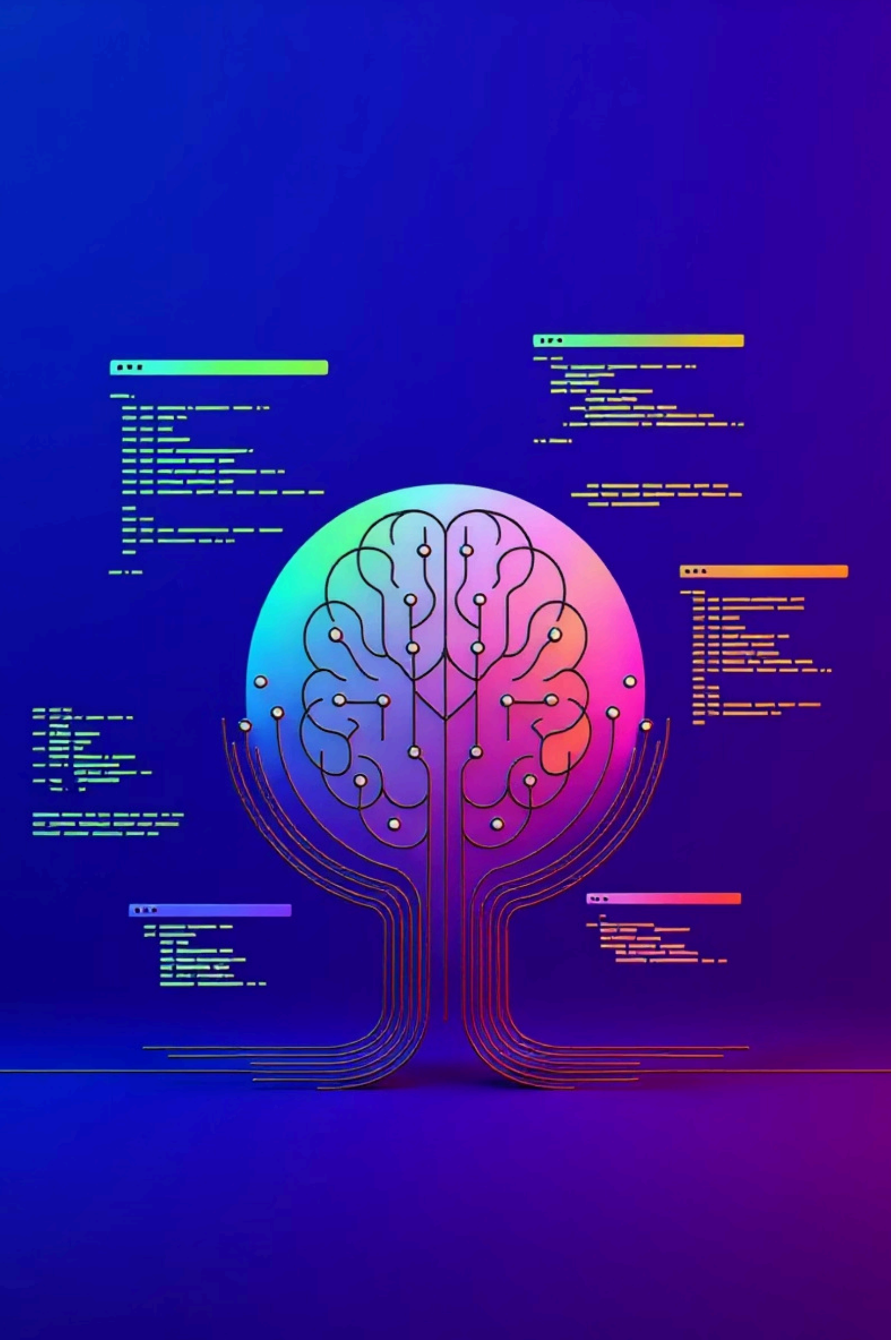


## Unlock AI's True Potential.

# Stop Fighting Your AI:

## Engineering Prompts That Actually Work

You've likely experienced the inconsistent magic of AI prompts. Today, we move beyond basic trial-and-error to a strategic, production-ready approach. Discover how clear structure, deep context, and understanding model-specific patterns transform your AI interactions from "kind of works" to "consistently delivers." Get ready to elevate your AI game.

### Find Slides @ www.martinrojas.dev

by Martin Rojas

# Agenda

## Prompt Architecture

The anatomy of production prompts

## Core Engineering Techniques

Clarity, Chain-of-Thought, Constraints, Compression

## AI-as-Coach

Using AI to improve your own prompts

## Advance Patterns

Layering techniques for reliable results

# Part 1: Prompt Architecture

## The Anatomy of Production Prompts

### Component Stack

| Component | Purpose | Business Example |
|---|---|---|
| System Message | Sets behavior and role | "You are a customer success analyst" |
| Instruction | What to do | "Identify customers at risk of churning" |
| Context | Background data | "Usage metrics, support tickets, ARR data" |
| Examples | Pattern demonstration | "Sample output: 'Found 23 high-risk accounts…'" |
| Constraints | Output limits | "Return risk score first, then ARR at risk" |
| Delimiters | Section separation | ###, ---, """ |

Think of prompts as modular—each component serves a distinct purpose. Like a well-structured business report, clarity in structure leads to clarity in output.

# Markdown: Structuring Prompts for Precision

Markdown syntax provides a clear, standardized way to format your AI prompts, ensuring models accurately interpret structure, hierarchy, and emphasis.

| | | |
|---|---|---|
| # Heading | Defines sections, outlines hierarchy | # Instructions: Create a plan |
| **Bold Text** | Emphasizes keywords or concepts | **Focus:** User benefits |
| - List Item | Presents clear, actionable points | - Step 1: Draft outline |
| `Code` | Isolates specific terms or variables | Filter where sport = 'basketball'`` |
| ```Block``` | Encapsulates larger blocks of text/code | ``` Output as JSON: { } ``` |

# Our Running Example

## Base Prompt: Customer Churn Risk Analysis

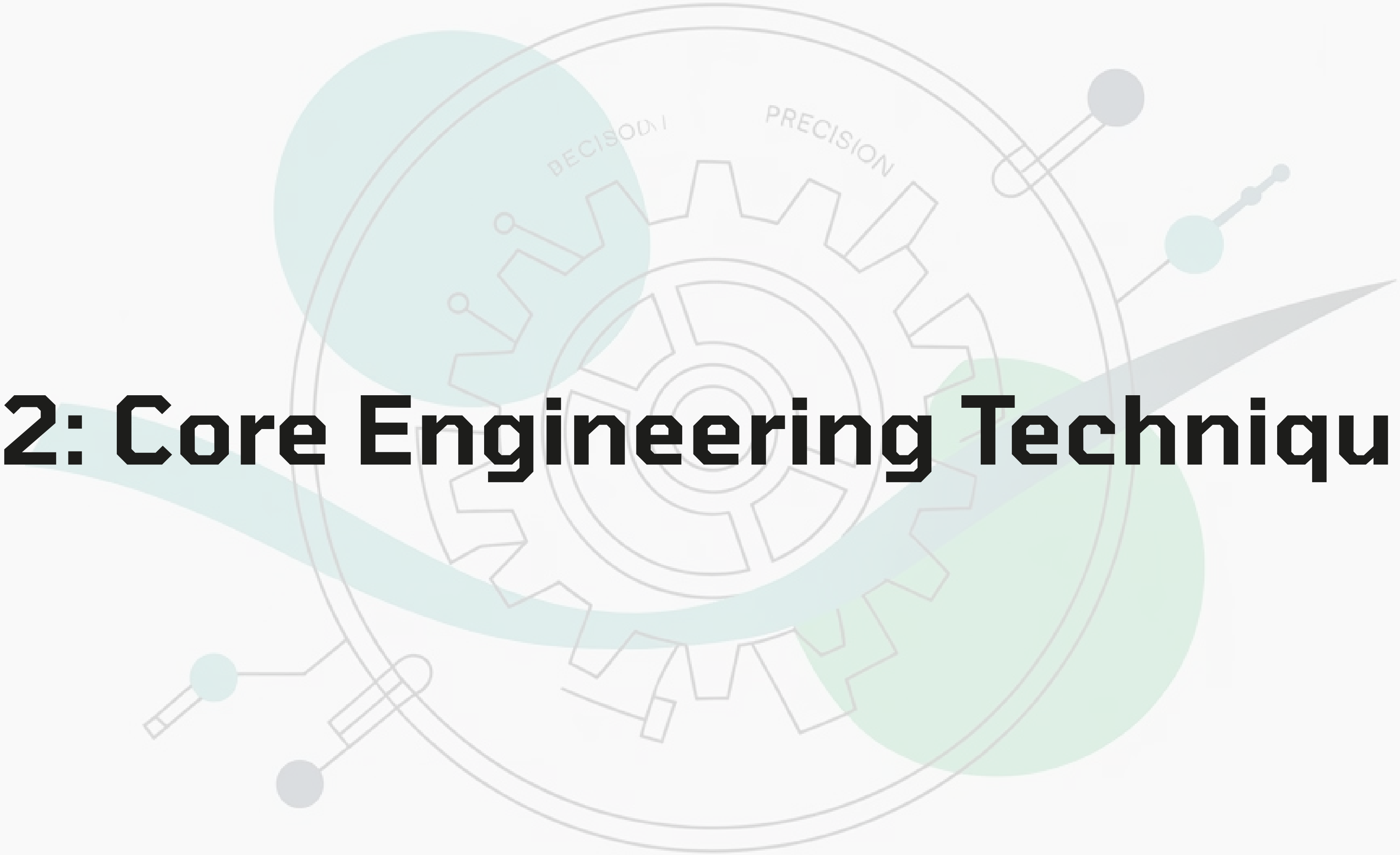
**This is our starting point—already decent but not production-ready. We'll transform this throughout the presentation.**

```
Which enterprise customers are at risk of churning?
How much revenue could we lose?
```

**What's missing?**

- No defined risk criteria or thresholds
- Ambiguous what "at risk" means (usage decline? support tickets? payment issues?)
- No instruction on time horizon or comparison periods
- Two questions bundled without clear separation

# Part 2: Core Engineering Techniques



# Foundational Techniques in Action

## The Basic Building Blocks

### Zero-shot - Direct Instruction

Which enterprise customers are at risk of churning?

✅ Simple, fast | ❌ Inconsistent quality, ambiguous

### One-shot - Format Setting

Example: "Analysis identified 23 high-risk enterprise accounts representing $4.2M ARR, with average usage decline of 47% over 90 days."

Now answer: Which enterprise customers are at risk of churning? How much revenue could we lose?

✅ Consistent format | ❌ Limited pattern learning

### Few-shot - Pattern Learning

Example 1 (High Risk): "Acme Corp - ARR: $180K, Usage: -62%, Support tickets: 12 in 30 days, Last login: 3 weeks ago → HIGH RISK"

Example 2 (Medium Risk): "TechStart Inc - ARR: $95K, Usage: -28%, Support tickets: 3 in 30 days, Last login: 5 days ago → MEDIUM RISK"

Based on this pattern, evaluate our enterprise customer base for churn risk and categorize accordingly.

✅ Adapts to context | ❌ Token intensive

### Role-based - Behavioral Framing

You are a customer success analyst preparing a churn risk report for the leadership team. Identify enterprise customers showing signs of potential churn and quantify the revenue at risk.

✅ Consistent voice | ❌ May override other instructions

# Technique 1 - Clarity & Specificity

## The Ambiguity Tax

### ❌ Vague Churn Query:

```
Which enterprise customers are at risk of churning?
```

**Problems:** What defines "at risk"? Which signals matter? What time horizon?

### ✅ Refined with Specificity:

```
You are a customer success analyst. Answer the following
question about our enterprise customer base.

QUESTION: Which enterprise customers (ARR > $50K) show high
churn risk based on the last 90 days of activity?

RISK CRITERIA:
1. Usage decline > 30% compared to prior 90-day period
2. Support ticket volume > 5 in last 30 days
3. No login activity in past 14 days
4. Meeting any 2+ criteria = HIGH RISK

OUTPUT FORMAT:
- First line: Count of high-risk accounts and total ARR at risk
- Second line: List of top 5 accounts by ARR with risk indicators
- Third line: Brief explanation of methodology

Audience: Customer Success leadership team
Tone: Professional, actionable
```

**Model-Specific Notes:**

- GPT-4o: Responds well to numeric constraints
- Claude: Needs explicit boundaries or tends to over-explain
- Gemini: Best with hierarchical structure (### headings)

# Technique 2 - Chain-of-Thought (CoT)

## Make the Model Think Like an Analyst

### ❌ Direct Approach:

Which enterprise customers are at risk of churning? How much revenue could we lose?

### ✅ With Chain-of-Thought:

Analyze the customer churn risk step-by-step:

1. First, identify all enterprise customers (ARR > $50K) in the dataset
2. Then, calculate usage change % for each (current 90 days vs prior 90 days)
3. Next, pull support ticket counts for the last 30 days
4. Check last login date for each customer
5. Apply risk scoring: 2+ risk indicators = HIGH RISK
6. Sum the ARR for all high-risk accounts
7. Present findings with count and ARR at risk

Think through each step before providing the final answer.

**Why CoT matters for business analysis:** Prevents the model from jumping to conclusions on complex queries. You can trace the logic and catch errors in reasoning.

# Technique 3 - Format Constraints

## Structure = Reliability

### ❌ Unstructured:

```
Tell me about customers that might churn.
```

### ✅ With Format Constraints:

```
Answer the following churn risk question with EXACTLY this structure:

SUMMARY: [One sentence with count of high-risk accounts and total ARR at risk]

DETAIL:
- Enterprise Customers Analyzed: [total count in dataset]
- Customers with Usage Decline >30%: [count]
- High-Risk Accounts (2+ indicators): [count]
- Total ARR at Risk: [dollar amount]

METHODOLOGY: [2-3 sentences explaining the risk scoring logic]

Return ONLY the formatted response. No additional commentary.
```

### Structured JSON Output:

```
Return ONLY valid JSON matching this schema:

{
  "analysis_date": "YYYY-MM-DD",
  "period_analyzed": "last_90_days",
  "summary": {
    "total_enterprise_customers": integer,
    "high_risk_count": integer,
    "total_arr_at_risk": decimal
  },
  "high_risk_accounts": [
    {
      "customer_id": "string",
      "customer_name": "string",
      "arr": decimal,
      "usage_change_pct": decimal,
      "support_tickets_30d": integer,
      "days_since_login": integer,
      "risk_score": "HIGH"
    }
  ],
  "methodology": "string describing risk criteria"
}

NO additional text. Only JSON.
```

Success Rate: 92% valid JSON (vs 45% with natural language requests)

# Technique 4: Prompt Compression

## Every Token Counts

### ❌ Verbose (147 tokens):

You are an expert customer success analyst with years of experience in churn prediction and retention analysis. Your task today is to carefully analyze the provided customer data that contains all the usage metrics, support ticket information, and revenue data, and then create a comprehensive breakdown that would be suitable for presentation to the leadership team. Please make sure to include information about which enterprise customers are showing signs of potential churn, the total ARR at risk, and what indicators suggest they might leave.

### ✅ Compressed (42 tokens):

Customer success analyst. Analyze enterprise customer data.
Find: High churn risk accounts (usage decline >30%, high support volume).
Output: Count, ARR at risk, top 5 accounts with risk indicators.
Format: Professional summary.

### Ultra-Compressed (26 tokens):

Task: Churn risk analysis
Query: Enterprise customers with declining usage + high support tickets
Output: Count, ARR at risk, top 5

**Savings:** 71-82% fewer tokens | Same output quality | Lower costs

> ### Challenge:
>
> Take your longest prompt, cut 40% of tokens. Drop filler words: 'please,' 'could you,' 'make sure.' Use headers and lists instead of sentences.

# Building Layers

## Progressive Enhancement

### Layer 0: Naked Prompt

```
Which customers are at risk of churning?
```

❌ Vague, inconsistent, may hallucinate

### Layer 1: +Role

```
You are a customer success analyst.

Which enterprise customers are at risk of churning?
```

✓ Consistent analytical tone

### Layer 2: +Specificity

```
You are a customer success analyst preparing a report for leadership.

Analyze customer data to find:
- Enterprise accounts (ARR > $50K) with usage decline > 30%
- Where support ticket volume > 5 in last 30 days
- Total ARR for accounts meeting both criteria

Use the last 90 days as the analysis period.
```

✓ Clear Parameters

### Layer 3: +Chain-of-Thought

```
[Previous layers...]

Before providing your answer, work through these steps:
1. Identify total enterprise customers in dataset
2. Calculate usage change % for each (current vs prior 90 days)
3. Filter for usage decline > 30%
4. Cross-reference with support ticket data
5. Count and sum ARR

Show your reasoning, then present the final answer.
```

✓ Traceable logic

### Layer 4: +Constraints

```
[Previous layers...]

OUTPUT FORMAT:
SUMMARY: [count] high-risk accounts | $[amount] ARR at risk

BREAKDOWN:
- Enterprise customers analyzed: [number]
- Customers with usage decline >30%: [number]
- High-risk (usage + support signals): [number]
- Total ARR at risk: $[amount]

TOP 5 AT-RISK ACCOUNTS:
[List with customer name, ARR, and risk indicators]
```

✓ Predictable structure

### Layer 5: +Validation (Production-Ready)

```
SYSTEM: You ONLY report figures from the provided data.
Never estimate or approximate. If data is missing, state "Data not available."

[All previous layers...]

VALIDATION: Before responding, confirm:
- All figures match source data exactly
- ARR totals are mathematically correct
- No customers are double-counted
```

✓ Production-ready

# Part 3:
# AI-as-Coach

# The AI-as-Coach Method

Instead of guessing what makes a good prompt, let AI help you improve your prompts. This creates a feedback loop that rapidly improves your results.

## Quick-Start Template

```
Act as a Senior Prompt Engineer. Your goal is to help me refine and optimize a prompt for maximum performance. When I provide a prompt, please follow these steps:

1. Critique: Review the prompt for structure, coherence, and clarity. Identify any ambiguous phrasing or "fluff" that might confuse an LLM.
2. Intent & Specificity: Evaluate if the prompt provides enough context and clear constraints. Identify what might be missing to achieve a high-quality result.
3. Clarifying Questions: Before providing the final version, ask me 2-3 targeted questions to bridge any gaps in context, tone, or intended output format.
4. The Optimization: Once I answer, provide the "v2.0" version of the prompt using best practices (e.g., Role-assigning, Delimiters, and Chain-of-Thought prompting).

Do you understand? If so, please ask me for the prompt you'd like to optimize.

[your current prompt]
```

## Why This Works:

### Sequential Logic

Instead of just guessing what you want, it forces the AI to stop and think (and ask you questions) before it writes the final version.

### Structural Integrity

By asking for "Delimiters" and "Role-assigning," you're prompting the AI to use professional engineering techniques in the output.

### A quick tip for the road

If you find the AI is still being too "wordy" in its critique, you can add a line to the end: *"Keep the critique concise and focused on actionable improvements."*

# The Lean Master Prompt

We can condense multiple optimization goals into a "Lean Master Prompt" that retains the powerful recursive logic of reviewing, questioning, and refining. This version trims unnecessary verbose phrasing without sacrificing effectiveness.

> Act as a Senior Prompt Engineer. Review my next prompt for **clarity, structure, and intent**. Identify any ambiguities or missing context, then **ask 2-3 targeted questions** to bridge those gaps. After I respond, provide an optimized "v2.0" using professional prompting standards (Roles, Delimiters, and Constraints). Ready?

## Why this keeps the logic:

### Bundled Review

"Clarity, structure, and intent" efficiently covers comprehensive critique, replacing multiple distinct review steps.

### The "Stop" Command

It still forces the AI to ask questions *before* generating the final version, preventing premature "best guess" outputs.

### High Standards

Referencing "professional prompting standards" succinctly instructs the AI to apply advanced techniques like role-assigning and Chain-of-Thought.

# Part 4: The AI Mindset

## Key Takeaways:

**Structure beats sophistication** — Clear formatting produces better results than clever wording

**Test techniques against YOUR use cases** — What works for one query may not work for another

**Measure what matters** — Accuracy, consistency, and speed

**Build prompt templates** — Reuse what works for common finance queries

**Use AI to improve AI** — The AI-as-Coach method creates a rapid feedback loop for prompt refinement

## Immediate Action:

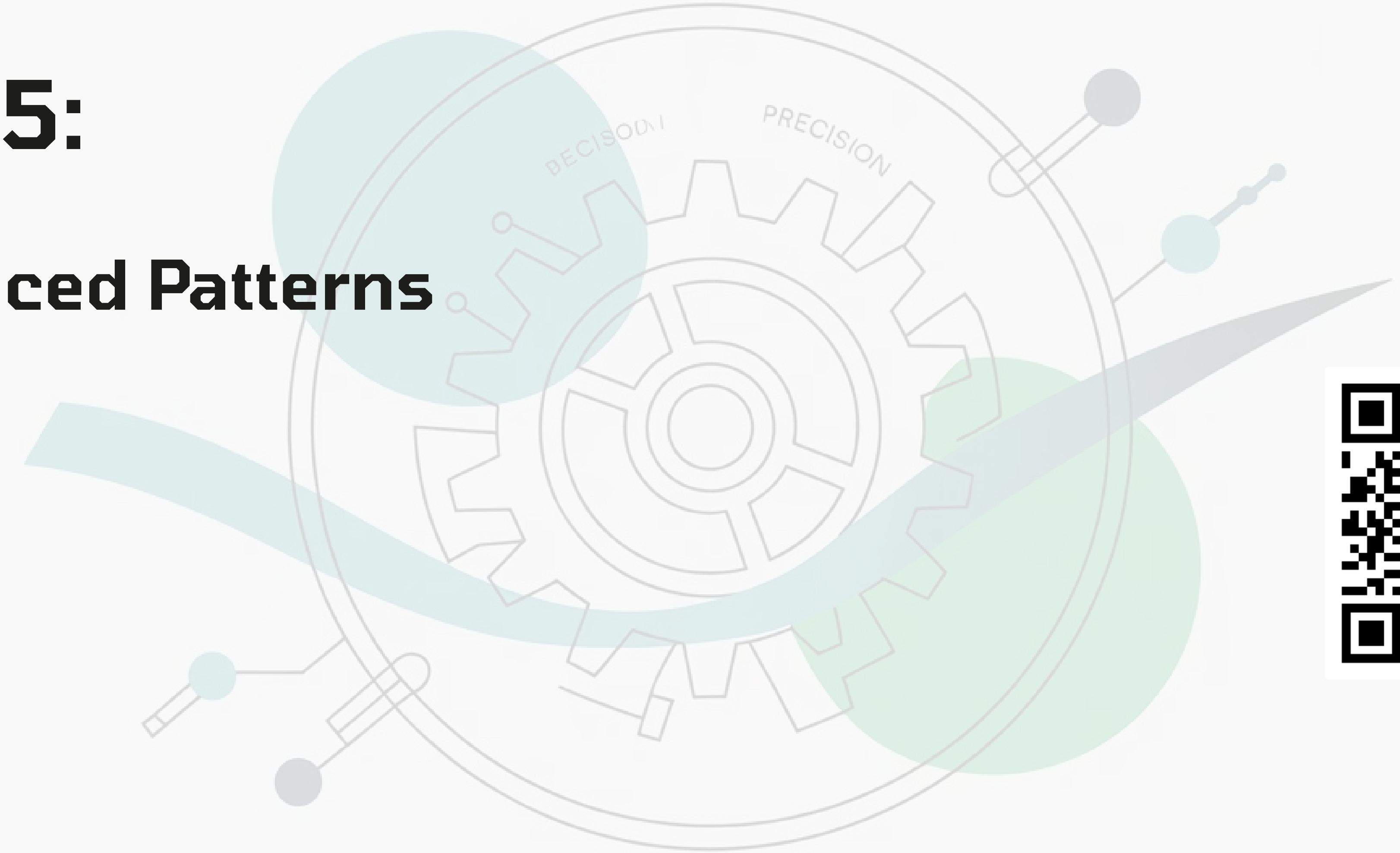
"Pick one advanced technique from today"

"Apply it to your most problematic prompt"

"Share results back to the community"

# Part 5:

## Advanced Patterns



# Tree of Thought (ToT)

## Explore multiple analytical approaches *simultaneously*, then evaluate which reveals the most insight.

```
We've identified 23 high-risk enterprise accounts. Before planning
interventions, we need to understand WHY they're at risk.

Explore this problem through THREE different analytical lenses:

LENS A — Product Engagement:
- Analyze feature adoption patterns for at-risk vs healthy accounts
- Which features are they NOT using?
- Did engagement drop suddenly or gradually?
- Is there a specific feature gap driving dissatisfaction?

LENS B — Support Experience:
- Analyze support ticket content and resolution times
- What are the recurring complaint themes?
- How does their support experience compare to healthy accounts?
- Are there unresolved issues creating frustration?

LENS C — Business Fit:
- Analyze company characteristics (size, industry, use case)
- Are at-risk accounts concentrated in specific segments?
- Did their business needs change (growth, contraction, pivot)?
- Are they potentially mismatched to our product?

After completing all three analyses:
1. Compare the explanatory power of each lens
2. Identify which approach reveals ACTIONABLE root causes
3. Recommend which lens should drive our retention strategy

Present findings as:
- Executive summary (which lens won and why)
- Key insight from each lens
- Recommended interventions based on the strongest analysis
```

**When to use:** When you're not sure which analytical lens will be most valuable, or when a problem could be approached from multiple valid angles.

**Why this works:** Instead of guessing which approach matters most, you let the data reveal it. "Usage decline" might be a symptom—the root cause could be support failures or product-market fit issues.

# Self-Consistency

## Majority Vote for Accuracy

```
Calculate total ARR at risk from high-churn enterprise customers.
This figure will appear in our quarterly board report, so accuracy is critical.

Perform THREE independent calculations:

METHOD 1 — Direct Filter:
- Start with all enterprise customers (ARR > $50K)
- Filter: usage_change < -30%
- Filter: support_tickets_30d > 5 OR days_since_login > 14
- Sum ARR for qualifying accounts
- Document customer count at each step

METHOD 2 — Risk Score Approach:
- Assign risk points: usage decline (1pt), high tickets (1pt), no login (1pt)
- Filter: risk_score >= 2
- Sum ARR for high-risk accounts
- Document score distribution

METHOD 3 — Cohort Comparison:
- Identify accounts matching historical churn profile
- Cross-reference current at-risk indicators
- Sum ARR for accounts in both sets
- Document overlap statistics

VALIDATION:
- Compare all three results
- If all match: Report with HIGH CONFIDENCE
- If 2 of 3 match: Flag the outlier method, investigate why
- If all differ: STOP — do not report, escalate for data review

Output format:
| Method | Account Count | ARR at Risk | Notes |
|--------|---------------|-------------|-------|
| Method 1 | | | |
| Method 2 | | | |
| Method 3 | | | |

FINAL ANSWER: $[amount] (Confidence: HIGH/MEDIUM/LOW)
```

**What it is:** Solve the same problem multiple independent ways, then validate that answers converge. Flag discrepancies for review.

**When to use:** High-stakes calculations where accuracy is critical—board presentations, investor reports, retention forecasts.

**Why this works:** Catches errors that single-path reasoning misses. If methods disagree, you've found a data quality issue or logic error *before* it reaches stakeholders.

# ReAct Pattern

## Iterative Reason → Act → Observe loops for multi-step investigations.

**When to use:** Exploratory analysis where each finding shapes the next question, or when solving complex, open-ended problems requiring a structured approach.

**Business Example:**

```
Investigate the concentration risk in our at-risk customer base.

Use this iterative pattern:

THINK: What's my first question?
→ "How concentrated is the ARR at risk among our high-risk accounts?"

ACT: Calculate the top 5 accounts' share of total ARR at risk.

OBSERVE: [Results]

THINK: What does this tell me? What should I investigate next?
→ [Your reasoning based on the observation]

ACT: [Next analytical step — perhaps: Are these accounts in the same
industry? Same customer success rep? Same onboarding cohort?]

OBSERVE: [Results]

THINK: Is there a pattern emerging?
→ [Your reasoning]

ACT: [Drill deeper into the pattern]

OBSERVE: [Results]

Continue until you can answer: "Is our ARR at risk concentrated in
a way that suggests a systemic issue, and if so, what's the root cause?"

Final output: Risk assessment with supporting evidence from each
Reason-Act-Observe cycle, plus recommended interventions.
```

**Why this works:** This pattern mimics human problem-solving, making the AI's investigative process transparent and auditable. By grounding each 'ACT' in prior 'OBSERVATIONS', it significantly reduces hallucination and allows the AI to dynamically adapt its inquiry based on intermediate findings, leading to more robust and accurate conclusions for complex, ambiguous problems.