Defying Gravity in Testing

@AssumptionBuster I'm trying to reproduce a 'remaining slots feels wrong' issue using fill → cancel → reserve quickly. List pitfalls, then suggest deeper paths to detect lingering state after transitions.

- Logic and assumptions: The approach assumes fill→cancel→reserve transitions will update remaining slots immediately and atomically. Underlying assumptions include synchronous in-memory state updates, single-threaded event ordering, and no optimistic UI drift.

- Adversarial critique — assumption weaknesses: Race conditions between reservation creation and cancellation can leave remaining slots inconsistent. FR-24 and FR-31 require UI refreshes, but this ignores delayed rendering or missed event propagation.

- Persona-based debating: As an admin who overrides remaining slots, could manual edits create slot mismatches? As a rapid concurrent-user scenario, could optimistic updates cause temporary negative or stale slots?

- Reflective challenge: Which timing, event-ordering, or UI-update assumptions must be justified for this reproduction? Clarify those assumptions to enable sharper identification of logical gaps and edge cases.

Type here… (e.g., '@Brainstormer Find unseen exploration paths in the uploaded SRS') ➤