

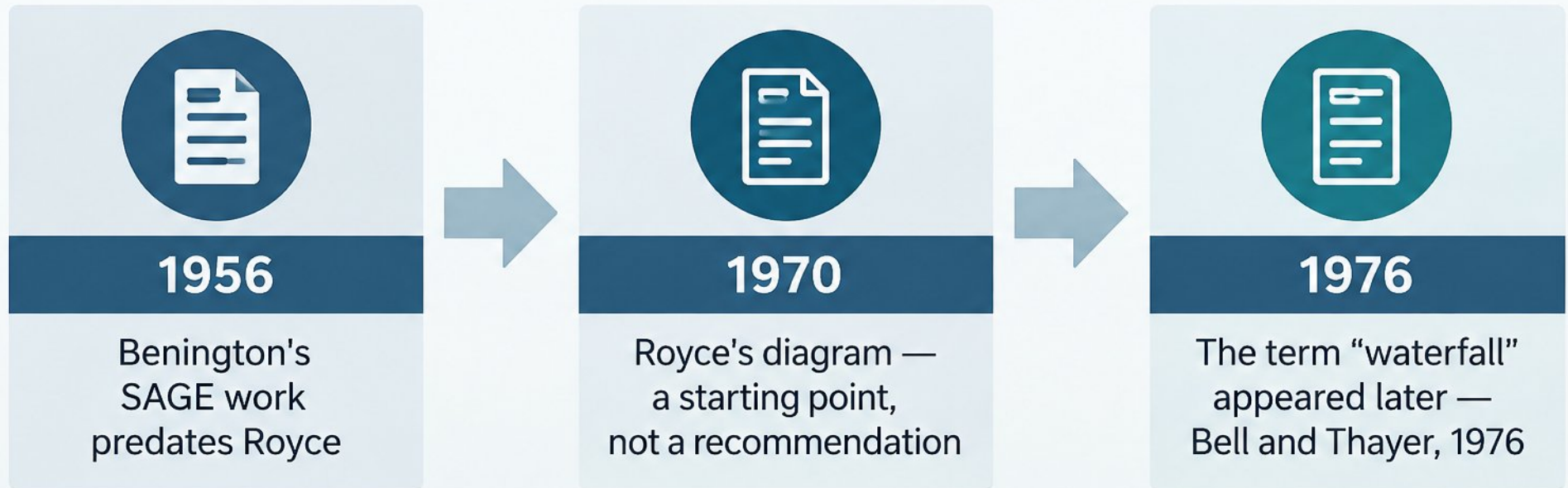
# Waterfall Project Management: Foundations for Staged Delivery

---



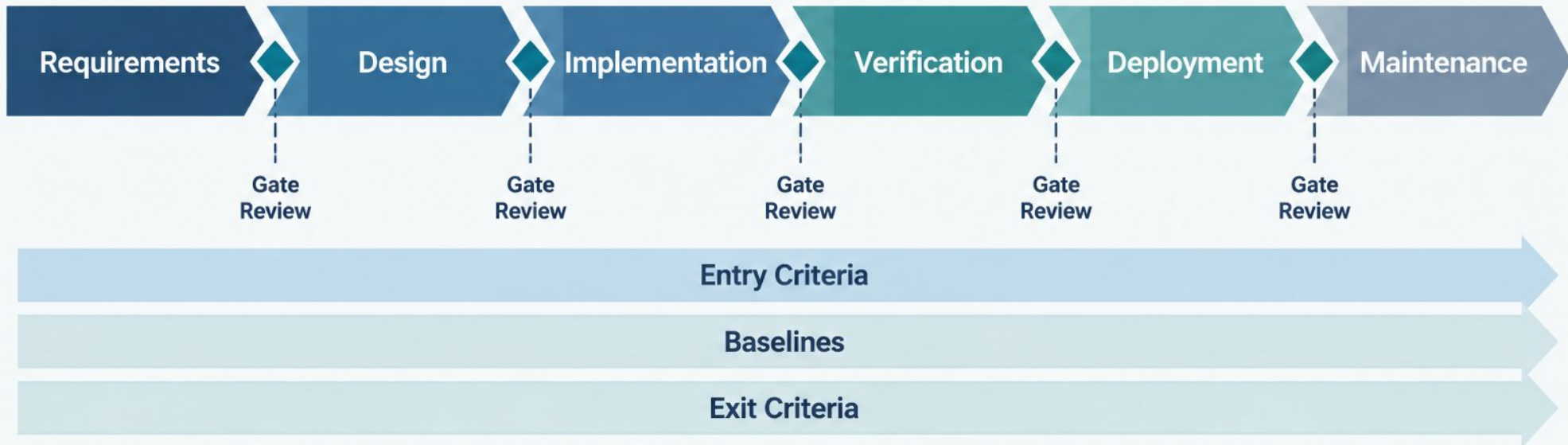
- Plan-driven, sequential lifecycle: each phase approved before the next begins
- Course goal: master phases, gates, baselines, change control, and handover
- For project managers, coordinators, business analysts, and staged-delivery teams
- Royce's 1970 paper called one-pass sequential delivery "risky and invites failure"
- Term "waterfall" coined later by Bell and Thayer, 1976
- Still common in construction, aerospace, healthcare, finance, and government

# A Brief History and the Persistent Misconception



- 1956: Benington's SAGE work predates Royce
- 1970: Royce's diagram — a starting point, not a recommendation
- Royce's five risk-reduction steps: design first, document, do it twice, control testing, involve the customer
- The term "waterfall" appeared later — Bell and Thayer, 1976
- The "frozen waterfall" critique targets a model Royce never advocated

# Core Concepts: Phases, Gates, and Baselines



## Phases

Phases: requirements, design, implementation, verification, deployment, maintenance; V-model pairs build with verification



## Entry criteria

Entry criteria: approved requirements, confirmed budget, identified resources, signed scope boundaries



## Exit criteria

Exit criteria: standards a phase must meet — not the deliverable itself



## Gate reviews

Gate reviews: formal go, no-go, hold, or recycle decisions with real authority to pause



## Baselines

Baselines: approved scope, schedule, and cost references; configuration management controls versions

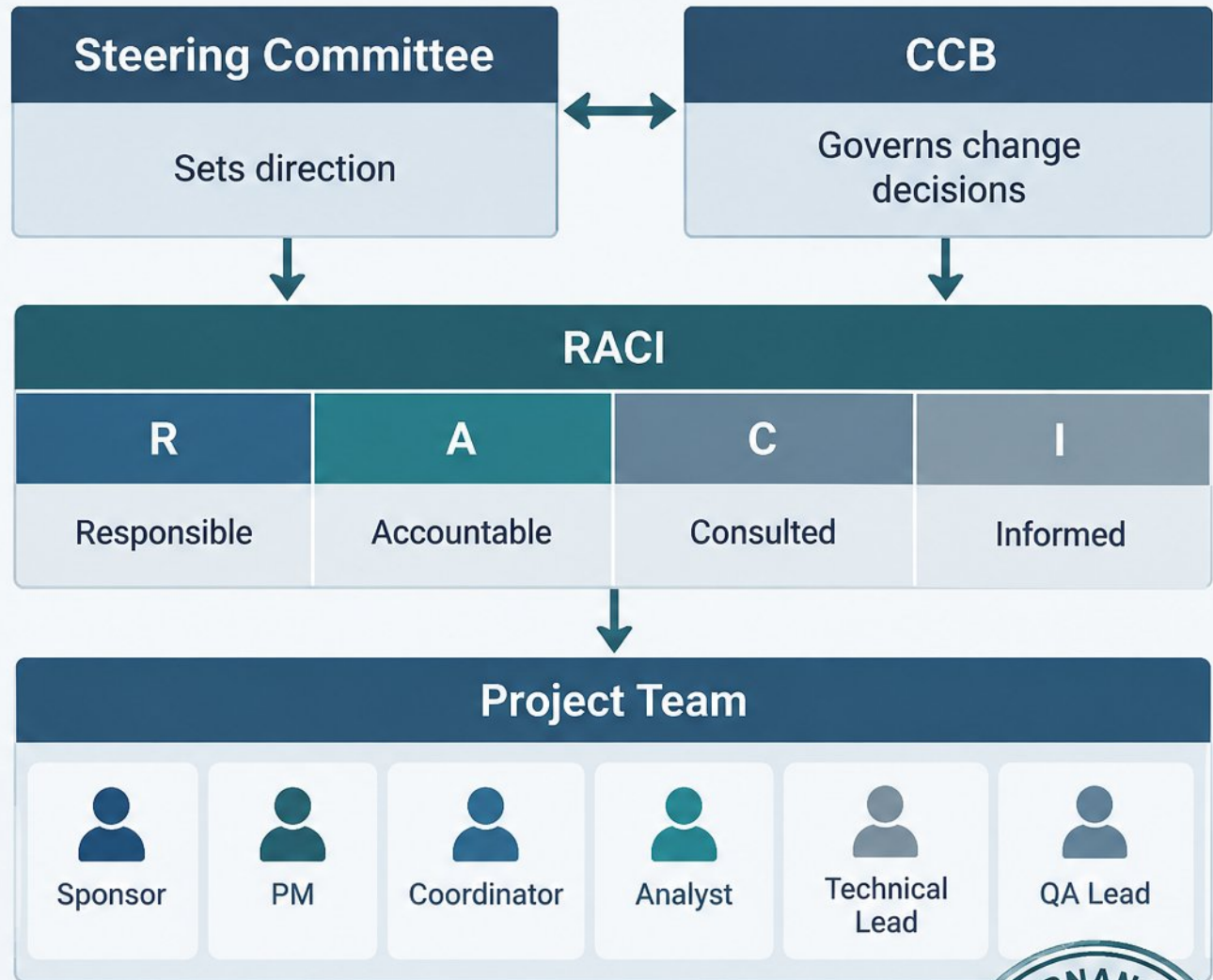


## Tollgate

Tollgate: formal submission package, structured evaluation, documented sign-off, audit trail

# Roles, Responsibilities, and Governance

- Core roles: sponsor, PM, coordinator, analyst, technical and QA leads
- Steering committee sets direction; CCB governs change decisions
- RACI maps Responsible, Accountable, Consulted, Informed per activity
- One Accountable per activity: two accountable people means zero accountability
- Cap Consulted at about three; Informed should be the most common code
- Analyst bridges business needs into requirements and traceability



# Requirements and Design in Staged Delivery



Elicit requirements via stakeholder interviews, structured workshops, and specification standards



Formal sign-off converts approved requirements into the baseline; changes then go through change control



Requirements Traceability Matrix (RTM): bidirectional traceability from stakeholder need to design, code, and test cases



Design outputs: architecture diagrams, detailed specifications, prototypes, interface definitions



Traceability best practices: start early, use unique IDs, review at major gates



Common pitfalls: ambiguous requirements, gold plating, late discovery of misalignment

## Requirements Traceability Matrix (RTM)

| Stakeholder Need | System Requirement | Design Element | Test Case |
|------------------|--------------------|----------------|-----------|
|                  |                    |                |           |
|                  |                    |                |           |
|                  |                    |                |           |
|                  |                    |                |           |

**End-to-end traceability links stakeholder needs to design, code, and test cases.**

# Planning, Scheduling, and Estimation

1

**WBS:** decompose deliverables into discrete, estimable work packages with clear completion criteria

2

**CPM:** the longest dependent sequence sets minimum duration; critical path tasks have zero float

3

**Forward and backward pass:** compute early/late start and finish to find float

4

**Estimation:** analogous (fast), parametric (data-driven), bottom-up (accurate), three-point/PERT

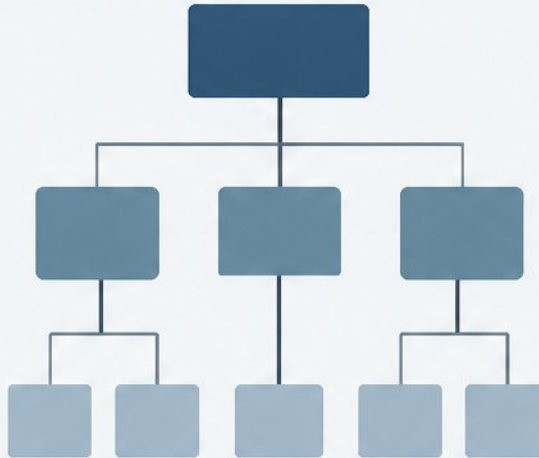
5

Resource leveling and smoothing  
balance workloads and protect critical tasks

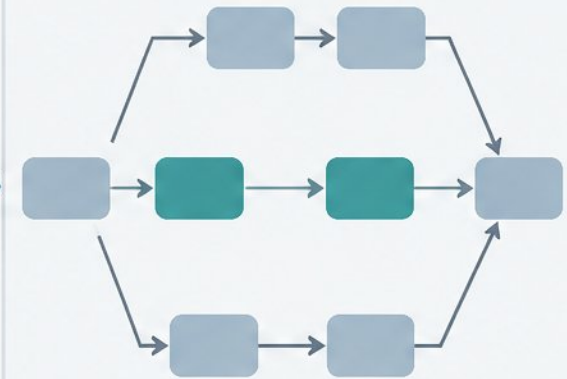
6

Approved scope, schedule, and cost baselines govern monitoring and change control

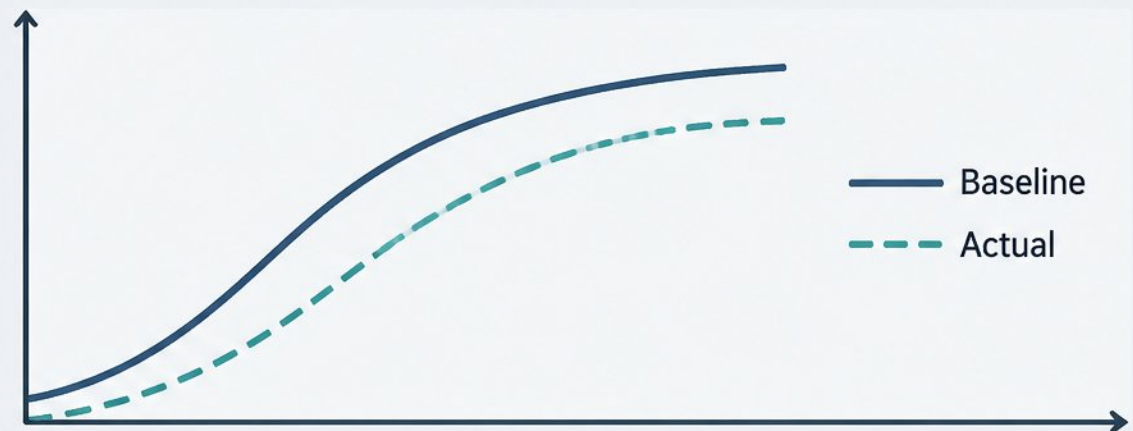
## Work Breakdown Structure (WBS)



## Critical Path Method (CPM)



## Baseline vs. Actual



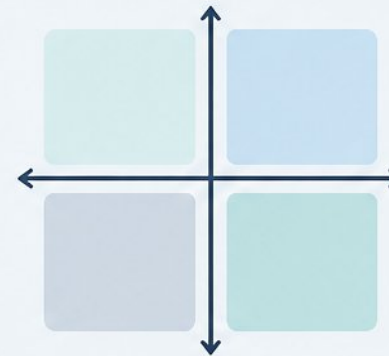
# Execution, Monitoring, and Control

- Track progress against baseline scope, schedule, and cost; catch variances early
- Core EVM metrics:  $PV$ ,  $EV$ ,  $AC$ ,  $SV = EV - PV$ ,  $CV = EV - AC$
- Indices:  $SPI = EV \div PV$ ,  $CPI = EV \div AC$ ; below 1.0 signals trouble
- $CPI\ 0.80$  means 80 cents of value per dollar spent;  $SPI\ 0.90$  means 90% pace
- Forecasting:  $EAC = BAC \div CPI$ ;  $TCPI = (BAC - EV) \div (BAC - AC)$
- Behind but under budget: accelerate; over but ahead: cost discipline

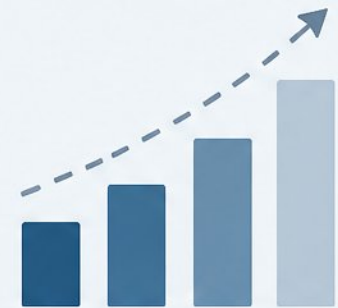
Earned Value Curves



SPI / CPI Quadrant



Estimate at Completion



EXECUTE

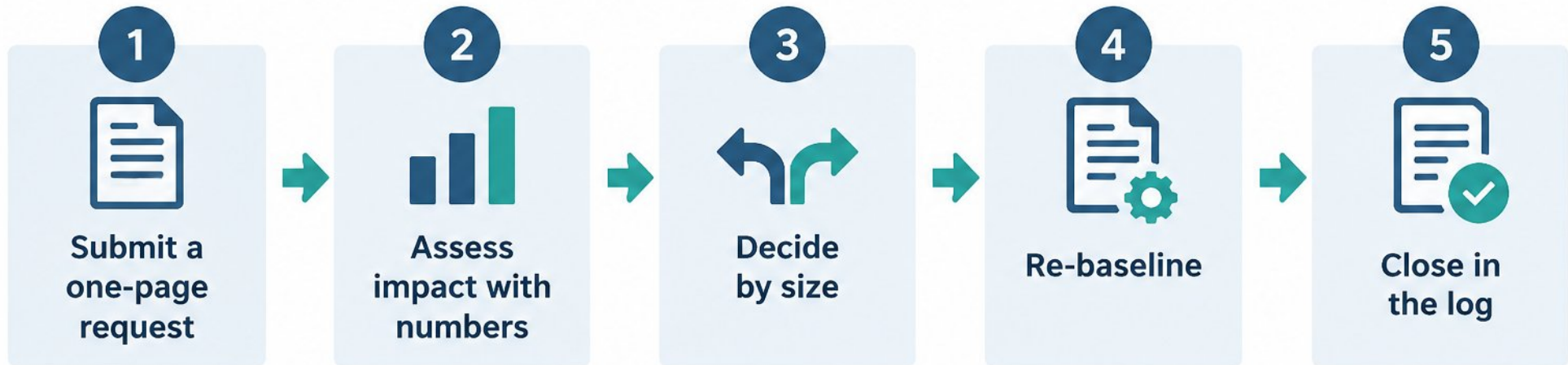
MONITOR

CONTROL



Baseline vs. Actual  
Variance

# Change Control and Scope Management



- Five steps: submit a one-page request; assess impact with numbers; decide by size; re-baseline; close in the log
- Minor changes (under 2 days or 2% of budget) go to the project manager
- Major changes go to the sponsor or change control board; contractual ones need a signed change order
- Scope creep is the same request absorbed without the process
- Log one line per request: ID, date, impact, decision, decider — including rejections

## Change Log

| ID | Date | Impact | Decision | Decider |
|----|------|--------|----------|---------|
|    |      |        |          |         |
|    |      |        |          |         |

# Risk and Issue Management in Stage-Gated Projects

Idea

Gate

Concept

Gate

Development

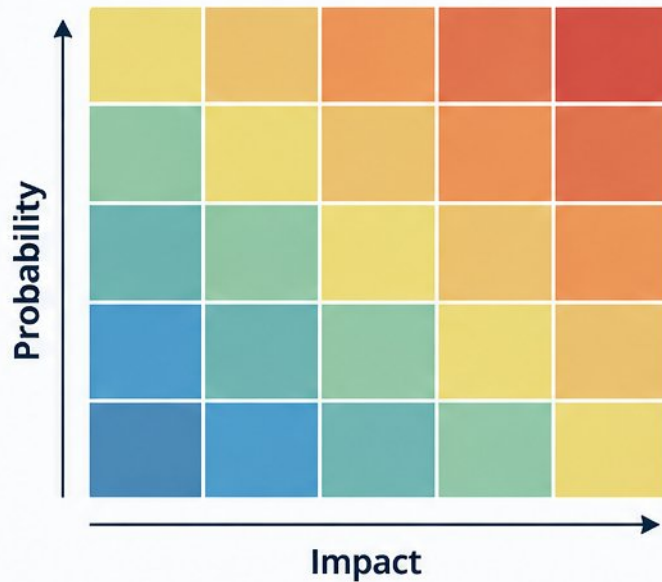
Gate

Validation

Gate

Launch

Risk Matrix (Probability vs. Impact)



Risk Register

| Risk | Probability | Impact | Owner | Response | Target Date |
|------|-------------|--------|-------|----------|-------------|
|      |             |        |       |          |             |
|      |             |        |       |          |             |
|      |             |        |       |          |             |
|      |             |        |       |          |             |
|      |             |        |       |          |             |
|      |             |        |       |          |             |
|      |             |        |       |          |             |



Risk register:  
probability, impact,  
owner, response,  
target date



Plan responses—  
avoid, mitigate,  
transfer, accept—  
before the phase opens



Issues are materialized  
risks: route through  
change control  
and governance

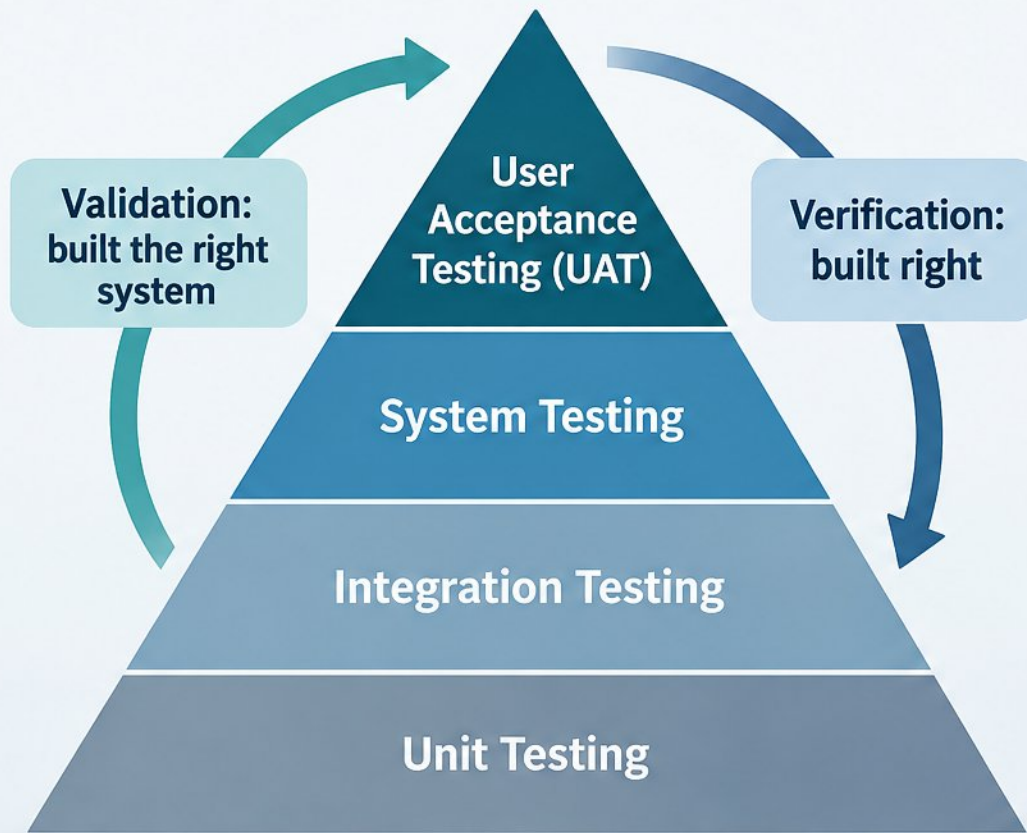


Unresolved high risks  
should block or  
condition a gate  
decision



Retire risks early  
with prototypes and  
pilots; capture lessons  
at each gate

# Verification, Testing, and User Acceptance



**Testing levels:** unit, integration, system, and user acceptance (UAT)



**Verification:** built right;  
**validation:** built the right system



**UAT team:** real users, power users, compliance, IT operations



**Daily defect triage:** assign severity; critical defects block go-live



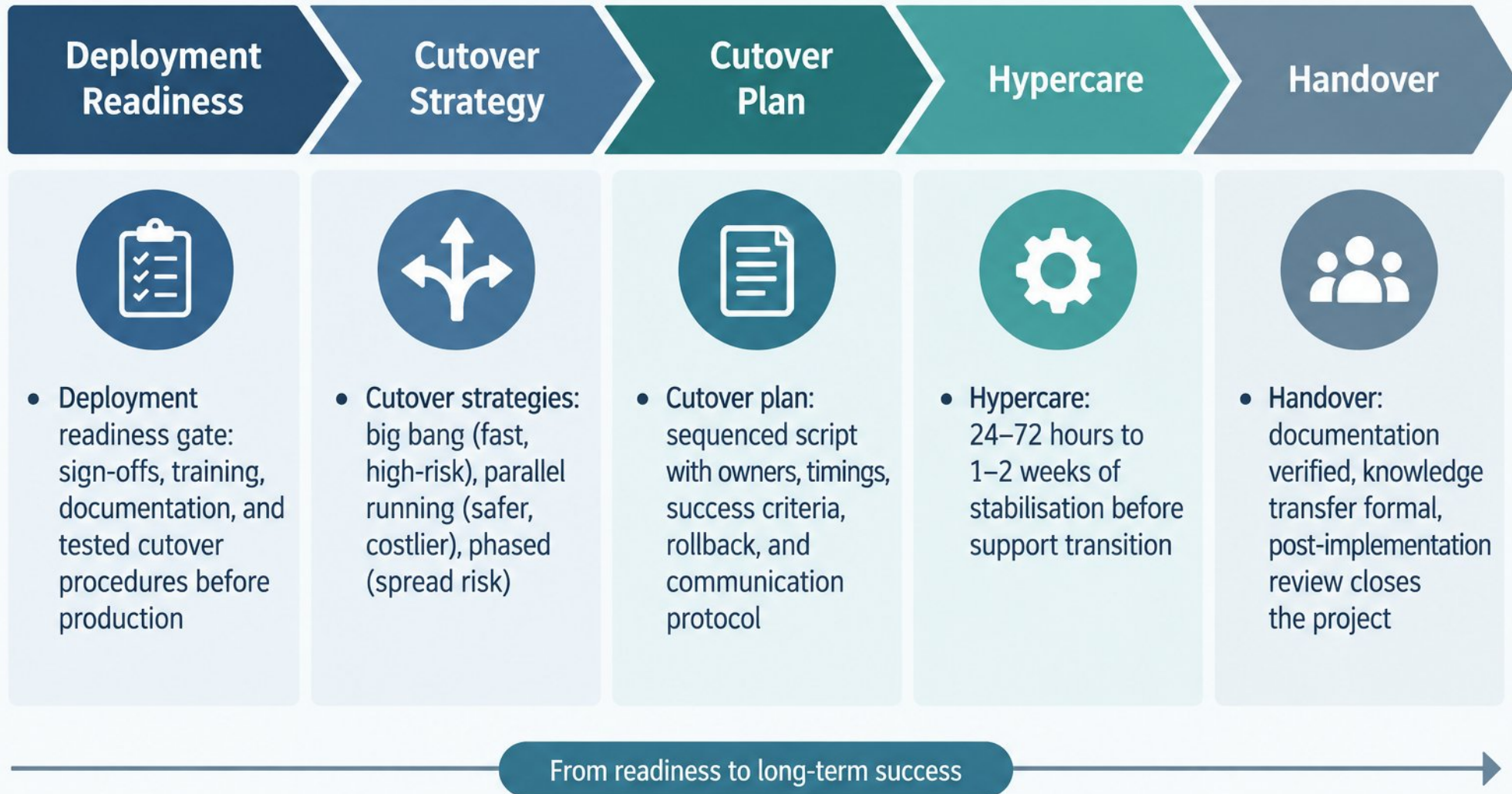
**UAT sign-off:** test summary, known defects, authorization to proceed



Acceptance testing checks contract; UAT checks operational fitness

# Deployment, Cutover, and Handover

A structured approach to move from preparation to stable operations.



# When to Choose Waterfall: Fit Criteria and Trade-offs

## Waterfall



Waterfall fits stable, well-understood requirements and fixed-price or fixed-scope contracts.



Choose Waterfall for regulatory constraints, physical build dependencies, and sequential infrastructure.



Waterfall can be 10–20% cheaper for stable scope.

## Agile



Choose Agile for evolving requirements, time-to-market pressure, and high uncertainty.

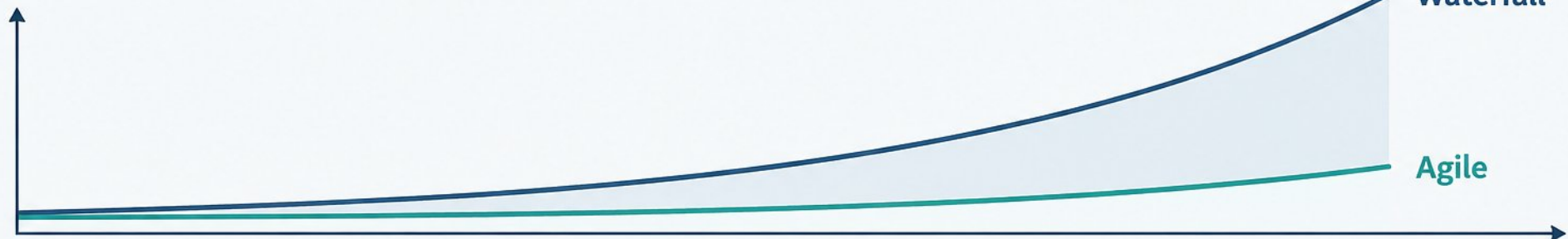


Cost of late change drives the choice: high for Waterfall, low for Agile.



Agile 15–25% cheaper for evolving scope.

### Cost of Late Change



**Hybrid reality: over 67% of large enterprises use blended frameworks.**

# Hybrid Delivery: Waterfall and Agile Combined



## Portfolio Governance

Requirements

Budget and  
Contracts

Release and  
Sign-off



## Release Planning

Stage-Gate

Stage-Gate

Change Control Boundary



## Team Delivery

Sprint

Sprint

Sprint



## Integration Management

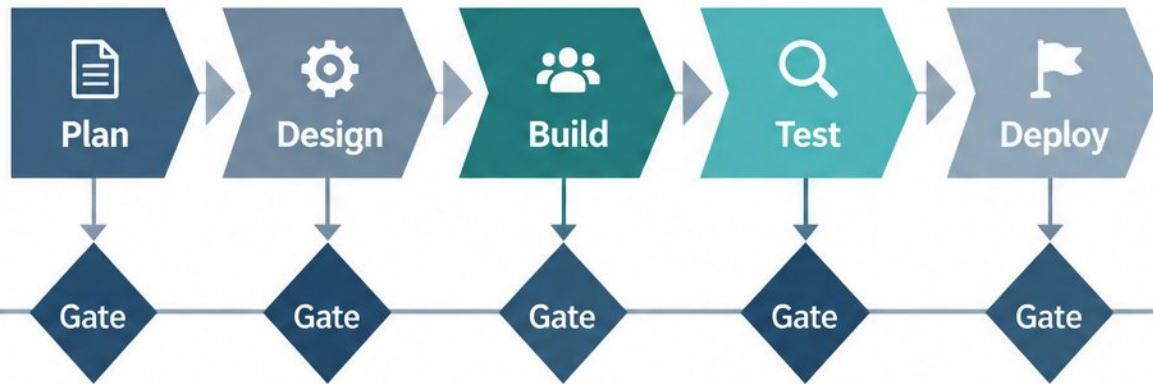
Integrate and Deliver

- Water-Scrum-Fall: Waterfall frames requirements, budget, and contracts; Scrum runs development sprints; Waterfall governs release and sign-off
- Agile + Stage-Gate suits regulated work: FDA, ISO 13485, DO-178C
- Four layers: Portfolio Governance, Release Planning, Team Delivery, Integration Management
- Change boundary: milestone or programme scope changes need formal control; sprint backlog priority changes do not
- Biggest failure mode: governance creep — Waterfall controls suffocating Agile teams
- Honest hybrid vs. disguised waterfall: can a Sprint Review challenge scope?

# Case Practice, Failure Modes, and Key Takeaways

## Case Walkthrough

Case walkthrough: staged ERP project, gates and baselines in action



## Key Failure Modes

Failure modes: unrealistic baselines, weak gates, late testing, unmanaged change



Unrealistic  
baselines



Weak gates



Late testing



Unmanaged  
change



**Weak gates always  
pass—willingness to pause  
makes "proceed" meaningful**

## Practical Checklist

Practical checklist: entry/exit criteria, named approvers, handoff checklists, gate reviews, change log

- ✓ Entry/exit criteria
- ✓ Named approvers
- ✓ Handoff checklists
- ✓ Gate reviews
- ✓ Change log



**Waterfall stays viable for  
stable, regulated work;  
match method to the work**



Change Log

# 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/aad976d4/public](https://personwise.ai/t/aad976d4/public)