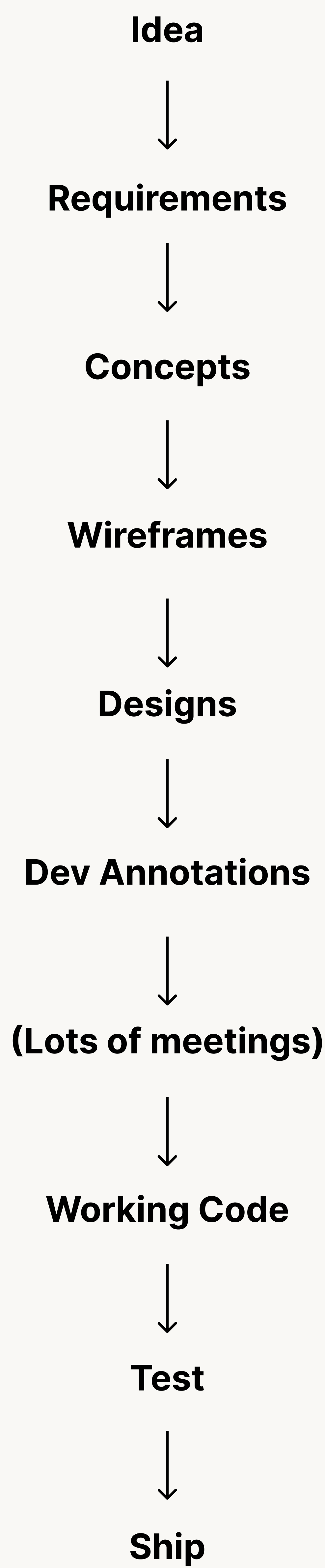
This approach aligns perfectly with AI-first thinking because it requires teams to deeply understand problems before deploying AI to generate solutions. The shaping process involves senior members like product leads and senior engineers who possess technical knowledge coupled with user- and product-oriented thinking.

The methodology operates in three distinct phases: **Shaping** (defining the problem and solution boundaries), **Betting** (choosing which projects to pursue), and **Building** (executing within fixed time constraints). Rather than maintaining endless backlogs, important ideas naturally resurface when they're truly valuable, while teams focus on meaningful work rather than backlog management.

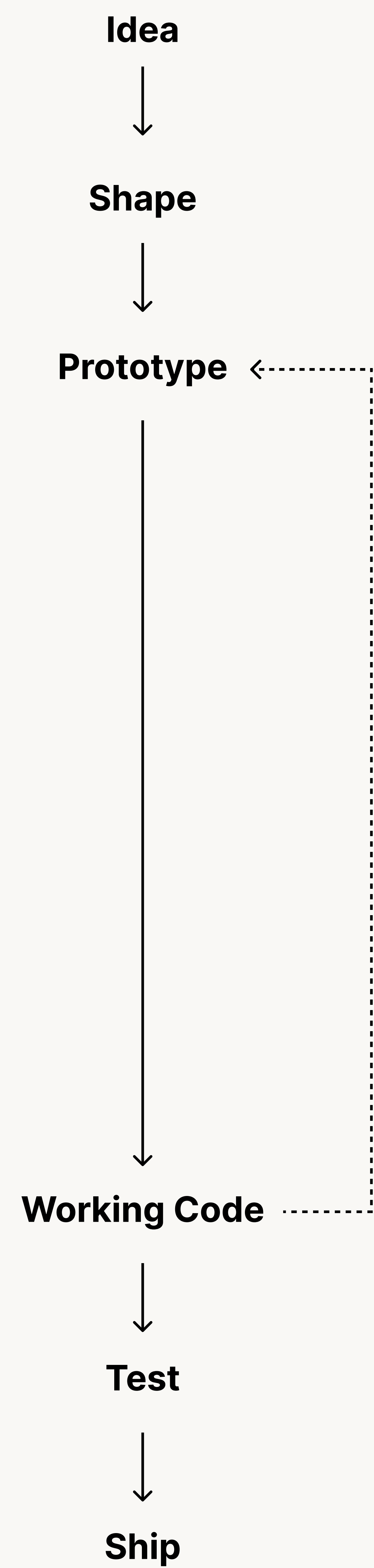
How AI-First Teams Build Differently

Instead of long, artifact-heavy cycles, AI-first teams move from shaping ideas to working code in days—using rapid prototypes as learning tools, not gated deliverables.

Traditional Approach



AI-First Approach



The Future of Product Design will be non-linear and multi-modal

Traditional product design assumes predictable user journeys—we map happy paths, design for specific use cases, and create interfaces that guide users through predetermined flows. AI-first product strategy fundamentally challenges this approach.

In an AI-first world, experiences emerge from the interaction between user intent, available context, and AI capabilities. Rather than designing fixed paths, we design adaptive systems that can respond intelligently to infinite combinations of circumstances. The strategic advantage comes not from predicting every scenario, but from building systems sophisticated enough to handle unpredictable ones gracefully.

This represents a profound shift from deterministic to probabilistic design thinking. Where we once optimized for efficiency along known paths, we now optimize for adaptability across unknown territories.

Shifting Artifacts & Methods

Instead of: User Journey Maps with Defined Touchpoints

Do: Intent Mapping with Context Variables and Adaptation Rules

Traditional journey mapping assumes linear progression through known touchpoints. AI-first intent mapping identifies user goals and maps how different contexts should trigger different system responses. Map "User intent: access insights" with variables like location, time, device, urgency, then define how the system adapts to serve that intent optimally in each context.

Instead of: Detailed Wireframes for Every Screen State

Do: Component Behavior Guidelines and Response Patterns

Rather than designing every possible screen, create comprehensive guidelines for how components behave in different contexts. Document response patterns like "When user attention is limited, prioritize actionable insights over detailed data" or "In voice-only contexts, provide confirmation before executing destructive actions."

Instead of: Linear User Stories ("As a user, I want...")

Do: Contextual Scenarios ("When context X occurs, enable Y")

Traditional user stories assume predictable needs. Contextual scenarios acknowledge that the same underlying need might be best served differently depending on circumstances. "When user needs project status AND is in a meeting AND using mobile, surface critical blockers with one-tap escalation" vs "When user has focus time AND using desktop, provide detailed analytics with drill-down capabilities."

“

"We are now moving to a world where we do not fully design the interface anymore for people who experience it, because we are going multimodal, multi-form factor, dynamic interfaces. To be able to control where things are going, you actually have to let go of control of that instinct."

-David Hoang, Head of AI
@ Atlassian

Craft through Speed, not despite it

Traditional product development treats speed and quality as competing forces—faster delivery often means compromising craft. AI-first product strategy inverts this relationship.

AI velocity enables deeper craft by allowing teams to explore more sophisticated hypotheses and iterate on higher-fidelity concepts. The strategic advantage comes from using speed to increase the quality of exploration, not to rush inferior solutions to market. Teams can afford to be more selective because they can explore more options at higher fidelity in less time.

This means teams can elevate expectations because they can afford to be more selective. Decision-making becomes more empirical and evidence-based because you can rapidly test sophisticated concepts rather than debating specs. Timeline planning shifts from linear schedules to exploration phases with multiple parallel tracks and continuous refinement loops. We'll discuss this more in a later section.

Shifting Artifacts & Methods

Instead of: Single Concept Development with Limited Iteration

Do: Parallel Exploration of Multiple Refined Directions

Traditional process develops one concept through sequential refinement stages. AI-enabled process explores 3-5 sophisticated directions simultaneously, each at high fidelity. Use AI to rapidly generate variations on core concepts, then invest human creativity in refining the most promising directions. This creates a portfolio approach where the final solution emerges from comparative evaluation rather than linear development.

Instead of: Detailed Upfront Specifications

Do: Rapid High-Fidelity Prototyping with Iterative Refinement

Replace lengthy specification documents with working prototypes that demonstrate actual behavior. Use AI to quickly generate functional prototypes from rough concepts, then iterate based on interaction rather than documentation. This shifts discussion from "Will this work?" to "How well does this work?" and "How can we make it better?"

Instead of: Resource Allocation for Single Bets

Do: Portfolio Approach to Concept Development

Allocate resources across multiple parallel explorations rather than committing early to single solutions. Use AI to reduce the cost of exploring alternatives, allowing teams to maintain multiple viable options longer into the development process. Make final decisions based on comparative testing rather than early commitment.

“

"Speed isn't the point. Craft is. Clarity is. AI gives you the space and time to go deeper. AI shouldn't be a shortcut - it should be your FLEX"

-Romina Kavcic,
thedesignteam.guide

From Problem-Solution Fit to Problem-Space Expansion

Traditional product strategy starts with known problems and seeks viable solutions within existing constraints. Teams focus on finding product-market fit for problems they can already see and solve. AI-first product strategy fundamentally expands this approach.

AI capabilities make previously impossible problems suddenly solvable, creating entirely new categories of value creation. The strategic advantage shifts from efficiently solving known problems to discovering newly accessible problem spaces that competitors haven't yet recognized.

This requires expanding from solution-oriented thinking to capability-oriented exploration. Success comes not just from building better solutions to existing problems, but from identifying problems that become solvable only with AI augmentation. This represents a shift from competitive optimization to category creation through expanded possibility spaces.

Shifting Artifacts & Methods

Instead of: Problem Statements with Known Solution Boundaries

Do: Problem Space Mapping with AI Capability Overlays

Create a Problem Impact Matrix- List all user frustrations regardless of current solvability. Rate by impact (1-10) and frequency (1-10). Include everything users complain about or work around.

Map AI Capability Overlays - For each high-impact problem, identify required intelligence type (pattern recognition, language processing, prediction). Research emerging AI capabilities in those areas.

Identify Newly Solvable Problems - Focus on high-impact problems where AI capability development is accelerating. Document specific technologies that could enable solutions.

Instead of: User Needs Assessment Within Current Constraints

Do: Latent Need Discovery Across Expanded Possibility Space

Speculative Design Sessions - Run workshops where users design "impossible" ideal workflows. Use prompts like "If you had a perfect AI assistant, what would your workday look like?"

Future Scenario Research - Create 2-3 year AI transformation scenarios. Interview users about these: "If this were possible, how would it change what you do?" Identify elements generating strongest excitement.

Backwards Capability Mapping - Start with ideal scenarios, work backwards to required capabilities. Map each to current/emerging AI technologies. Identify which desires are closest to becoming feasible.

“

"AI doesn't just help solve existing problems—it expands the realm of what problems are solvable. This means your problem space is no longer constrained by old limitations."

-Brian Balfour, CEO @ Reforge

To Vibe or not to Vibe

Vibe coding—jumping into a code-capable AI tool and just building something—is becoming a hallmark of the AI era. It’s not about detailed specs or perfectly structured user stories. It’s about sitting down with an idea and letting AI help you feel your way into a working prototype.

Vibe coding is about embracing imperfection while you build. You’re not obsessing over best practices or code elegance—you’re focused on getting something functional now. That’s a strength for early creativity.

But there’s a flip side: many people vibe code without knowing how things are working under the hood. For product teams, we agree with the former but think the latter matters—especially if you plan to grow or maintain what you’ve built. That’s why understanding its limits is key.

Example: You might end up with an AI-generated app that works beautifully for a demo—but the code is scattered across unorganized files, routes are hard-coded, and it’s storing API keys straight in the client side. It runs fine in your local environment, but the moment you try to deploy it or onboard another developer, you’re fighting spaghetti code and security holes you didn’t know you had.

When Vibe Coding Shines

- Raw ideation, without friction – Perfect for getting from “idea in your head” to “something you can click, tap, or test” in minutes.
- Lowering the barrier for non-technical creators – PMs, designers, marketers, and ops folks can express ideas in code without waiting for an engineering cycle.
- Learning by doing – The best tools show you how they scaffold, reason, and generate code, so you build literacy as you build products.
- Fast prototyping... to a point – Great for early proofs-of-concept, experiments, or quick demos before committing serious resources.

When to Step Back

- When stakes are high – Anything security-sensitive, highly regulated, or core to revenue should move beyond “vibe” into disciplined engineering.
- When scale is coming – Quick hacks rarely have the architecture to handle growth.
- When the cost of failure is high – You don’t want a “just felt right” build running payroll.

Redefining Team Structure and Roles

The traditional three-legged stool of product development—design, product management, and engineering—was built for a world where each discipline had clear boundaries and sequential handoffs. AI doesn't just augment these roles; it fundamentally reshapes how they collaborate, what they're responsible for, and what new capabilities teams need to operate effectively. When AI can generate code, create design variations, and analyze user data simultaneously, the question isn't how to make existing roles more efficient. It's how to reorganize teams around the unique value that humans and AI each provide, and what entirely new functions emerge to orchestrate this collaboration.



How AI Transforms the Traditional Triad

Product Managers

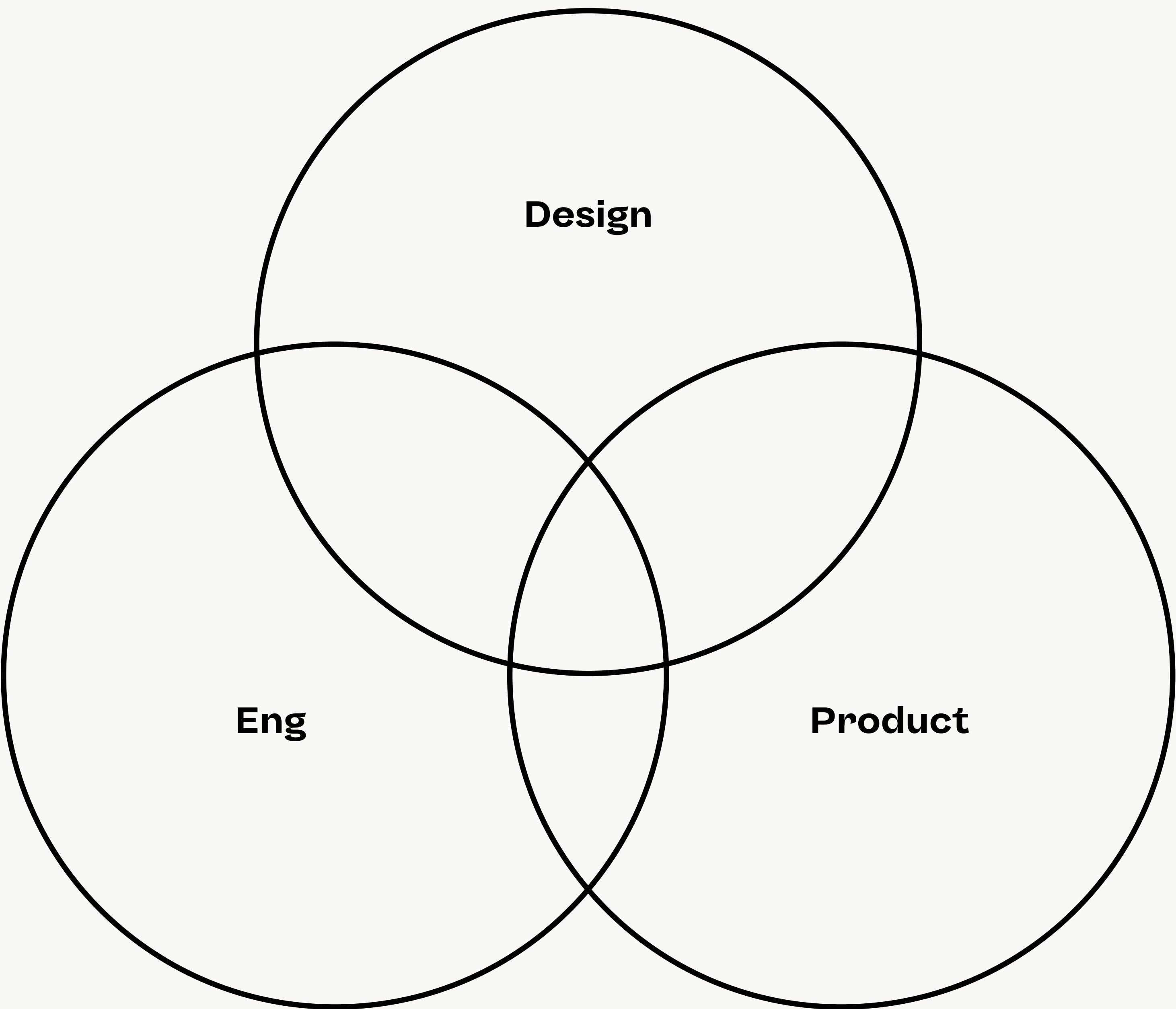
Product Managers evolve from primarily coordinating tasks to acting as strategic architects. Instead of focusing on backlogs and detailed requirements, they concentrate on defining problems, creating business context, and guiding AI systems to deliver outputs that align with objectives and user needs. This shift calls for deeper technical understanding of AI's capabilities and limitations, along with skills in prompt engineering and AI system evaluation.

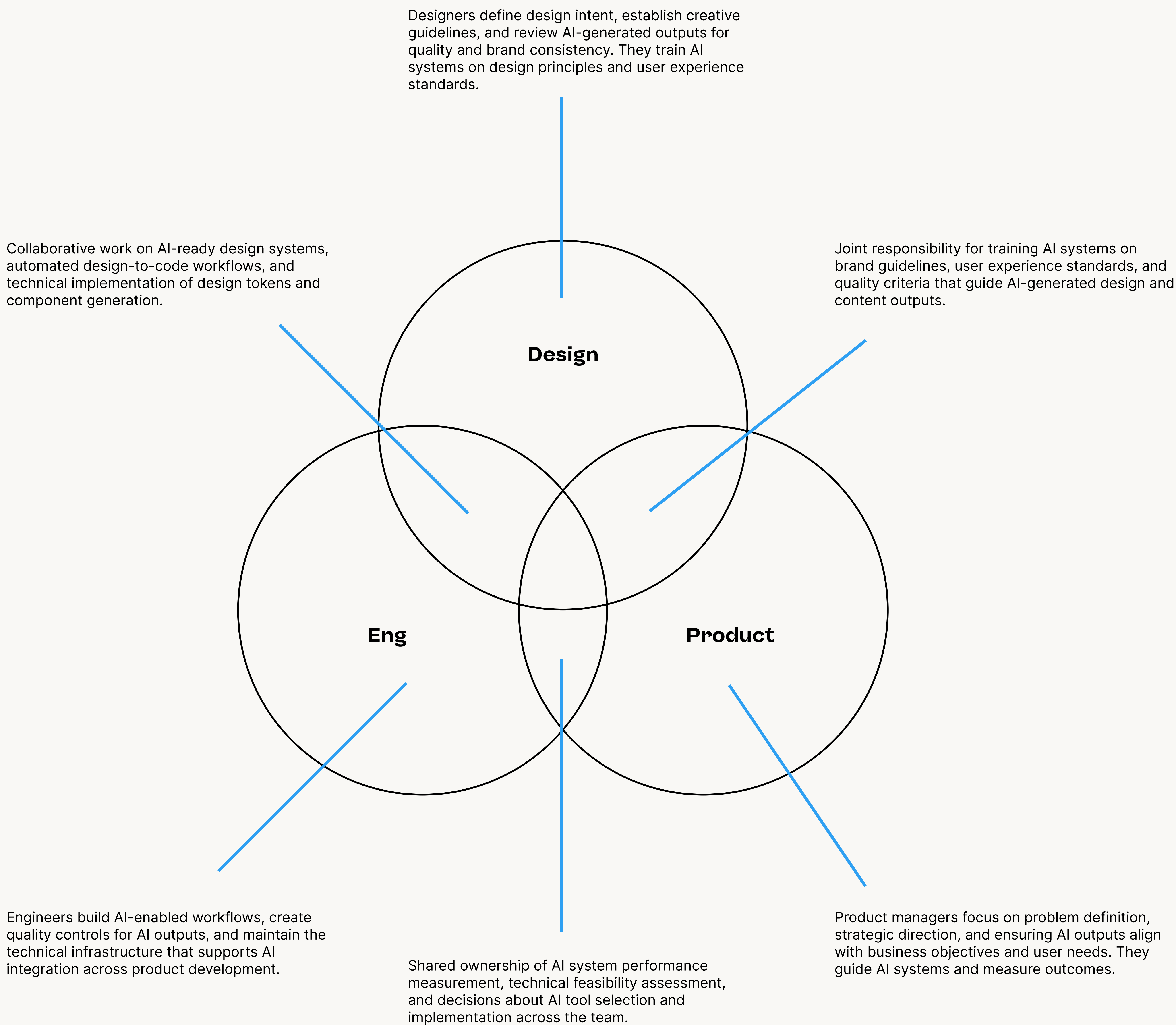
Designers

Designers transition from producing every individual component to shaping creative direction. They define design intent, set parameters, and guide AI in generating outputs that meet quality and brand standards. The role emphasizes curating and refining AI-assisted designs, as well as training AI systems on brand guidelines, design principles, and user experience standards.

Engineers

Engineers expand from building features to architecting AI-enabled systems. Beyond writing code, they design AI-integrated workflows, implement quality controls for AI outputs, and maintain the infrastructure that enables human–AI collaboration. Their work includes prompt optimization, AI model integration, and ensuring AI-generated code meets production standards—enabling designs and products that are both context-aware and high quality.





The traditional rigid boundaries between product, design, and engineering dissolve into fluid, collaborative areas of shared responsibility. AI doesn't replace these core functions—it amplifies their collective capabilities while requiring new coordination patterns, shared standards, and governance frameworks. Teams that master this balance ship better products faster while maintaining the user focus and technical quality that drives long-term success. The center represents the new collaborative core that enables sustainable AI-first product development across all functions.

The Product Manager's AI Evolution: From Coordinator to Strategic Orchestrator

AI isn't just another tool in the PM toolkit—it's fundamentally reshaping what product managers do, how they create value, and which skills separate the top performers from the rest. The PMs who understand this shift will become force-multipliers for their teams. Those who don't risk becoming obsolete.

The Product Manager's AI Evolution

AI compresses the traditional PM loop of understanding problems, prioritizing solutions, and shipping value by automating much of the "writing, analysis, and coordination" work that consumes PM time today. PMs who spent 60% of their time on documentation, status updates, and administrative coordination can now focus that energy on judgment, strategy, and orchestrating complex cross-functional problems.

This evolution creates a bifurcation in the PM role. Some individual contributor PM positions will collapse as AI-tooled teams require fewer coordinators and can operate with smaller, more autonomous units. But the PMs who evolve into strategic orchestrators—designing learning systems, managing complex stakeholder dynamics, and making nuanced product decisions—become more valuable than ever.

The craft itself shifts from creating perfect artifacts to designing effective systems. Rather than spending weeks perfecting a PRD, AI-first PMs focus on creating workflows that help teams learn quickly, make better decisions, and ship more valuable features. The leverage comes from designing the learning system around the team, not from writing comprehensive documentation.

The AI-First PM Toolkit

Discovery & Research: AI research copilots can synthesize thousands of user feedback entries into actionable themes in minutes, but smart PMs maintain structured discovery habits with strong guardrails. The key is using AI to accelerate analysis while preserving direct customer contact and verifiable evidence.

Strategy & Planning: LLM-assisted PRD drafting provides strong first passes for problem statements, acceptance criteria, and edge cases. Opportunity mapping accelerators can draft jobs-to-be-done frameworks and risk assessments, though they require human validation with real users. Portfolio simulators help model the impact of different feature combinations.

Execution & Operations: AI summarizers collapse customer feedback into decision-ready insights with traceability to original sources. Auto-tagging systems identify themes across support tickets and feature requests. Model-assisted cost and impact estimates provide better baselines for prioritization decisions.

Communication: Automated changelog generation, release note drafting, and executive update creation eliminate routine communication overhead. But the highest-value PM work—crafting narratives that align executives and unlock resources—becomes even more important.

The Emergence of AI Operations

Traditional teams lack the specialized expertise needed to orchestrate human-AI collaboration effectively. This gap creates the need for AI Operations—a new discipline focused on maximizing the productivity and quality of AI-augmented teams.

Technical Operations: Keeping AI-Powered Products Running

Someone needs to handle the practical challenges: "Why did our design system AI generate broken components?" "How do we prevent our user research tool from losing insights?" "What happens when our content generation workflow fails during a deadline?" This function ensures AI tools enhance rather than disrupt productivity.

The work includes managing prompt libraries, setting up quality checks for AI outputs, and building workflows that integrate AI seamlessly into existing tools. They handle API integrations between design tools and AI services, monitor costs across the team, and maintain version control for prompts and configurations. When your AI-powered design-to-code pipeline breaks, these people fix it.

Governance & Quality: Maintaining Standards with AI-Generated Work

Product teams face quality challenges when AI joins the creative process: "How do we ensure AI-generated designs meet accessibility standards?" "What's our review process for AI-written requirements?" "How do we maintain consistency when AI generates variations?" This function creates quality standards and review processes that prevent AI from degrading work.

The work includes establishing review workflows for AI content, creating quality checklists teams follow before using AI outputs, and defining when human review is required versus when AI outputs can be used directly. They develop testing protocols for AI-generated code and maintain oversight processes for AI-assisted decisions.

Process Strategy: Optimizing Team Workflows with AI

This capability answers: "How should AI change the way our product team actually works?" Rather than randomly adopting AI tools, teams identify where AI enhances workflows versus where it creates complexity. They handle questions like "Should sprint planning include AI-generated user stories?" and "How do we restructure design reviews when AI generates multiple variations?"

The work involves analyzing workflows to identify AI opportunities, designing new processes that leverage AI capabilities, and measuring whether AI adoption improves team performance. They manage training programs, change management for workflow modifications, and continuous improvement based on team feedback.

“

While prompt engineering might not be a sustainable role in the future, there is definitely a need for AI Operations. THIS is the big next role for companies.

-Rachel Woods

AI Operations: What’s your most important project right now?

We asked a handful of AI Operators what their priorities were right now. Here’s what they told us.

Reducing Prompt & Model Drift

“Our top priority is building a centralized prompt library. We track how prompts evolve, run A/B tests on variations, and monitor model output drift over time. This keeps our AI outputs consistent, even as underlying models update. One team started with a Google Sheet, then moved to a GitHub repo so prompts could be version-controlled like code.”

Quality Gates for AI-Generated Code & Content

“We’ve implemented multi-stage quality checks before AI outputs make it into production—automated linting, accessibility scans, and human sign-off. This reduced post-launch defects from AI-generated work by 40%.”

Proactive Cost Management

“We track model usage and API costs at a granular level—by team, workflow, and even by prompt—so we can optimize high-traffic automations, swap to smaller models where quality is unaffected, and avoid runaway bills. We’re still figuring out how to balance accountability with giving teams freedom to experiment.”

AI Literacy & Change Management

“We’re working with HR and running a structured AI literacy program for every role, from designers to PMs. The focus is not just tool training, but also judgment: knowing when to trust AI outputs, when to intervene, and how to spot subtle quality issues. One thing we underestimated was how much hands-on proof (and pilot projects) it would take to build trust in the new process.”

Ethics & Compliance Automation

“We built automated compliance checks for regulated content—flagging outputs that might violate legal or brand guidelines before they reach end users. It’s sped up approvals while reducing the burden on legal teams.”

The Emergence of Design Engineering

The Role of Design Engineering in AI-First Teams

AI can produce design variations, code snippets, and prototypes in seconds—but it can’t guarantee that what it generates will be usable, performant, or aligned with the product vision. That’s where Design Engineering comes in. In AI-first teams, this role ensures that AI-driven speed translates into high-quality, production-ready outcomes that meet both creative and technical standards.

From Concept to Production

Design Engineers bridge creative vision and technical execution—interpreting AI-generated outputs, refining them for feasibility, and implementing them in production. They handle complex interactions, micro-animations, and responsive layouts while maintaining fidelity across devices, frameworks, and browsers. This includes spotting where AI outputs need human refinement to meet accessibility, performance, and security requirements.

Prototyping at AI Speed

Rapid prototyping is essential when AI can generate dozens of concepts at once. Design Engineers decide which concepts to develop further, validate them through functional prototypes, and adapt them quickly based on feedback. They balance the exploration power of AI with the discipline of delivering designs that are both desirable and buildable.

Design Systems AI Can Understand

Well-structured, reusable components remain critical—but now they also need to be “AI-readable.” Clear naming conventions, metadata, and consistent documentation not only help human teams work faster, they also help AI tools generate outputs that align with brand and technical standards.

Why It Matters

Without Design Engineering, AI outputs risk becoming throwaway experiments. With it, teams can move from inspiration to implementation without losing fidelity, accessibility, or performance—turning rapid ideas into reliable, scalable product experiences.

Where do these capabilities fit in our team?

In practice, these capabilities map to specific responsibilities that existing team members often take on, with dedicated roles emerging as teams mature their AI usage. The structure varies significantly by team size—small product teams often have one person wearing multiple hats, mid-size teams typically distribute responsibilities across several people, and larger organizations often formalize these into dedicated roles.

Area	Common Titles	Where They Sit
Technical AI Operations	AI Tools Admin, Prompt Engineer, AI Systems Specialist, AI Workflow Engineer	Engineering Team, Platform Team, DevOps
Governance & Quality	AI Quality Lead, AI Manager, AI Compliance Coordinator	QA, Product Ops, Design Ops, Design Systems
AI / Process Strategy	AI Ops Manager, AI Program Lead	Product Ops, Design Ops, Cross-Functional
Design Engineering	Design Engineer, Creative Technologist, UX Engineer	Design, Engineering

Change Management: The Human Side of AI Transformation

Adopting AI isn't like building software. With software, you can spec, build, and ship with predictable outcomes. AI's value depends on how people use it, adapt their work around it, and trust it with decisions.

Most AI initiatives fail for human reasons, not technical ones. The gap isn't whether the model works—it's whether teams can prompt effectively, judge outputs, and integrate AI into daily workflows. Many already dabble with ChatGPT, CustomGPTs, or tool-based assistants, but without structure, adoption stalls.

AI-First teams build AI literacy alongside rollout, start with visible wins that build trust, and set feedback loops that make systems better over time. It's not just tool skills—it's cultural readiness and the judgment to know when to rely on, override, or improve AI.

Build an AI Council

- Cross-functional group (product, design, engineering, operations, compliance) that sets adoption priorities, defines governance, and shares learnings across the org.
- Prevents siloed AI experiments, ensures ethical and compliant use, and aligns investments with business strategy.
- Also acts as the “nerve center” for monitoring emerging tools and deciding when to pilot or scale.

Create Safe-to-Fail Sandboxes

Give teams a dedicated space to test AI workflows with real data but low business risk. This accelerates learning and reduces the fear of making mistakes in production.

Appoint AI Champions

Identify individuals in each team who have both credibility and curiosity. They act as the first line of support, source ideas for new use cases, and feed back adoption insights to the AI Council.

Track AI Adoption Metrics

Don't just track model accuracy or technical KPIs — measure usage rates, time saved, and trust levels across the org. This shows leadership where adoption is working and where it's stalling.

Communicate Wins Relentlessly

Share success stories internally: “This sprint, AI reduced our bug triage time by 60%” or “We shipped a feature 2 weeks earlier thanks to AI-assisted design.” Wins build momentum faster than mandates.

“

"We'll go from makers to managers, from doing the work to learning how to allocate resources—choosing which work to be done, deciding whether work is good enough, and editing it when it's not. It means a transition from a knowledge economy to an allocation economy."

-Dan Shipper, Founder @ Every

The CRAFT Framework for AI Transformation

Rachel Woods, prolific AI strategist and teacher, developed the CRAFT framework for transforming business processes into AI-powered systems. This methodology provides a structured approach to change management that bridges the gap between current operations and AI-first workflows.

The CRAFT framework succeeds because it addresses both technical and human elements of change. By starting with thorough process documentation, teams understand exactly what's changing and why. The incremental approach allows people to build confidence with AI gradually, while the focus on continuous improvement ensures systems evolve alongside user needs and capabilities.

1

Collect

Gather comprehensive information about current processes through structured questionnaires that capture inputs, steps, outputs, deliverables, existing materials, and templates currently in use. This creates the foundation for understanding what needs to change.

2

Refine

Analyze collected information to identify inefficiencies, bottlenecks, and opportunities for AI enhancement. This step involves mapping current workflows against desired outcomes and determining where AI can create the most impact.

3

Automate

Design AI-powered systems that replace manual processes while maintaining quality and control standards. This includes developing prompts, selecting appropriate tools, and creating feedback mechanisms.

4

Fine-Tune

Implement continuous improvement processes that optimize AI performance over time. This involves monitoring outputs, gathering user feedback, and adjusting systems based on real-world performance.

5

Transform

Scale successful AI implementations across the organization while building internal capabilities for ongoing innovation and adaptation.

The ROI of AI-First Product Design

Once teams are ready to work with an AI-First mindset, the next step is proving it's worth the investment—because budgets follow results.

The reason ROI keeps coming up is simple—someone upstairs is asking for it. Whether it's the CEO, CFO, or the board, leaders want to know how AI spend translates into measurable outcomes. Budgets are under scrutiny. The “just try it and see” era of 2024–2025 is ending. In 2026, the bar is higher: AI initiatives must earn their place in the plan by showing a clear return—whether that's in cost savings, growth acceleration, risk reduction, or competitive advantage.

AI's value isn't just about “efficiency” in the abstract—it's about freeing up the right resources at the right time so your team can move faster toward business outcomes. Done well, ROI becomes a feedback loop: efficiency → faster cycles → earlier market entry → greater learning velocity → smarter next investments.

Calculating ROI

There's no one "magic formula" for AI ROI—context matters. Here are common approaches you can adapt:

Time & Cost Saved

One of the most direct ROI measures for AI is labor efficiency. By automating repetitive or manual tasks, you reduce the hours needed to achieve the same output—freeing your team for higher-value work.

Hours saved per week × average hourly rate × number of roles impacted × time period

Example: Automating design specs saves 8 hrs/week for 5 designers. At \$80/hr, over a quarter: $8 \times \$80 \times 5 \times 13 = \$41,600$ saved.

Speed-to-Market Impact

AI can shorten cycle times across design, development, and campaign production. Releasing faster means capturing revenue sooner and beating competitors to market.

Revenue impact of faster release × % acceleration

Example: A product that earns \$1M per month launches 2 weeks earlier. That's ~\$500K earlier revenue capture.

Demand Generation from Faster Launch Cadence

By removing production bottlenecks, AI allows you to run more campaigns, launches, or experiments in the same time period—directly increasing pipeline and sales opportunities.

Incremental campaigns × avg. campaign ROI

Example: AI asset versioning lets you run 3 extra campaigns/quarter, each yielding \$50K pipeline → \$150K/quarter.

“

It is more important for your first few AI projects to succeed rather than be the most valuable AI projects. They should be meaningful enough so that the initial successes will help your company gain familiarity with AI.

-Andrew Ng, Co-Founder
DeepLearning.ai

Calculating ROI (continued)

Cost Deferred or Avoided

AI can reduce or eliminate the need for external resources, as well as avoid the downstream costs of delays, rework, or inefficient processes.

(Cost of external resources or delays) – (Cost after AI implementation)

Example: AI-driven QA reduces reliance on an outsourced vendor by \$60K/year.

Risk Reduction

Shorter cycles mean you can test ideas earlier and kill underperforming ones before they consume significant budget. This lowers the financial impact of failure while enabling more experimentation.

(Cost of failed late-stage initiatives) – (Cost of failed early-stage initiatives with AI)

Example: \$150K – \$5K = \$145K preserved capital per avoided late-stage failure.

However you're measuring success—start small.

You don't want to be in a position where you've spent six months and six figures on an AI initiative only to realize the business impact is unclear or impossible to measure. That's how AI programs lose momentum—and budget. Early wins build credibility. Pick projects that are meaningful but manageable, where you can show progress in weeks, not quarters. When you can point to something concrete—"we cut campaign production time in half" or "we freed up 200 design hours last month"—you give leadership the proof they need to keep investing.

Identifying ROI: The Navigator™ exercise

In AI adoption, the hardest part isn't "can we build it?"—it's knowing what's worth building in the first place.

Navigator is our structured, 2-hour working session designed to surface the AI initiatives most likely to deliver measurable value fast. We help you cut through scattered ideas, vendor noise, and internal debates to arrive at a clear, defensible set of priorities backed by a business case you can take straight to leadership.

How it Works

1. Understand Your Priorities

We align on your goals, current tools, data landscape, and where your team feels the most friction. We work around constraints like data security or regulatory compliance from the start.

2. Map Your Workflows

Together, we trace the key steps in your processes—looking for places where AI can automate repetitive work, accelerate cycle time, or augment human decision-making.

3. Assess Proven Use Cases

We bring a vetted library of AI workflows across product, marketing, and operations. You'll see what's working in similar organizations and how it applies to your environment.

4. Prioritize for ROI

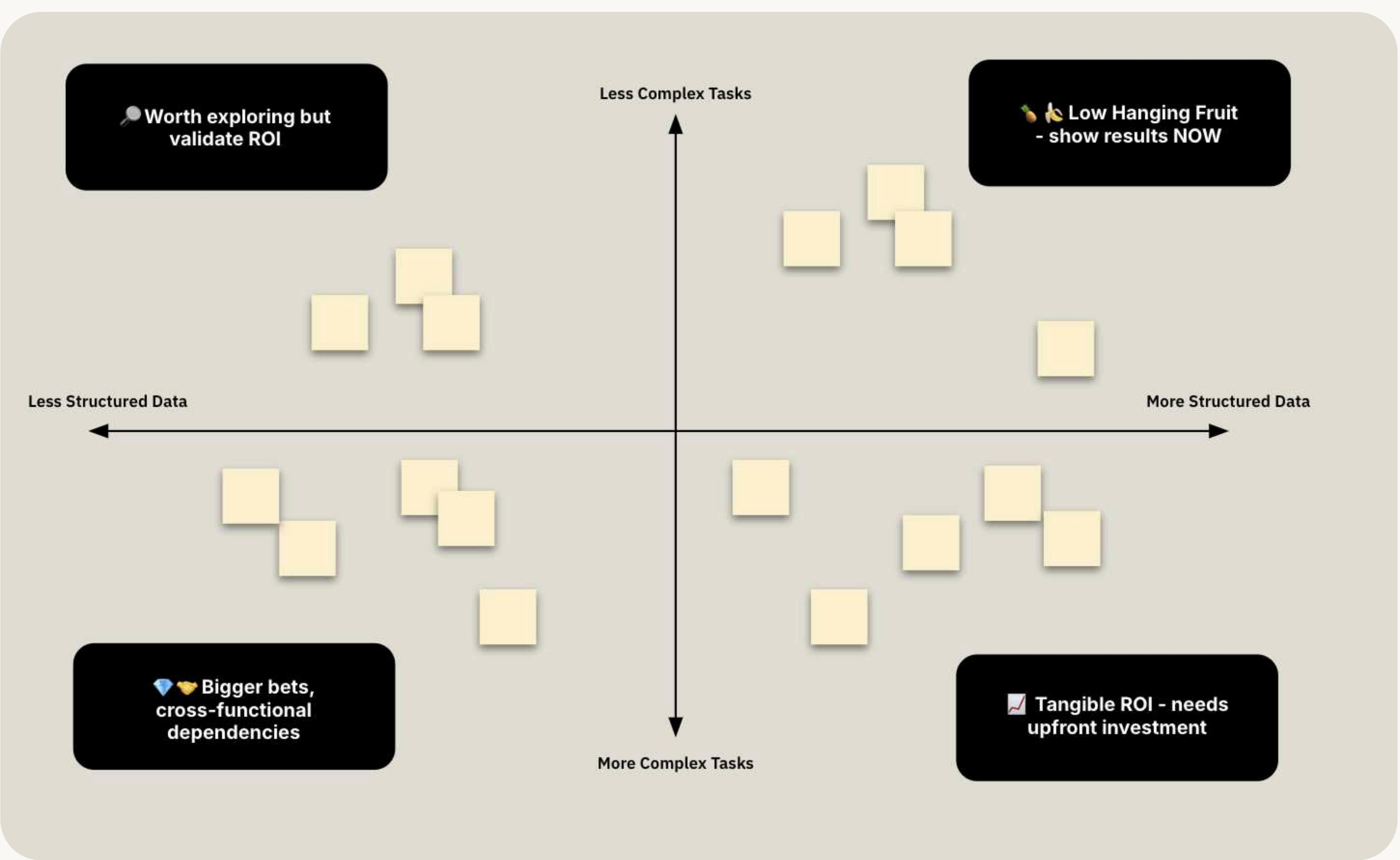
We facilitate a 2×2 exercise to quickly distinguish:

- Quick wins – low effort, high return
- Bigger bets – longer horizon, significant upside
- Low-value distractions – avoid the hype traps

5. Deliver Your Action Brief

You walk away with:

- A metric-driven business rationale for your top AI opportunities
- A clear implementation path (phased or immediate)
- A visual opportunity map you can share with your leadership team



Learn more here:

summerfriday.co/navigator

Final Thoughts

AI has already changed the way product teams work. The difference between teams that get incremental gains and those that fundamentally outperform isn't the tools they choose—it's how intentionally they design their systems, workflows, and culture around AI.

Most teams have already experimented—asking ChatGPT a quick question, spinning up a CustomGPT, or letting an AI assistant write first drafts. The real advantage comes when AI isn't a sidekick but a trusted part of how you design, build, and deliver—embedded in processes, informed by your context and history, and able to act with speed and precision.

The work ahead isn't about chasing every new tool. It's about shaping a way of working that can absorb new capabilities, adapt to new challenges, and keep improving without constant reinvention. That means building intelligent systems, investing in change management, and making AI fluency as fundamental as technical or craft fluency.

The companies that win in the AI era will be the ones that pair ambition with discipline—teams that move fast, learn faster, and never lose sight of the human judgment that keeps AI useful and trustworthy.

Are you ready for AI-First product design?

This checklist helps you assess your current capabilities and identify the gaps that matter most. Rather than trying to implement everything at once, use this as a diagnostic tool to prioritize your AI transformation efforts. Focus on the areas where you're missing critical foundations—these often become the bottlenecks that prevent AI from delivering real value.

AI-First Principles

- ✓ Treat prototypes as learning tools, not precious deliverables
- ✓ Focus team energy on strategic work while AI handles routine tasks
- ✓ Establish a culture of safe experimentation with clear success metrics
- ✓ Redesign workflows around AI capabilities, not just add AI features
- ✓ Create transparent guidelines and human oversight for all AI outputs

Context + Memory + Tools Infrastructure

- ✓ Build design systems with semantic structure and encoded design intent
- ✓ Maintain organized libraries of prompts, templates, and AI configurations
- ✓ Integrate tools to enable seamless human-AI collaboration
- ✓ Track AI performance data and user feedback for continuous improvement
- ✓ Establish reliable data sources that AI can access consistently

Team Capabilities & Roles

- ✓ Assign ownership of technical AI operations (prompt management, quality monitoring)
- ✓ Designate someone to manage AI governance and quality standards
- ✓ Identify who drives AI process strategy and workflow optimization
- ✓ Define AI collaboration patterns for PM, design, and engineering teams
- ✓ Create clear escalation paths when AI systems produce problematic outputs

Change Management & Adoption

- ✓ Communicate clear leadership vision for AI transformation
- ✓ Provide team training on AI tools and best practices
- ✓ Run pilot projects to test AI workflows before full adoption
- ✓ Measure AI impact on productivity, quality, and team satisfaction
- ✓ Develop processes for scaling successful AI implementations

Product Development Process

- ✓ Include AI-powered research synthesis with human validation in discovery
- ✓ Use AI to accelerate ideation while maintaining strategic direction
- ✓ Leverage AI for consistency and speed in design-to-development handoffs
- ✓ Establish quality gates for AI-generated content and code
- ✓ Account for AI-assisted features in testing and validation processes

Measurement & Optimization

- ✓ Track cycle time improvements from AI adoption
- ✓ Measure quality metrics for AI-generated outputs
- ✓ Monitor team satisfaction and adoption rates for AI tools
- ✓ Define clear ROI metrics that justify AI investments to leadership
- ✓ Regularly assess and optimize AI workflows based on performance data

Take the next step.

Summer Friday helps product teams move from scattered experiments to systematic, measurable AI advantage.

With the right mix of AI strategy, design system expertise, and operational know-how, we'll help you:

- Identify your highest-ROI AI opportunities
- Redesign workflows around AI capabilities
- Build the infrastructure that unlocks compounding returns
- Train your team to work confidently alongside AI

Some of the companies we've helped



EmblemHealth



Some of the ways we can work with you

Navigator™ Workshop

Pinpoint quick wins and bigger bets in a single working session

AI-Ready Design Systems

Encode intent, not just appearance, so AI can design with you

Workflow Redesign

Integrate AI where it accelerates your team the most

Training Programs

Give your team the skills to make AI a natural part of their toolkit

Let's build your AI advantage

summerfriday.co