

# Client Project Discovery

From Vague Requirements to Working Prototype

Dhara Mungra

University of Mannheim; SimPPL

---

# The Challenge: Loose Requirements

Clients often come with ideas, not specifications.

- "Build me an AI tool for my business"
- "I need automation for customer support"
- "Can you create a data analysis agent?"

# Discovery Framework (5 Stages)

**1** **Internal Discussion**  
Align team understanding

**2** **Client Clarification**  
Ask structured questions

**3** **Market Research**  
Existing solutions & APIs

**4** **User Interviews**  
Validate assumptions

**5** **Prototyping**  
Build, test, iterate

# Stage 1: Internal Discussion

- What assumptions are we making about this request?
- What technical constraints do we foresee?
- What questions **MUST** we ask the client?
- Who should lead which aspect of discovery?

## Stage 2: Essential Client Questions (Part 1)

- **Purpose:** What problem does this solve?
- **Users:** Who will use this? Daily frequency?
- **Context:** How will they access and use it?
- **Success:** How do you define project success?

## Stage 2: Essential Client Questions (Part 2)

- **Existing solutions:** Why not use Tool X or Y?
- **Must-have vs nice-to-have:** What's essential?
- **Constraints:** Budget, timeline, integrations?
- **Data access:** What systems must we connect?

# Why Ask About Existing Tools?

This question reveals:

- Their specific unmet needs
- Features they actually care about
- Integration or workflow requirements

## Stage 3: Market Research

- Identify 3-5 similar tools in the market
- Analyze their feature sets and pricing
- Research available APIs for quick integration
- Document technical approaches and patterns

# Stage 4: User Interviews

Goal: Understand real user needs without bias

- Talk to 3-5 representative end users
- Focus on their current workflows and pain points
- Avoid leading them to predetermined answers

# Crafting Unbiased Questions

- **Use neutral language:** Avoid positive/negative words
- **Ask open-ended:** Encourage detailed responses
- **Avoid assumptions:** Don't presume problems
- **One topic per question:** No double-barreled questions

# Leading & Biased Questions

- "How *simple* was this feature?" (assumes simplicity)
- "Why was the navigation *confusing*?" (presumes problem)
- "Would you *love* an AI assistant here?" (suggests answer)
- "Don't you think automation would help?" (leading)

# Neutral and Open-Ended Questions

- "Walk me through how you completed this task last week"
- "What was your experience using the navigation?"
- "Describe the most frustrating part of your workflow"
- "How do you currently handle [specific task]?"

# Understanding Current Workflow

Questions to ask:

- "Show me how you do [task] step-by-step"
- "What tools do you use for this? Why those?"
- "When did you last encounter this problem?"

**What you're seeking:** Actual behavior, tool choices, frequency, context

# Uncovering Pain Points

Questions to ask:

- "What takes the most time in your daily work?"
- "Describe a recent situation where you felt stuck"
- "If you could eliminate one task, what would it be?"

**What you're seeking:** Time sinks, blockers, manual repetitive tasks

# Understanding Desired Outcomes

Questions to ask:

- "What would a successful solution look like to you?"
- "How would you know this tool is working well?"
- "What metrics matter most to your team?"

**What you're seeking:** Success criteria, measurable goals, priorities

# Interview Best Practices

- Focus on past behavior, not hypotheticals
- Ask "why" to understand motivations
- Listen more than you talk (80/20 rule)
- Record sessions (with permission) for analysis

## Stage 5: Prototyping

- Start with low-fidelity UI mockups (paper/Figma)
- Build minimal working prototype with core features
- Test with 2-3 users and gather feedback
- Iterate based on real usage patterns

# Prototyping Tips for AI Agents

- Use free LLM APIs (Groq, Hugging Face) for demos
- Mock complex integrations initially
- Focus on 1-2 core use cases first
- Document what works and what doesn't

# Common Pitfalls to Avoid

- Building before understanding the problem
- Assuming client knows what they need
- Skipping user validation
- Over-engineering the first version

# Discovery Phase Deliverables

- Requirements document with user stories
- Competitive analysis summary
- User interview insights report
- Working prototype or UI mockups



# Ask Me Anything!

Questions?  
Please reach out to  
[dhara.atul.mungra@uni-mannheim.de](mailto:dhara.atul.mungra@uni-mannheim.de)

Thank you!