SESSION 2: ADVANCED PATTERNS & WORKFLOWS

Orchestrating Multiple Prompt Patterns for Complex Tasks

90 minutes (15 min presentation + 60 min hands-on + 15 min review)

From Individual Patterns to Orchestrated Workflows

- **Session 1 Recap:**
  - Foundational patterns: Few-shot, Chain-of-Thought, Persona, Template
  - Configuration standards: .cursorrules, Copilot instructions
  - Decision documentation: ADRs vs structured prompts

- **Today's Goal:**
  - Learn advanced patterns: ReAct, Tree of Thoughts, Meta-prompting

Advanced Prompt Patterns Overview

- **ReAct Pattern (Reason + Act) - Yao et al. 2022**
  - *Think → Act → Observe → Think → Act → Observe…*
  - For multi-step tasks requiring validation at each stage

- **Tree of Thoughts Pattern - Yao et al. 2023**
  - *Generate multiple solution branches → Evaluate → Choose best*
  - For decisions with tradeoffs, maps to ADR 'Alternatives Considered'

Workflow Orchestration Strategies

- Strategy A: ADR-Driven (Industry Standard)
  - Write ADR → Reference in .cursorrules → Use AI → Validate
  - Proven, lightweight, widely understood

- Strategy B: Structured File Workflow (Today's Example)
  - System prompt → Task spec → Execution plan → Decisions → Generate
  - Explicit pattern application, good for learning

ReAct Pattern: Iterative Validated Steps

- Without ReAct: "I'll migrate everything at once"
  - Something breaks, unclear which change caused it

- **With ReAct: Iterative, Validated Steps**
  - THINK: What dependencies must be satisfied first?
  - ACT: Replace javax.* with jakarta.* imports
  - OBSERVE: mvn compile → SUCCESS ✓
  - THINK: Now annotations can be safely updated…

Tree of Thoughts: Exploring Alternatives

- When to Use Tree of Thoughts:
  - Multiple valid approaches exist
  - Tradeoffs matter for your specific context
  - Team needs to understand decision rationale

- **Example: Exception Handling Strategy**
  - Branch A: Keep current (minimal changes, low risk) ✓
  - Branch B: Adopt ProblemDetail (modern, more testing)