Leading Agile in a Non-Agile Infrastructure Environment

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

We'll examine practical strategies for navigating the hybrid reality many IT project managers face today, where development teams embrace Agile methodologies while infrastructure teams operate in traditional frameworks.

kW by Kimberly Wiethoff, MBA, PMP, PMI-ACP

Managing Projects The Agile Way

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

The Hybrid Reality Challenge

Agile Development Teams

Focused on iterative delivery, fast feedback loops, and continuous improvement. Work in sprints with flexible scope and regular delivery cadence.

Traditional IT Infrastructure

Tied to change control boards, scheduled maintenance windows, procurement cycles, and strict compliance mandates requiring upfront planning.

The Project Manager's Dilemma

Must coordinate teams operating at different speeds with different methodologies while maintaining project momentum and stakeholder satisfaction.

This fundamental mismatch in operating styles creates friction that can derail projects if not actively managed. The challenge lies in creating harmony between these different approaches while ensuring neither team feels their process is being compromised.

Common Friction Points

Mismatched Timelines

Development teams plan in 2-week sprints while infrastructure changes might require 30-day approval cycles, creating scheduling conflicts.

Environment Bottlenecks

Agile teams ready to deploy but waiting on server provisioning, firewall rules, or networking configurations from infrastructure teams.

Documentation Requirements

Infrastructure teams often need detailed specifications upfront, while Agile teams prefer to evolve documentation as the solution develops.

Dependency Management

Infrastructure work often requires sequential steps that can't be parallelized, creating critical path dependencies for Agile deliverables.

Strategy 1: Map Dependencies Early

Conduct Dependency Workshop

During sprint zero or project kickoff, gather all teams to identify infrastructure dependencies that will impact development timelines.

Create Comprehensive Inventory

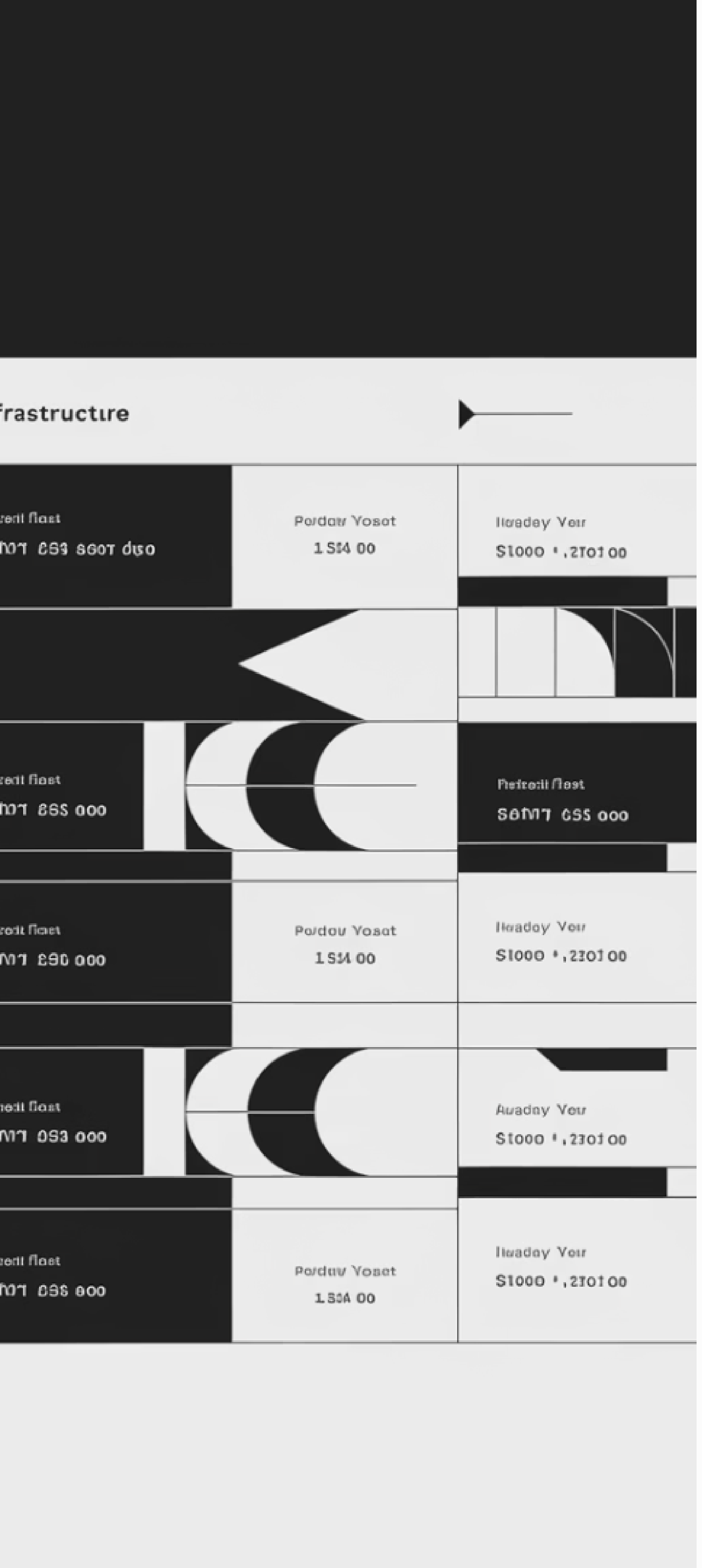
Document every infrastructure deliverable tied to development milestones, including firewall changes, DNS entries, backup setups, and hardware requirements.

Visualize Dependencies

Develop a shared visualization showing how infrastructure work connects to development tasks, with clear timelines and ownership.

Establish Regular Reviews

Schedule bi-weekly dependency reviews to ensure alignment as requirements evolve throughout the project lifecycle.



Strategy 2: Create an Infrastructure Backlog

Translate Requirements

Work with infrastructure teams to convert their traditional task lists into backlog-style items that align with sprint planning.

Prioritize Infrastructure Work

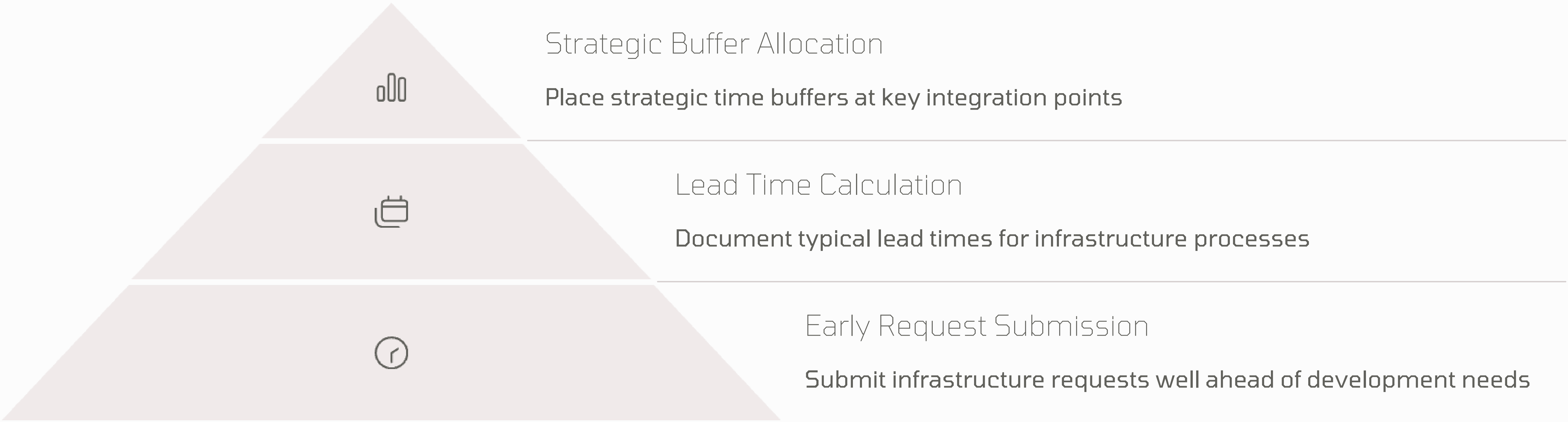
Apply MoSCoW prioritization (Must have, Should have, Could have, Won't have) to infrastructure tasks based on development dependencies.

Create Shared Visibility

Add infrastructure tasks into Jira or Confluence alongside development stories for complete project transparency.

Even if infrastructure teams don't fully adopt Agile methods, creating a backlog-like structure helps both teams visualize work in progress and identify potential bottlenecks before they impact delivery. This approach respects traditional processes while improving coordination.

Strategy 3: Build in Buffer and Lead Time



Strategic Buffer Allocation

Place strategic time buffers at key integration points

Lead Time Calculation

Document typical lead times for infrastructure processes

Early Request Submission

Submit infrastructure requests well ahead of development needs

Unlike Agile teams that can pivot quickly, infrastructure work often involves significant lead times for vendor provisioning, security reviews, or internal approvals. Smart project managers pad their roadmaps with realistic buffers based on historical infrastructure timelines.

Develop a reference guide documenting the typical lead time for common infrastructure requests. This helps development teams understand when to initiate requests and sets appropriate expectations with stakeholders about delivery timelines.

Strategy 4: Educate, Don't Convert

Focus on Outcomes, Not Terminology

Avoid forcing Agile jargon on infrastructure teams. Instead, emphasize shared goals like improved collaboration, transparency, and predictable delivery.

Infrastructure teams often have valid reasons for their processes, many tied to regulatory compliance or operational stability requirements that Agile methods don't always address.

Practical Education Approaches

- Share success stories from similar hybrid environments
- Invite infrastructure teams to sprint reviews to see value delivery
- Demonstrate how small process improvements can benefit them
- Speak their language when discussing project needs

Respect that infrastructure teams may never fully adopt Agile, and that's okay. Your goal is creating enough mutual understanding to enable smooth collaboration, not forcing a complete transformation.

Strategy 5: Cross-Functional Sync Cadence

Weekly Syncs

Schedule brief weekly touchpoints between dev and infrastructure leads to align priorities and identify blockers

Dependency Check-ins

Review critical dependencies and adjust timelines based on current progress

Risk Assessment

Collaboratively identify emerging risks that could impact either team's delivery

Commitment Alignment

Confirm key commitments for the next period and document any changes needed

These cross-functional sync meetings don't need to be formal stand-ups—they should be efficient check-ins focused on coordination between teams with different work cadences. The key is creating a regular forum where potential issues can be surfaced before they delay sprints.

Communication Techniques That Bridge the Gap

Language Translation

Serve as a "translator" between Agile and traditional terminology. Help each team understand the other's priorities and constraints without requiring them to learn new frameworks.

Visual Management

Create visual dashboards that represent work in both Agile and traditional formats, providing a unified view that respects both approaches while highlighting dependencies.

Documentation Balance

Find the middle ground between infrastructure's need for detailed documentation and Agile's preference for just-in-time information by creating flexible, evolving specification templates.

Effective communication is perhaps the most critical skill for project managers working in hybrid environments. By adapting your communication style to bridge different working approaches, you create understanding without forcing either team to abandon their effective processes.

Technical Solutions for Hybrid Environments

Infrastructure as Code

Automate infrastructure provisioning where possible

Environment Templates

Create standardized, repeatable environment configurations

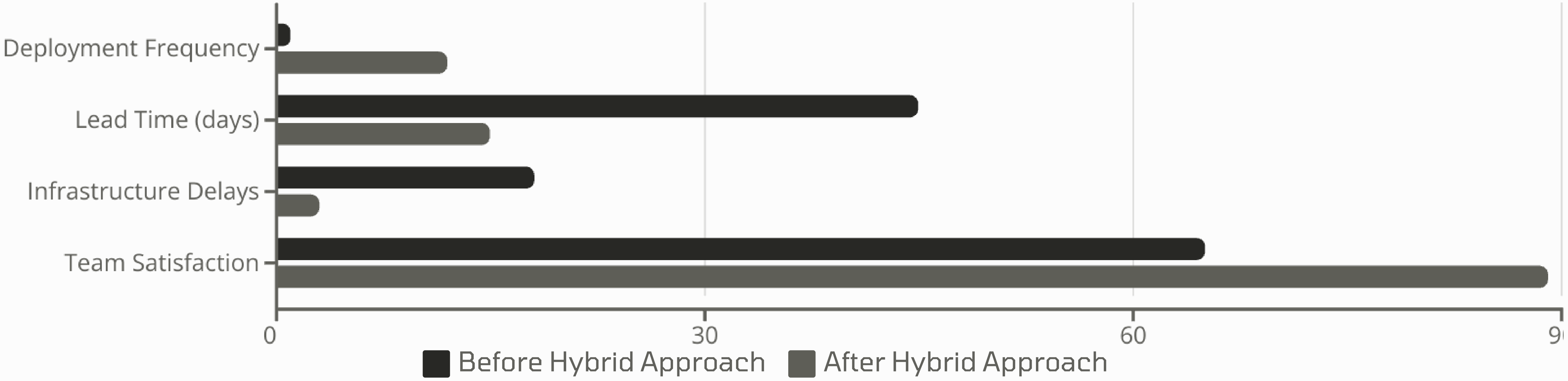
Self-Service Capabilities

Develop portals for dev teams to request infrastructure

Technical approaches can help bridge the gap between Agile and infrastructure teams. Infrastructure as Code (IaC) brings automation to traditionally manual processes, allowing infrastructure to be provisioned and configured through code rather than manual intervention.

Self-service portals can empower development teams to request infrastructure resources through standardized forms that satisfy infrastructure governance while reducing manual handoffs. These technical solutions reduce friction without requiring infrastructure teams to abandon their control mechanisms.

Case Study: Financial Services Success



A major financial services firm implemented these hybrid Agile strategies when modernizing their loan processing system. Their development team was fully Agile, while infrastructure operated under strict compliance controls in a traditional framework.

By implementing cross-functional planning, visualization of dependencies, and strategic buffering of infrastructure lead times, they achieved dramatically improved results. Both teams reported higher satisfaction as friction decreased and delivery became more predictable.

Key Takeaways: Deliver Agility, Not Just Agile

Value Outcomes Over Methodology

Focus on creating consistent value delivery and responsiveness to change, not on having everyone follow the same framework. True agility comes from adaptability, not rigid adherence to any single approach.

Create Flow Between Different Systems

As a project manager, your role is to create harmony between teams with different operating styles. Build bridges that allow work to flow smoothly while respecting each team's constraints and strengths.

Iteratively Improve the Hybrid Model

Apply the core Agile principle of continuous improvement to your hybrid approach itself. Regularly retrospect on what's working and what's not in your cross-team coordination, adapting as you learn.

Remember that successful hybrid project management isn't about forcing infrastructure to become Agile or slowing development down to match infrastructure pace. It's about creating an environment where both teams can work in their optimal way while delivering integrated value to the organization.