Part A – Theory

1. What are requirements candidates, and why shouldn't they be treated as final requirements immediately after elicitation?

Requirements candidates are unvalidated statements drawn from elicitation activities — interviews, workshops, observations, document reviews — that *might* become formal requirements after further analysis. Think of them as hypotheses: they represent what stakeholders said they need, but they haven't yet been scrutinized for quality, feasibility, conflicts, or completeness.

They should not be treated as final requirements immediately because elicitation captures raw, often messy information. A stakeholder's statement may be ambiguous ("the system should be fast"), compound ("book, pay, and notify the patient"), based on a solution rather than a need, or simply unverifiable as stated. Promoting candidates to formal requirements without analysis risks building to vague or contradictory specifications, which causes rework downstream. The pipeline — elicitation inputs → notes/transcripts → candidate statements → analysis & validation → approved requirements — exists precisely because that intermediate validation step is where quality is added. A candidate also lacks traceability and acceptance criteria until someone has deliberately attached them; without those, you can't verify that the requirement was ever met.

2. The role of stakeholders in elicitation, and two risks of missing key stakeholders

Stakeholders are the primary source of "truth" in requirements work. Different stakeholders contribute different kinds of knowledge: end users describe workflows and pain points, sponsors define success measures and constraints, IT operations surface technical limitations, and compliance teams flag regulatory boundaries. Because no single person holds the complete picture, elicitation quality is fundamentally bounded by *who you talk to*. Missing a stakeholder category means that entire dimension of need — or constraint — simply doesn't enter the candidate pool.

Risk 1 — Late-breaking constraints. If a compliance or security stakeholder is excluded early on, their requirements (audit logging, data privacy rules, access controls) surface only after significant design or build work is done. Retrofitting compliance constraints is expensive and sometimes forces architectural rework.

Risk 2 — Rework due to "surprise" stakeholders. A stakeholder who was not consulted during elicitation but has veto power over the solution (a clinic director, a legal team, a key vendor) can reject or significantly alter scope late in delivery. This creates wasted effort on features that must be changed or removed, and erodes trust in the requirements process.