



Suhas Bhairav

# AI Dependency Mapping Guide

**Identify and manage AI-driven dependencies to protect delivery timelines.**

Create actionable dependency maps to protect delivery timelines when adopting AI in projects.

**Suhas Bhairav**

[suhasbhairav.com](https://suhasbhairav.com)

# Dependency Mapping Foundations

This page defines dependency types and how to identify them in PM projects. It explains what to automate vs. what to review manually.

**Key move** Start with one project; the map should evolve weekly.

## PRACTICAL CHECKLIST

- 01** Map seven dependency types: task, team, vendor, system, data, approvals, timeline.
- 02** Capture owner, source tool, data fields, and update cadence for each dependency.
- 03** Automate data collection from Jira/Asana, calendar feeds, vendor portals, and data warehouses.
- 04** Identify critical path blockers and assign owners; distinguish delays caused by people, tools, or data quality.
- 05** Example: in a software release, map task dependencies, vendor SLAs, and data readiness.
- 06** Keep human review for subjective risk, strategic impact, and change approvals.

### Suhas Bhairav AI Resource

Create actionable dependency maps to protect delivery timelines when adopting AI in projects.

# Automation, Review, & Guardrails

Outline how AI agents can help maintain dependency maps and flag risks. Specify what to automate and what to preserve for human judgment.

**Key move** Never rely on AI alone for critical path decisions.

## PRACTICAL CHECKLIST

- 01** Automate: data extraction from PM tools, calendars, and vendor portals.
- 02** Automate: dependency graph updates every sprint, with versioning.
- 03** Human review: risk scoring, impact on milestones, and change-approval gates.
- 04** Data readiness: ensure source quality, timeliness, and access controls.
- 05** Tools: Claude/ChatGPT agents for mapping, Slack for alerts, Jira/Asana for data pull.
- 06** Guardrails: preserve human override, audit trails, and approval provenance.

### Suhas Bhairav AI Resource

Create actionable dependency maps to protect delivery timelines when adopting AI in projects.

# Execution, Ownership, & Metrics

Outline adoption steps across weeks, define ownership, and set measurable indicators. Emphasize governance and continuous improvement.

**Key move** Document ownership and revision history in the dependency atlas.

## PRACTICAL CHECKLIST

- 01** Week 1: inventory dependencies and assign owners; capture current data sources.
- 02** Week 2: establish lightweight AI-assisted maps and update cadence.
- 03** Week 3: run automated checks for data readiness and path risks.
- 04** Week 4: formalize governance, approvals, and ownership in project playbooks.
- 05** KPIs: on-time delivery rate, dependency slippage, AI coverage, and review cycle time.
- 06** Ownership: assign a PM, data steward, and AI platform owner.
- 07** Risks: data gaps, tool outages, misaligned approvals, over-reliance on AI.

### Suhas Bhairav AI Resource

Create actionable dependency maps to protect delivery timelines when adopting AI in projects.

## ABOUT THE CREATOR

# Suhas Bhairav

I build AI systems that combine distributed architecture, retrieval, knowledge graphs, security, and practical workflow design. My focus is AI that can be used repeatedly by real teams: understandable, reliable, and grounded in the work people already do.

## EXPERTISE

- Principal systems architecture and applied AI engineering
- Production-grade AI systems, agentic workflows, RAG, and knowledge graphs
- Full-stack workflow applications across AWS, Google Cloud, Azure, on-premise, and hybrid environments
- Practical AI implementation patterns for automotive, manufacturing, finance, logistics, cybersecurity, sales, and customer operations
- Security, governance, observability, approvals, and failure-mode thinking for operational AI systems

### WEBSITE

**[suhasbhairav.com](https://suhasbhairav.com)**

### EMAIL

**[suhasbhairav@gmail.com](mailto:suhasbhairav@gmail.com)**