

Project Management Case Studies: Patterns and Tradeoffs



Practice evaluating real decisions in documented projects, not retelling stories



Apply four lenses to every case: context, decision point, tradeoff, outcome



Build pattern recognition across projects to speed up early-career judgment



Separate recorded facts from hindsight interpretation when reading cases

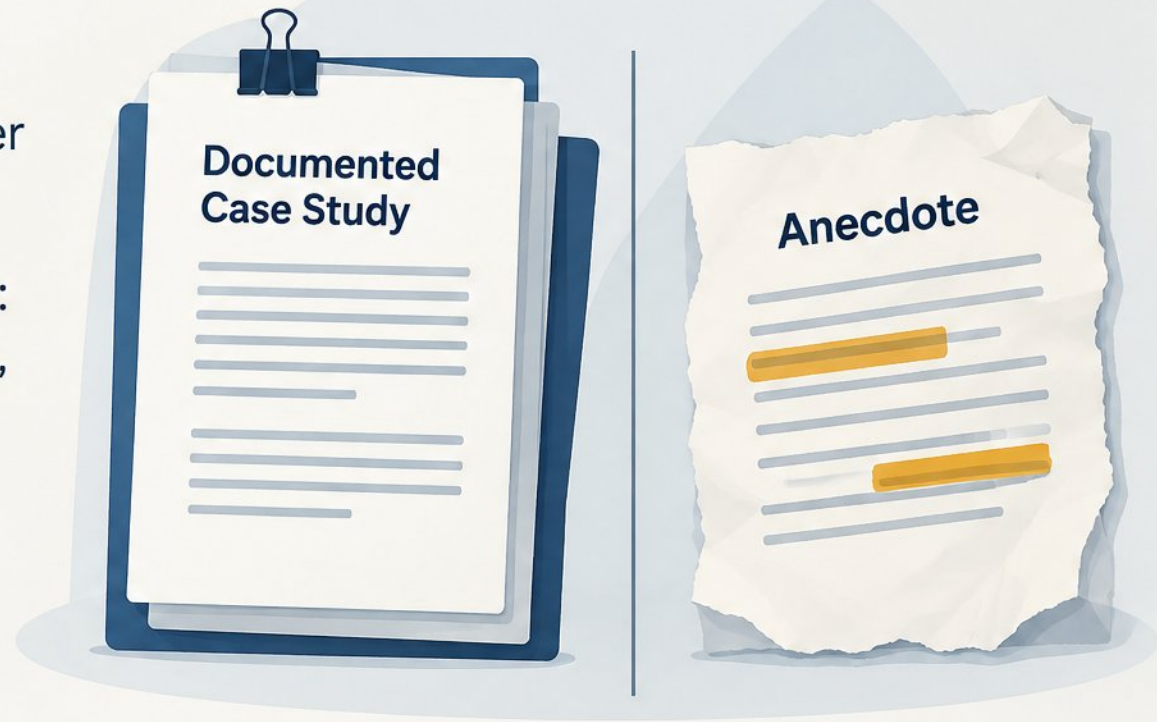


Roadmap: evaluation framework, five case patterns, cross-case synthesis, practice



Cases as Evidence, Not Anecdote

- Documented case study: structured evidence you can verify against records
- Anecdote: selective story told after the fact, shaped by hindsight
- Tradeoff categories in this course: scope, schedule, cost, quality, risk, benefits
- Coordinator decision points: schedule compression, change requests, resource conflicts, escalation timing
- Triple constraint as starting lens; quality, benefits, and risk often stand apart



The Five-Step Case Evaluation Framework



Judging Decisions Without Hindsight Distortion



Hindsight bias: outcomes feel predictable only after they are known



Outcome bias: identical decisions rate better when they succeed



Near-miss bias: lucky escapes are filed as successes and never learned from



Countermeasures: judge against information available at the time



Capture decisions while events are fresh — avoid selective memory in reviews



Pattern 1: Schedule Pressure and Scope Creep



Optimistic estimates plus unfrozen scope trigger late-stage compression



Key decision points: accept aggressive deadline, allow informal scope additions, delay escalation



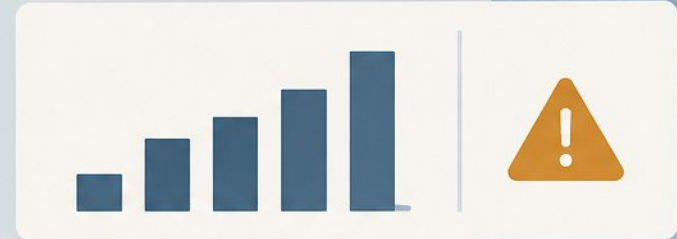
Tradeoffs: speed vs. quality; stakeholder goodwill vs. team sustainability



Track early warnings: falling float, unapproved changes, rising overtime



Ask: what would a better call at the inflection point look like?



Case Clinic: The Chunnel and Denver Baggage System

Channel Tunnel



Denver baggage system



Emirates Bank



Channel Tunnel: scope not fully assessed early; IGC held scope control without funding authority



Denver baggage system: rigid deadline, underestimated complexity, new features added after start



Shared pattern: scope decisions made outside an integrated change process with funding attached



Counter-case: Emirates Bank held schedule, resisted scope change through a merger, slipped only 7%

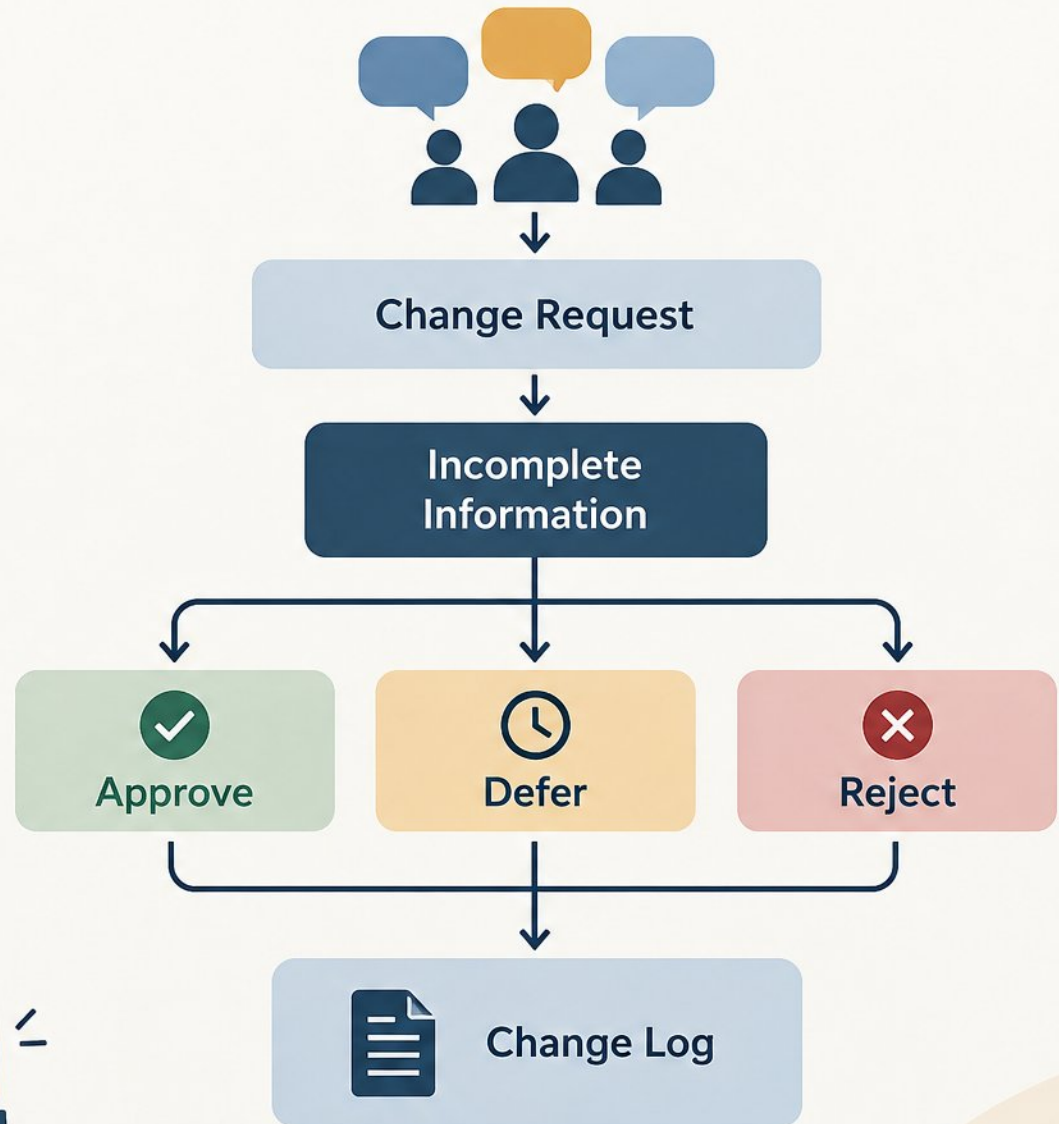
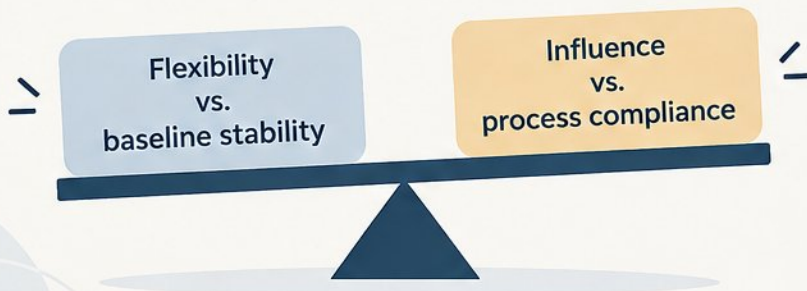


Discuss: locate the one decision point in each case where the outcome was still open

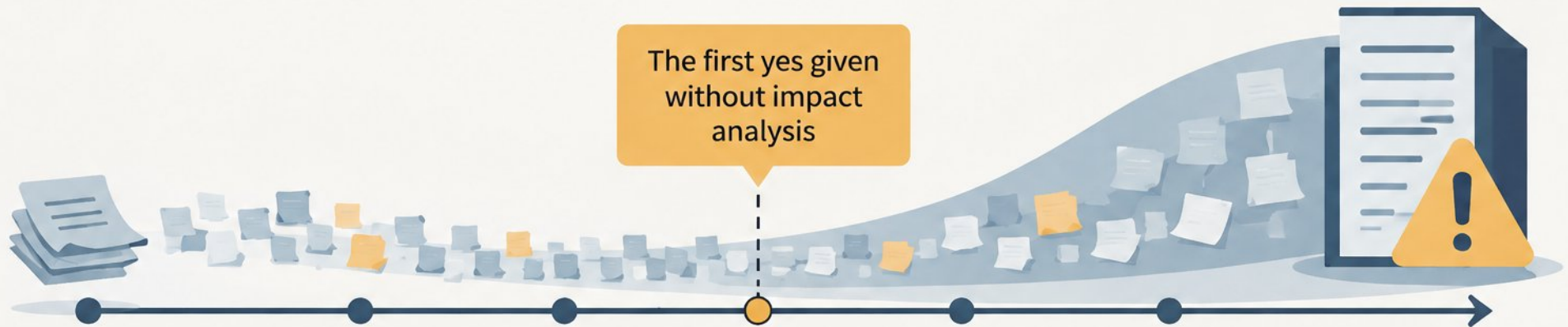


Pattern 2: Stakeholder Conflict and Change Requests

- Conflicting stakeholder priorities surface as change requests
- Decision point: approve, defer, or reject under incomplete information
- Tradeoffs: flexibility vs. baseline stability; influence vs. process compliance
- Minor approvals accumulate into shadow scope absent from baselines
- Coordination: impact assessment, change log, approval authority, escalation paths



Case Clinic: Death by a Thousand Changes



- Pinnacle Insurance: 147 "minor" changes approved without impact analysis
- Scope grew ~35% with no baseline updates; 40% over budget, six months late
- PM's inflection point: the first yes given without impact analysis
- Contrast: LASCAD succeeded when 23 of 24 stakeholder risks were mitigated
- Discuss: which changes would impact assessment have stopped — and which were worth doing?

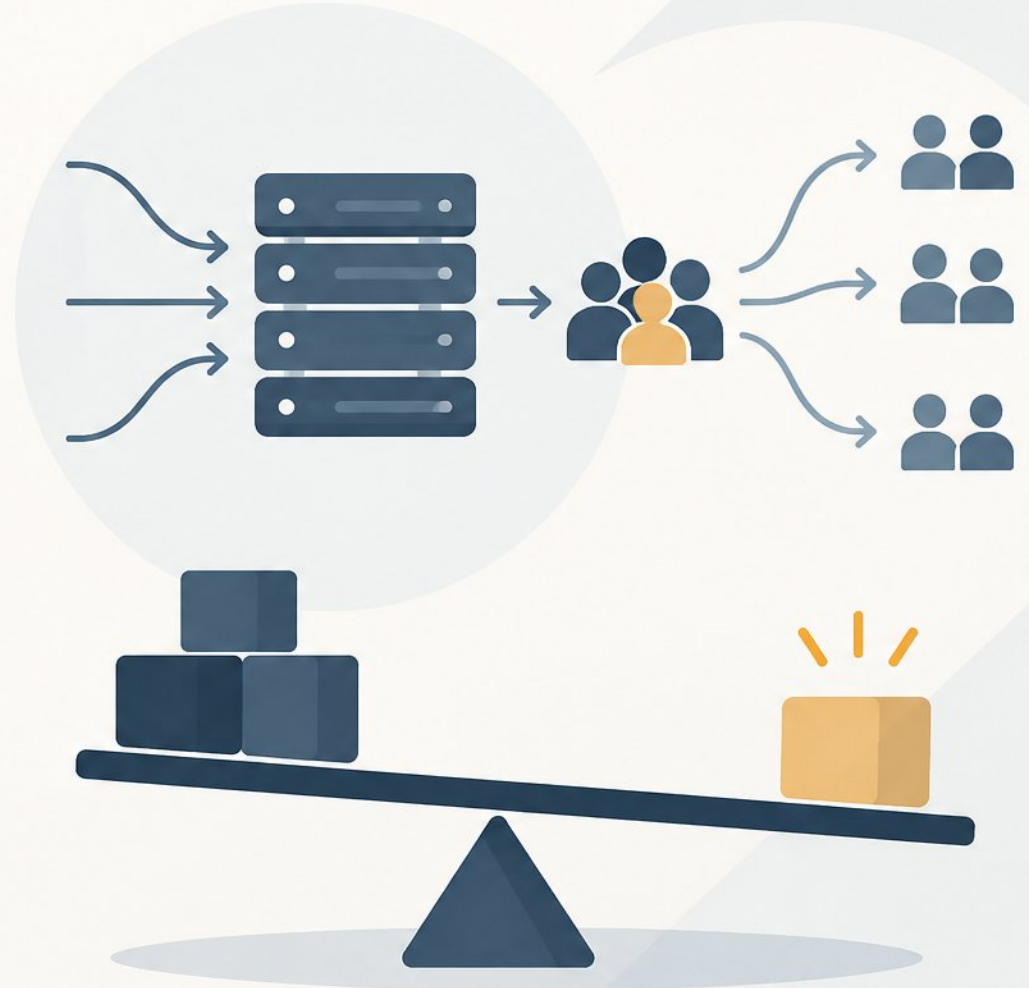
Pattern 3: Resource Constraints and Resourcing Tradeoffs

- **Shared specialists** create hidden queues, context switching, and idle time
- **Decision point:** borrow an expert, train internally, re-sequence, or add buffer capacity
- **Tradeoffs:** short-term delivery speed vs long-term capability; cost vs risk
- **Early signals:** rising wait times, overallocated named resources, queue growth before one role
- **Post-hoc test:** was the constraint actually the bottleneck?
- **Evidence:** adding an expert who shortens long-tail tasks beat adding capacity at the bottleneck



Case Clinic: Shared Specialists and Bottlenecks

- Four shared test platforms forced night testing and multi-week delays
- Response: dynamic feature teams swarming the biggest work items
- Documented results: 63% more throughput, 42% fewer defects, \$1.73M+ saved
- Simulation: an expert shortening long-tail tasks beat adding bottleneck capacity
- Discussion: which decision here was reversible, and which was not?



Pattern 4: Risk Response and Escalation Timing

- Known risks linger when ownership and thresholds are undefined
- Decision point: escalate, absorb, or assign — and when
- Tradeoff: autonomy vs. transparency; early cost vs. future impact
- Late escalation narrows options — the same risk costs more later
- Predefined thresholds and alarm bells make escalation routine, not personal



Case Clinic: Warnings That Were Already on Record



- Marble Arch Mound: repeated cost warnings; checks bypassed to hit timetable
- NCSX: initial risk plan and contingency inadequate; better visibility too late
- Queensland Health payroll: same issues raised across phases, never resolved
- Common structure: unclear ownership, reconstructed reporting, untested escalation



Discussion: at what point was escalation still cheap in each case?

Pattern 5: Quality, Rework, and Technical Debt



Deferring quality work compounds rework later



Decision: accept known defects to hit milestone, or delay



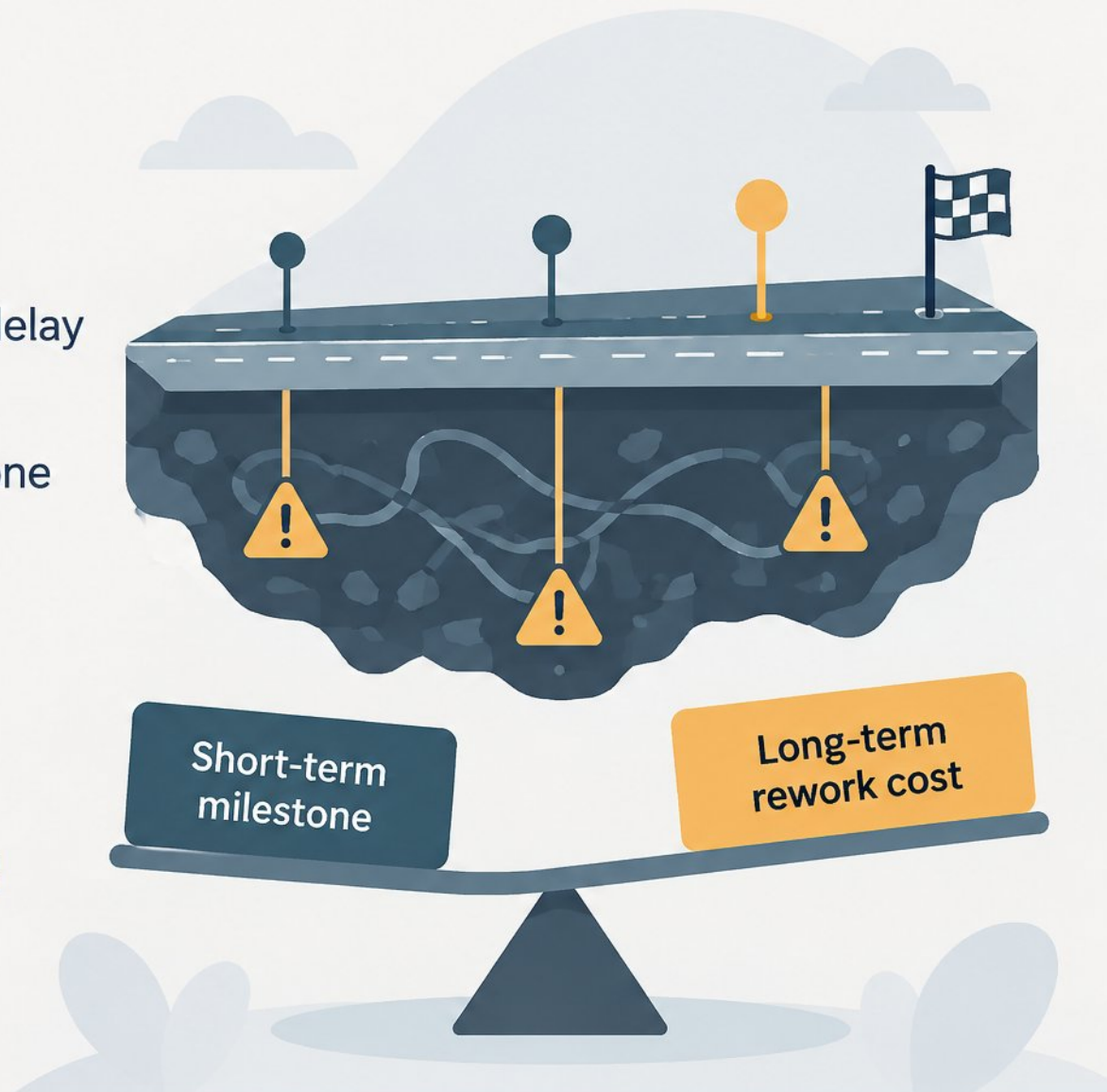
Tradeoff: short-term milestone vs long-term rework cost



Defect data reveals where and when problems began



Ask why the tradeoff looked reasonable at the time



Case Clinic: The Compounding Cost of Deferred Quality

Mining technology delivery



Construction



Aircraft development



- Mining technology delivery: 17%+ of revenue shifts from cost to profit with right-first-time
- Construction: repeated non-conformances common; failure costs exceed prevention and appraisal
- Aircraft development: premature phase overlap and unfrozen data drove asynchronous rework
- Quality problem-solving often oversimplified; corrective actions fail to prevent recurrence
- Discussion: which milestone tradeoff in these cases was genuinely reversible?



Comparing Patterns and Applying the Framework to Your Own Decisions

- Cross-case shared causes: late information, unclear ownership, untested escalation
- Separate genuinely unavoidable tradeoffs from those created by process gaps
- Test each heuristic's boundary conditions before reusing it in new contexts
- Log live decisions: context, options, rationale, reversal criteria, review trigger
- Monthly habit: one structured case review, process judged separately from outcome

Cross-Case Comparison

Shared Causes



Distinct Triggers



Your Decision Log



Context



Decision
Point



Tradeoff



Outcome

PersonWise.

PersonWise • AI digital-human course platform



Scan the code or click anywhere to watch this course live, with voice Q&A

personwise.ai/t/d476657c/public