CSC 510 Project 1D1 - Fall 2025

Group 28 - Dhananjay Raghu, Ishaan Patel, Shuba Kalyanasundaram

1. What are the pain points in using LLMs?

- The first difficulty was that unconstrained or lightly constrained prompts often produced content that was only partially relevant. The model tended to generalize beyond the immediate scope, which created additional editing work and slowed progress. When the prompt did not specify structure, definitions, and boundaries, the model filled gaps with confident but tangential material.
- The second difficulty was inconsistency across different models when given the same instructions. Some models respected requested structure and formatting, while others ignored headings or output schemas and offered free-form text. This variability undermined repeatability and increased the number of prompt iterations required to obtain usable material.
- The third difficulty arose from the realities of stakeholder trade-offs. Requirements that serve different stakeholder goals such as operational speed for staff versus flexibility for customers cannot be resolved by the model without explicit guidance. When the prompt did not express a clear prioritization, the model hedged or blended incompatible aims, which made the resulting artifacts harder to adopt.

2. Any surprises? Eg different conclusions from LLMs?

- The most striking surprise was how small changes in prompt staging altered results. When the order of context, examples, and constraints was varied, the same model produced meaningfully different drafts. This demonstrated that the sequence of steps discovery, outlining, drafting, and formatting, has a direct effect on outcome quality, independent of the specific wording.
- Second, the strategy used to stage the work had as much impact as the wording of any single prompt. Introducing an explicit discovery step, such as requesting a list of salient topics or risks before drafting requirements, improved relevance in subsequent generations. In effect, a small amount of guided structure early in the process led to more focused and useful outputs later.

3. What worked best?

- The most effective practice was careful prompting with explicit examples and a strict output schema. When we specified the required sections, the acceptance criteria style, and the exact headings to use, the model produced content that was immediately actionable.
- A close second was deliberate scoping. When we defined a narrow minimum viable product and stated, in plain language, which features were in scope and which were out of scope, the model stayed within those guardrails. This reduced rework and conflict between generated material and the intended release.

4. What worked worst?

- Prompts that attempted to cover too many features at once produced long, internally inconsistent drafts. This created editing overhead and reintroduced items that had already been intentionally deferred. In addition, when the model was asked to follow a particular structure but the prompt did not enforce that structure tightly, the output frequently drifted, and we lost time repairing formatting rather than refining substance.
- Another consistently poor outcome arose when the prompts did not fix a shared vocabulary before generation. Different runs of the model produced different names for the same concept, such as “pickup monitor,” “order board,” and “status display” for a single user interface element. This variation complicated traceability and increased the chance of logical gaps. When a short glossary was introduced at the top of the prompt and the model was required to reuse those terms, the inconsistency decreased immediately.

5. What pre-post processing was useful for structuring the import prompts, then summarizing the output?

- Before generation, three steps proved valuable. First, a brief topic discovery pass created a shared outline and vocabulary. Second, a simple “include and exclude” list made scope explicit and reduced ambiguity. Third, short guidance about stakeholder priorities told the model which trade-offs to prefer.
- After generation, three checks were consistently helpful. First, a structured quality check confirmed the presence of the required sections, such as preconditions, main flow, and alternative flows, and flagged omissions. Second, a scope conformance check prevented the reappearance of excluded features. Third, a stakeholder sanity review assessed whether the draft genuinely reflected the agreed priorities rather than only mentioning them.

6. Did you find any best/worst prompting strategies?

- The most effective strategy was to separate work into deliberate phases. We first asked the model to generate a skeletal outline that reflected stakeholder priorities and scope, and only then asked it to fill each section in turn. This two-step approach produced concise sections, reduced redundancy, and made stakeholder review more straightforward. A second effective strategy was to require evidence for every substantive claim. By asking the model to pair each requirement with a brief rationale and a clear stakeholder or source reference, we converted vague statements into testable assertions and exposed unsupported assumptions early. A third effective strategy was to restate the complete set of governing constraints whenever we requested a consolidated or final output. This prevented the conversation from accumulating contradictions and ensured that later drafts did not silently violate earlier rules.
- The least effective prompts were open-ended brainstorming requests after scope had already been decided, because they expanded the solution space and reintroduced deferred features, which prolonged editing and delayed agreement. A second ineffective strategy was the omnibus prompt that attempted to combine discovery, drafting, and formatting in a single pass; this increased ambiguity and produced outputs that were difficult to verify against the agreed constraints.