Revised Reflection Document:
A Study on the Practical Application of LLMs

Aniruddh Sanjeev Bhagwat   Hardik Majhetia   Pranshav Gajjar   Ishan Patel

What are the pain points in using LLMs?

A main pain point we encountered was the tendency for LLMs to produce generic or incomplete outputs, especially with simple, zero-shot prompts. It takes significant effort to guide the model toward a detailed and actionable engineering document. Another challenge, which became evident during our model comparisons in `proj1b1`, is the inherent difference between models; what works for one does not guarantee success with another. For instance, while one model excelled at structured output, it sometimes overlooked critical details, such as alternative and error flows in use cases, rendering the output less useful from an engineering perspective.

Any surprises? E.g., different conclusions from LLMs?

The most significant surprise was the stark difference in output between the LLMs when we used the same iterative prompting strategy. One model, **Gemini 2.5 Pro**, demonstrated a capacity for strategic exploration, diverging from the direct task to provide a high-level analysis of the problem space and even introducing external concepts like “Choice Architecture.” Meanwhile, **DeepSeek**, was exceptional at structured refinement, adhering closely to instructions to produce a meticulously detailed requirements document. This showed that some LLMs can act as strategic partners, while others are better suited as high-powered documentation assistants.

What worked best?

The iterative, conversational prompting strategy was unequivocally the most effective method. This became clear early on in `proj1a1`, and was validated throughout `proj1b1` and `proj1c1`. Here, we learned that a careful, iterative approach consistently yields more sophisticated and nuanced results than a single prompt. For example, through this process, the LLM was able to expand upon our initial stakeholder list to identify not just broad categories like “Customers,” but also specific subgroups like “Families” and “Business Professionals,” and even non-obvious secondary stakeholders such as “Environmental Groups.” This experience showed us that LLMs can be powerful tools for discovery, not just documentation. The workflow of using a model like **Gemini** for initial exploration and then a model like **DeepSeek** for structured documentation proved to be a powerful approach.

What worked worst?

The “zero-shot” prompting approach worked the worst for this complex task. While fast, it produced superficial and generic documents that lacked the critical analysis necessary for a real-world project. Relying on a single, complex prompt and hoping for a perfect output was the least effective strategy.

What pre-post processing was useful for structuring the prompts and summarizing the output?

For pre-processing, breaking down the complex task of requirements analysis into a sequence of smaller, conversational prompts was key. This allowed for a guided exploration of stakeholders, conflicts, and finally, use cases. For post-processing, it was useful to manually synthesize and structure the outputs, especially from the more "creative" LLMs. For example, the strategic insights from **Gemini** had to be organized into a coherent analysis, as the model did not produce a final, polished document on its own.

Did you find any best/worst prompting strategies?

Yes, the best prompting strategy we found was a hybrid approach. We would begin with an open-ended, exploratory conversation with a model like **Gemini** to define the problem space and uncover hidden requirements. Then, we would take those refined insights and use a structured, goal-oriented conversation with a model like **DeepSeek** to generate the formal, detailed use cases and requirements documentation. The worst strategy was relying on a single, complex prompt and hoping for a perfect output.