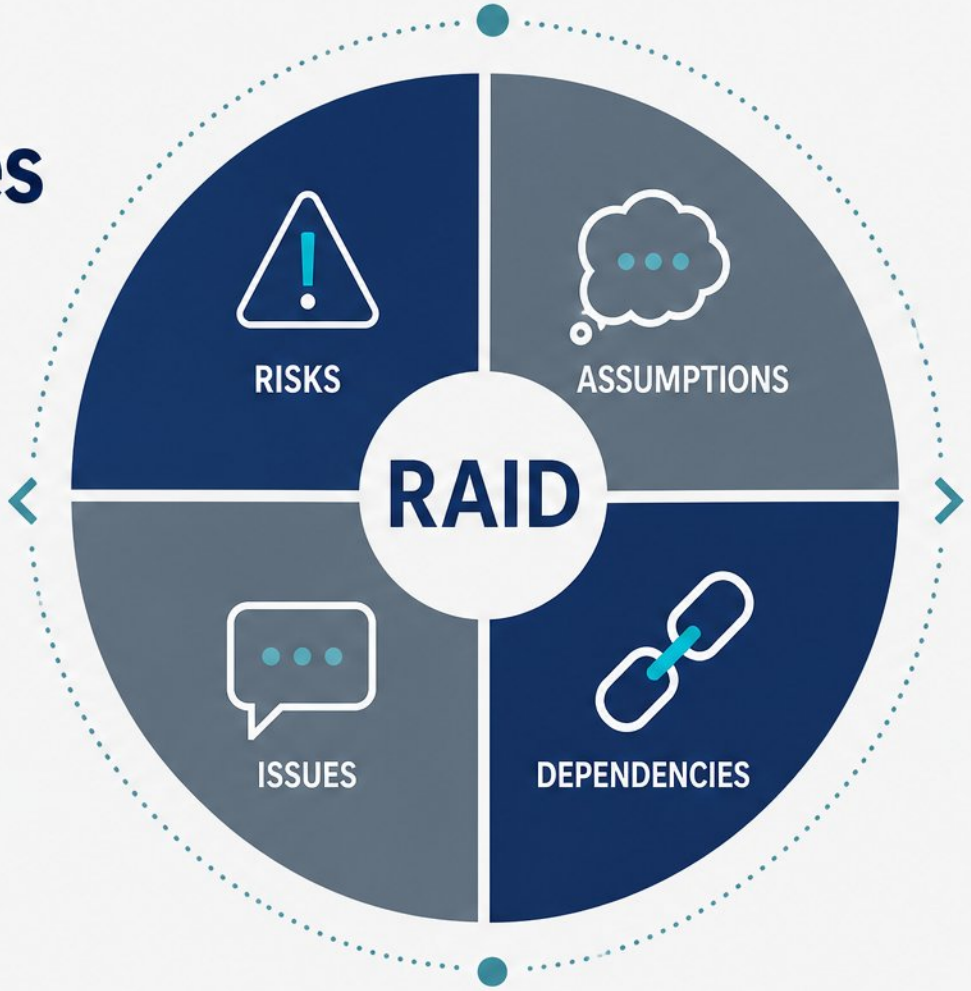
RAID Meaning in Project Management: Risks, Assumptions, Issues, and Dependencies

- RAID = Risks, Assumptions, Issues, Dependencies
- The four info categories most likely to derail delivery
- Lightweight governance layer, not a methodology replacement
- Works alongside waterfall, agile, and hybrid delivery
- Builds visibility, ownership, and better decisions
- For PMs, PMO practitioners, coordinators, delivery leads, students



Why RAID Matters for Delivery Governance

- Projects rarely fail because teams forget the work.
- They fail when uncertainty, blockers, and reliance go unsurfaced.
- RAID gives sponsors, PMOs, and teams one shared health reference.
- A live practice catches material risks earlier and cuts schedule slip.
- It turns scope events into managed decisions, not write-offs.
- It creates an auditable record of who raised, owned, and acted.
- Executive questions answered: what might go wrong, what we assume,
- what is happening now, and what we are waiting on.

ESCALATION PATH

Delivery Team → PMO / Project Leadership → Steering Committee

What Each RAID Element Means

Risks

uncertain future events affecting scope, schedule, cost, or quality; tracked by probability, impact, and owner

Assumptions

believed true for planning, unverified; must be validated or converted into risks

RAID

Issues

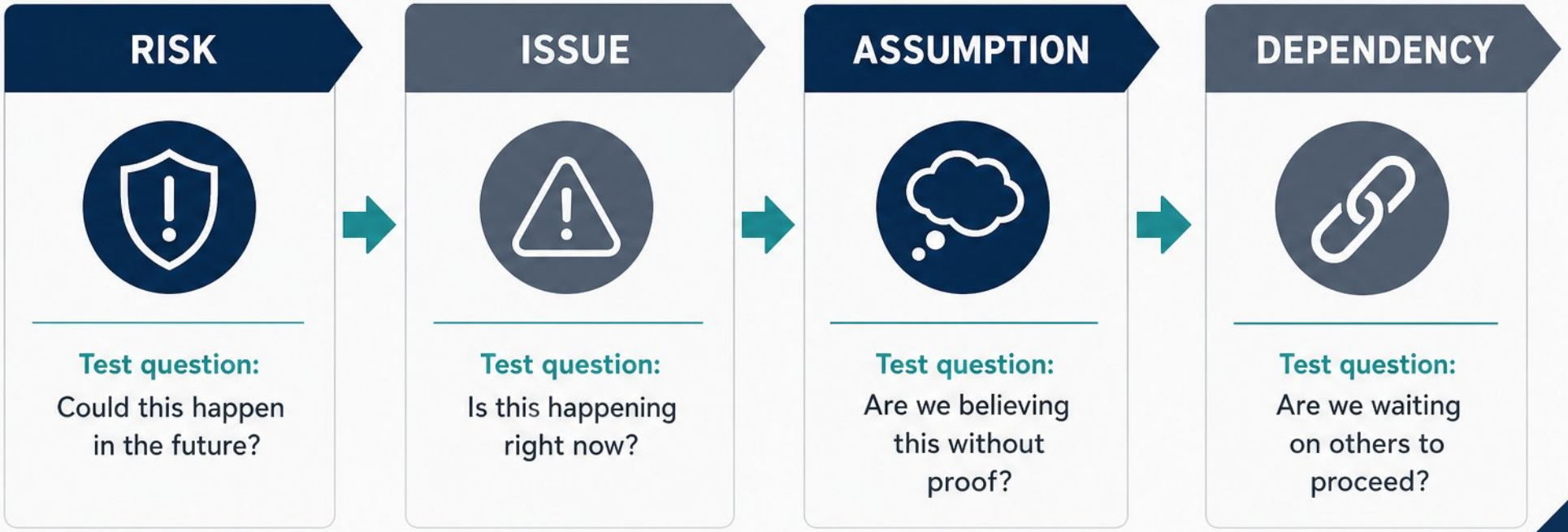
problems already occurring and requiring resolution; managed by severity, priority, and due date

Dependencies

deliverables or approvals you rely on; tracked by provider and required-by date

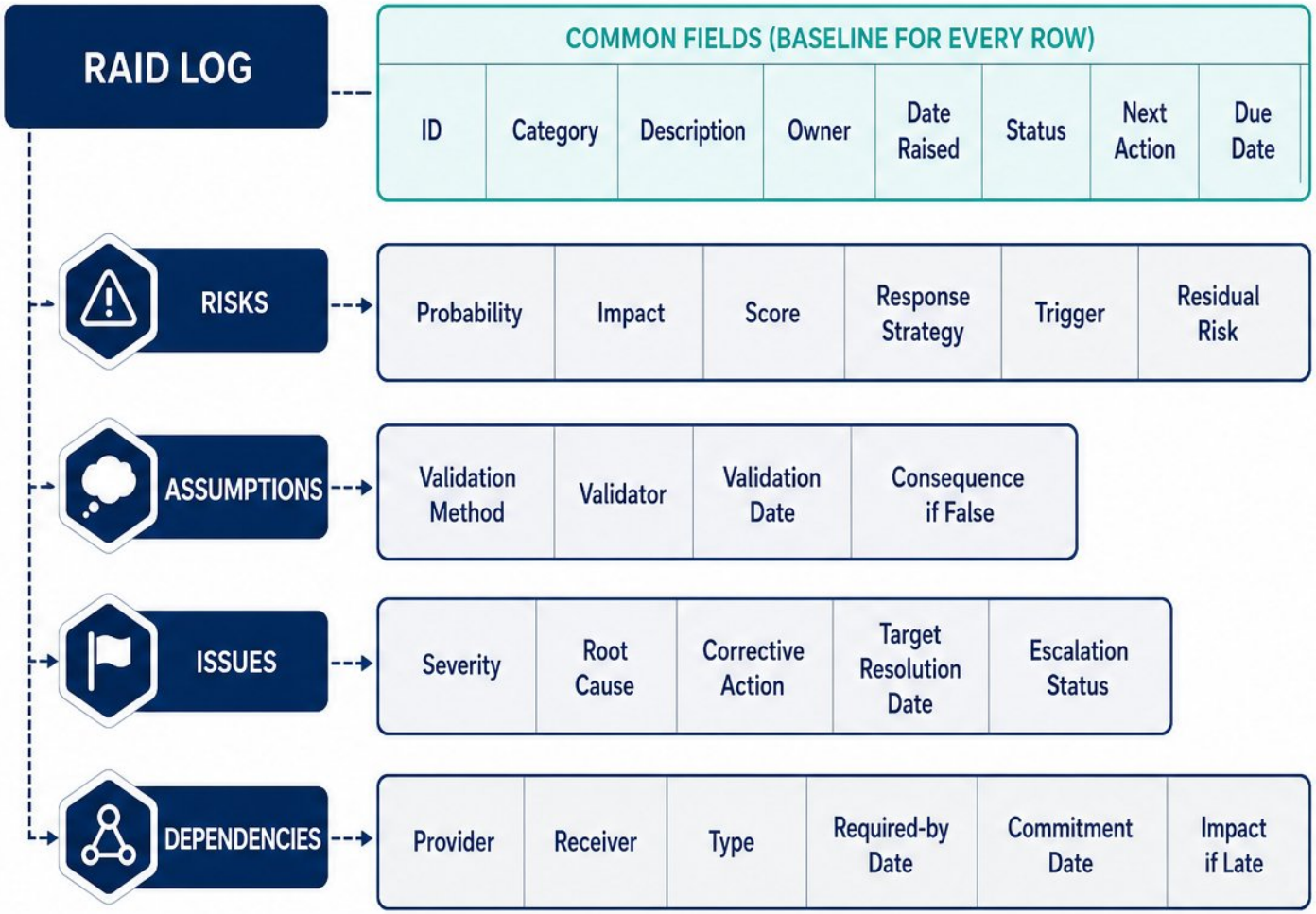
Test: risk may happen, issue is happening, assumption is unproven, dependency is outside your control

Distinguishing Risks, Issues, Assumptions, and Dependencies



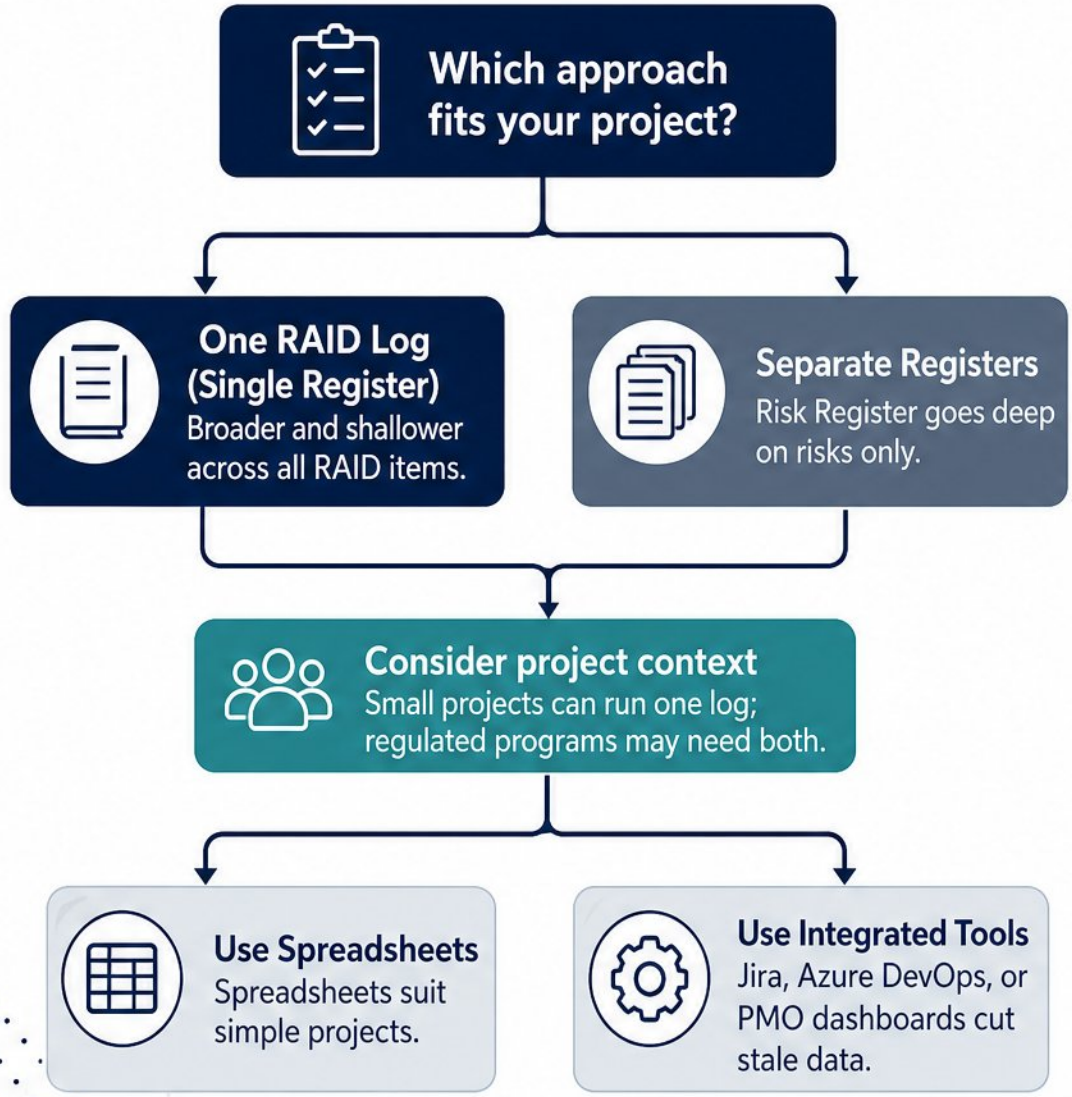
- Might happen = risk; happening now = issue
- Believed without proof = assumption; waiting on others = dependency
- Common errors: risks logged as issues, assumptions treated as facts
- Chains: false assumption → risk → issue → blocked dependency
- When a risk materializes, close it and open a linked issue
- Specific descriptions: state condition plus consequence, never vague labels

RAID Log Structure and Core Fields



- Baseline for every row: ID, category, description, owner, date raised, status, next action, due date
- Risks add probability, impact, score, response strategy, trigger, residual risk
- Assumptions add validation method, validator, validation date, consequence if false
- Issues add severity, root cause, corrective action, target resolution date, escalation status
- Dependencies add provider, receiver, type, required-by date, commitment date, impact if late

Practical Design Choices for RAID Logs



One RAID log is broader and shallower; a risk register goes deep on risks only

Small projects can run one log; regulated programs may need both

Spreadsheets suit simple projects; Jira, Azure DevOps, or PMO dashboards cut stale data

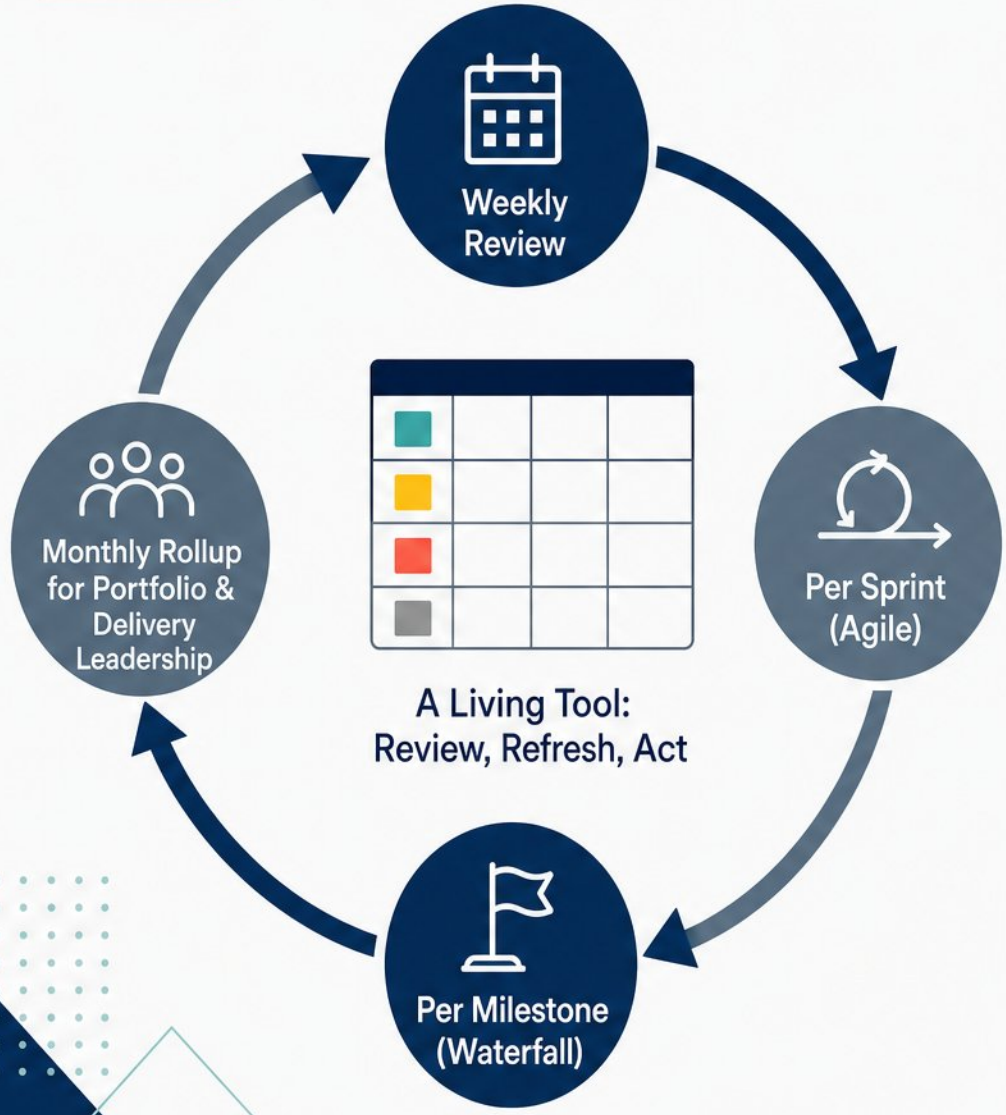
Drop any column not reviewed in status meetings

Hygiene: unique IDs, named owners, dated actions, archive (never delete) closed items

Building the RAID Log from Kickoff

- Start at kickoff: scope, objectives, constraints, assumptions, dependencies are fresh
- Use workshops, interviews, brainstorming, lessons learned, and checklist prompts
- Capture all four categories early; unspoken assumptions become late surprises
- Classify each item correctly; assign one named accountable owner, not a team
- Prioritize by impact, urgency, probability, and risk exposure

Maintaining the RAID Log as a Living Tool



- Review weekly; per sprint for agile, per milestone for waterfall
- Monthly rollup for portfolio and delivery leadership
- Walk the log in status meetings, planning, retros, and stage gates
- Close, convert, or escalate items with no movement in two reviews
- Link entries to tasks, milestones, resources, and decisions
- At project close, review which risks, assumptions, and dependencies failed

RAID in Agile, Waterfall, and Hybrid Delivery

Waterfall

Plan-driven delivery with stage-gate governance.

Risks | Assumptions | RAID | Issues | Dependencies

Cadence: At phase gates and reviews

Artifacts: Formal risk registers, RAID log, gate assessments

Key Focus: Predictability, control, and assumption validation

Agile

Iterative delivery with continuous inspection and adaptation.

Risks | Assumptions | RAID | Issues | Dependencies

Cadence: Daily stand-ups, sprint planning, refinement, reviews, and retrospectives

Artifacts: Lightweight RAID list, team backlog annotations

Key Focus: Transparency, adaptation, and impediment removal

Hybrid

Blends plan-driven governance with iterative execution.

Risks | Assumptions | RAID | Issues | Dependencies

Cadence: Governance gates and iterative cadences

Artifacts: Integrated RAID log, governance reports, team views

Key Focus: Alignment, flow, and assumption management

- **Waterfall:** RAID feeds stage-gate governance, formal risk registers, and phase-gate assumption validation
- **Agile:** lightweight RAID reviewed in stand-ups, sprint planning, refinement, reviews, and retrospectives
- **Hybrid:** RAID bridges formal governance and iterative delivery; dependencies and assumptions are key
- **Scaling:** SAFe program board dependencies, Scrum impediments, and Kanban blockers map to RAID
- **Pitfalls:** one-time documents, trivial-item overload, and RAID items never linked to decisions

RAID Reporting, Metrics, and Dashboards

RAID reporting answers four questions:

- What is at risk? What is blocked? What is assumed? What are we waiting on?

Key metrics to track:

- Open risks by severity, aging issues, unvalidated assumptions, overdue dependencies
- Escalation volume and time-to-resolution

Visual aids that work:

- Risk heat map, dependency map, issue burn-down, assumption tracker, RAG status

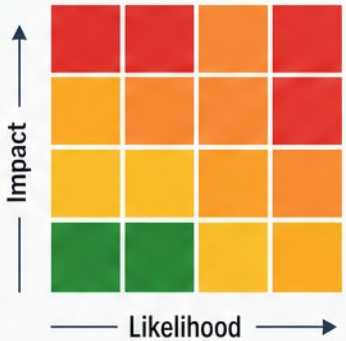
Tailor the view by audience:

- Delivery teams: line-item detail; PMO: cross-project patterns
- Executives: business impact and decisions required

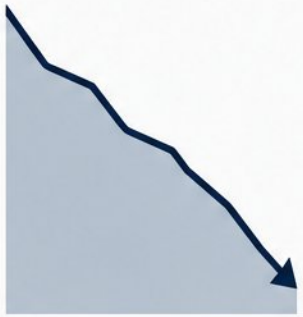
Use RAID to drive decisions:

- Go/no-go calls, contingency planning, scope trade-offs, steering escalations

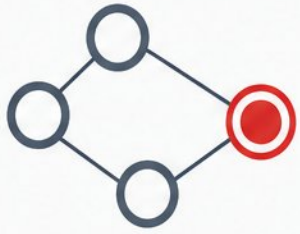
RISK HEAT MAP



ISSUE BURN-DOWN

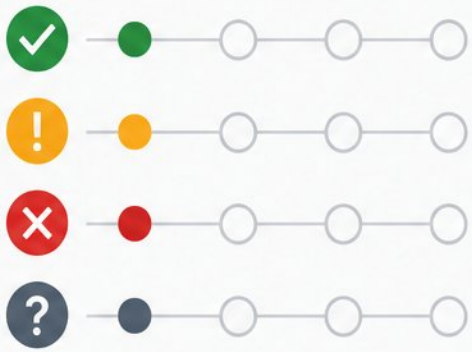


OVERDUE DEPENDENCIES



Overdue Dependency

ASSUMPTION VALIDATION TRACKER



RAG STATUS

Practical Case Scenarios and Classification Practice

Late Vendor Delivery

R → I

Risk → Issue

- Vendor dependency: delay risk becomes an issue; escalate and reassign

Failed Funding Assumption

R → I

Risk → Issue

- Funding assumption proves false: reduce scope, re-plan, escalate to sponsor

Regulatory Change

R → I

Risk → Issue

- Regulatory change: risk materializes into an issue; log owner and response

Cross-Team Dependency

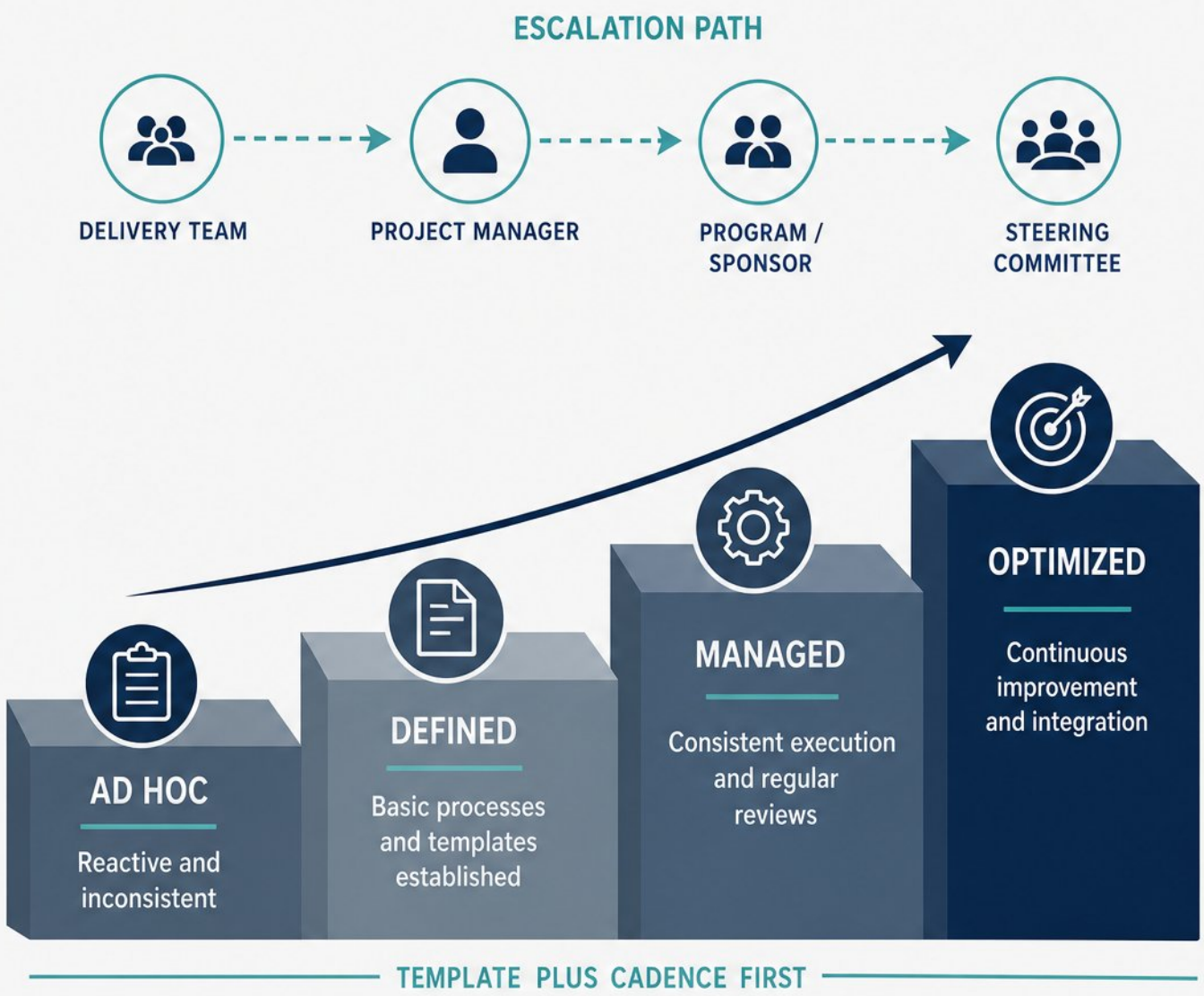
R → I

Risk → Issue

- Agile cross-team dependency: blocks sprints; RAID drives reprioritization

- **Workshop: classify sample items, propose next actions and owners**

Common Mistakes, Governance, and RAID Maturity



COMMON MISTAKES

- Confusing risks with issues; ignoring assumptions; unowned dependencies; stale logs
- Never mark everything High — if all items are red, nothing is ranked

GOVERNANCE

- Governance: log owner, review cadence, escalation ladder, PMO and audit links

MATURITY

- Maturity: ad hoc → defined → managed → optimized; template plus cadence first

QUALITY CHECK

- Quality check: clarity, single named owner, current status, dated action, archived closures

Knowledge Check and Key Takeaways

KNOWLEDGE CHECK

- Which RAID element relates to future events that may happen?
- Which RAID element relates to something happening now?
- Which RAID element is something we believe to be true but is not yet verified?
- Which RAID element relies on something outside our control?

KEY TAKEAWAYS

- RAID = Risks, Assumptions, Issues, Dependencies — each with distinct owner and approach
- A RAID log is a living governance tool: transparency, accountability, delivery success
- Apply RAID across agile, waterfall, and hybrid; adapt cadence to complexity
- Distinction test: risk = future, issue = present, assumption = unverified, dependency = external
- Next steps: review your RAID log, assign owners, schedule a review cadence

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/2ea195e1/public