# Leading Agile in a Non-Agile Infrastructure Environment: Strategies for Project Managers

**Published on 24 April 2025 at 10:19**

**Author: Kimberly Wiethoff**

Agile has become the gold standard in software development—but what happens when your project includes infrastructure teams that still operate in a traditional, waterfall-style framework? Welcome to the **hybrid reality** many IT project managers face today.

Successfully leading Agile initiatives when infrastructure isn’t fully Agile requires a blend of adaptability, empathy, and strategic planning. Here’s how project managers can bridge the gap and drive progress—even when everyone isn’t sprinting at the same pace.

## 🛠️ The Reality: Agile Development, Traditional IT

Infrastructure teams are often tied to change control boards, scheduled maintenance windows, procurement cycles, and strict compliance mandates. Agile, on the other hand, is all about iterative delivery, fast feedback, and continuous improvement.

This can lead to challenges like:

* Dev teams waiting on servers or firewall rules
* Mismatched timelines and dependencies
* Delays in CI/CD setup or environment readiness
* Friction over documentation and change approval processes

So how do you keep your Agile project moving when part of your team is running on rails?

## 🔑 5 Practical Strategies to Lead Agile in a Non-Agile Environment

**1. Map Dependencies Early**

Use a dependency mapping workshop during sprint zero or project kickoff. Identify every infrastructure deliverable tied to development milestones. Include firewall changes, DNS entries, backup setups, etc.

**2. Create an Agile-Compatible Infrastructure Backlog**

Even if infra doesn’t use Scrum, work with them to develop a backlog-like task list that aligns with your sprint cadence. Add infrastructure tasks into Jira or Confluence for visibility.

**3. Build in Buffer and Lead Time**

Unlike Agile teams that can pivot quickly, infrastructure work often involves lead times (e.g., vendor provisioning or internal approvals). Pad your roadmap and communicate dependencies clearly.

**4. Educate, Don’t Convert**

Don’t force Agile jargon on teams unfamiliar with it. Instead, focus on outcomes: collaboration, transparency, and predictable delivery. Help infrastructure leads see the value without pushing for a full Agile transformation.

**5. Create Cross-Functional Sync Cadence**

Weekly syncs between dev and infra teams help surface blockers early. These don’t need to be formal stand-ups—just enough to align timelines and flag issues before they delay sprints.

## ✅ Key Takeaway: Deliver Agility, Not Just Agile

True agility is about **responding to change and delivering value consistently**, not forcing everyone to adopt the same framework. As a project manager, your job is to create flow between teams—even when their operating styles differ.

By meeting infrastructure teams where they are, while bringing Agile structure and transparency to the process, you can **lead hybrid projects to success** without bottlenecks or burnout.

**#AgileLeadership #ITProjectManagement #HybridDelivery #InfrastructurePM #DevOps #ScrumMasterTips #AgileTransformation #WaterfallToAgile #DigitalDelivery #CrossFunctionalTeams**

## ✅ Checklist for Agile PMs Working with Traditional Infrastructure Teams

Use this checklist to help manage Agile software development projects that depend on infrastructure teams still operating in traditional or hybrid models.

🗺️ Map all infrastructure dependencies at project kickoff (e.g., firewalls, DNS, backup).

📋 Create a shared backlog or task list with infrastructure tasks aligned to sprints.

⏳ Build in buffer time for infrastructure tasks that require lead time (e.g., procurement, change control).

📣 Communicate infrastructure dependencies clearly in sprint planning sessions.

🤝 Schedule regular sync meetings between infrastructure and Agile teams to identify blockers.

🎓 Educate infrastructure teams on Agile concepts without pushing a full framework change.

🔍 Ensure visibility of infrastructure tasks in project tracking tools like Jira or Confluence.

📅 Coordinate change windows and maintenance schedules with sprint cycles.

🧩 Foster collaboration by aligning goals and delivery milestones across both teams.

📈 Track delivery risks and escalate early when cross-team alignment issues arise.