Reflection on LLM Use for Requirements Engineering

Pain Points in Using LLMs

- Inconsistency in answers: similar prompts produced different conclusions, making it difficult to treat the output as a single source of truth.
- Hallucinations: LLMs sometimes introduced requirements we never asked for, confusing scope.
- Verbosity: without guidance, outputs were long and buried important details.
- Prompt ambiguity often led to vague or off-target results.

Surprises

- LLM could rationalize trade-offs effectively, especially in justifying exclusions (e.g., complexity vs. core value).
- However, identical prompts sometimes yielded conflicting results, showing the need for structured pre- and post-processing.

What Worked Best

- Using the LLM for requirements amplification—expanding a single requirement into detailed use cases and stakeholder impacts.
- Brainstorming exclusions: producing strong justifications for why to defer features.
- Generating stakeholder-specific perspectives when explicitly requested.

What Worked Worst

- Asking the model to 'summarize everything' without structure, resulting in overly long, unfocused output.
- Difficulty handling numbers, constraints, and technical feasibility accurately.
- Occasional self-contradictions across re-asked prompts.

Pre- and Post-Processing That Helped

- Pre-processing: scaffolding prompts (e.g., structured bullets by stakeholder) and clarifying MVP boundaries.
- Post-processing: condensing raw output into concise bullets/tables and filtering hallucinations.

Prompting Strategies (Best & Worst)

- Best: role assignment (e.g., 'act as a product manager'), stepwise prompting (breaking into sub-tasks), and structured formats (bullets, numbered lists).
- Worst: open-ended brainstorming without constraints, and re-asking prompts without structure (causing contradictions).

If Building a Startup: Tools Around LLMs for RE

- Consistency Checker: compares multiple runs of the same prompt to highlight contradictions.
- Requirement Structurer: formats LLM I/O into standard RE artifacts (use-cases, user stories, acceptance criteria).
- Amplify/Condense Toggle: expands one requirement into details or condenses text into summaries.
- Stakeholder Lens Tool: reframes requirements for each stakeholder (customers, business, regulators).
- Traceability Mapper: links requirements → use cases → acceptance tests via LLM subroutine.