

Top 10 PMO Worst Practices: Pitfalls to Avoid

Gartner Research estimates¹ that Project Management Offices (PMOs) have a failure rate of 50 percent or more on their first try. Daptiv has engaged with hundreds of enterprise customers to help ensure they are successful with their PMO organizations from the outset and avoid common pitfalls that lead to failure. Based on our experience, we've compiled the top ten PMO worst practices that we've seen and advice on how to avoid these.

“Without tracking actual time worked on a projects and other work, it is impossible to know any department’s true capacity.”

1. The PMO playing methodology cop.

If there is a common refrain to failed PMOs, this is it: The PMO becomes the “Methodology Police” tasked to enforce methodologies – often ill-fitted ones – rigorously. At best, this leads to complaints from project managers; at worst it leads to open revolt and ignoring the methodology entirely. Rather than becoming an aid and guide to project managers that creates consistency in PM practices, this PMO has become the PM’s worst enemy. The PMO loses visibility into the project portfolio, and often fails due to lack of support.

2. Implementing a methodology without a framework.

We have seen many PMOs try to implement a Software Development Life Cycle (SDLC) across their organization. This is great for some software development teams, but can be completely inappropriate for infrastructure and process improvement teams. A better structure is an overall project management framework, with just the basic phases and gates and a few key controlling artifacts (business case, project schedule, status report, etc.). This is sometimes known as a PDLC (Project Development Life Cycle), and many different SDLCs can fit under the framework, tailored to the needs of the project type.

3. Not implementing a methodology.

Are we contradicting ourselves? Not really. If the PMO does not implement some kind of overarching framework, it cannot create an apples-to-apples view of the projects in the portfolio. However, a methodology is required to ensure consistency of project execution and to reduce the risk stemming from inexperienced project managers. For instance, a rookie PM might not do the proper project planning, assignment of work, or risk assessments leading to problems down the road and often heroic efforts to get the project back on track. So, an established methodology is essential – or perhaps several under an over-arching framework. Finally, federal regulations, especially Sarbanes-Oxley, require methodologies that meet certain process and change management requirements.

¹ Gartner ITxpo, 2010.

“Automating bad or non-existent processes simply makes problems come at you faster.”

4. Not matching demand to supply.

In steering committee meetings, the focus is almost always on prioritization. That’s a good thing, since good prioritization processes result in a good understanding of project demand. But when it comes to deciding how many of the top projects can actually be done, discussion too often turns to pure guess work, e.g. “How overloaded is your department, Tom?” or “If we take on the top 10, can we handle the load?” Without a metrics-based understanding of resource capacity, it is impossible to match that wonderfully organized demand with the actual supply of human resources.

5. Not logging time.

This is actually the source of worst practice #4. Without tracking actual time worked on actual projects and other work, it is impossible to know any department’s true capacity. Planning, even at the most detailed level, is merely guesswork if it does not involve the feedback loop of actual time.

6. Gathering unnecessary information.

PMOs are great at gathering all sorts of statistics. If that information is not used in the decision-making process, why gather it? It just creates an extra burden on the project team without generating any useful result. An example is gathering time data at the detail task level, including sub-day tasks like “Enter time,” “Respond to email,” and “Attend staff meeting.” These may be useful reminders as tasks, but for planning purposes all we really need to know is how much time each resource is spending performing “Admin” tasks.

7. Keeping an ad-hoc project request process.

The most common process for dealing with new project requests is to analyze them one by one and decide if each merits becoming a new project. But what about the other 100 requests? Where’s the prioritization? Upgrading the OS on the servers may be important, but is it more important than the new data center in EMEA? If you evaluate requests in isolation, then the next “important” project trumps an already in-flight project, until it is trumped by the next “emergency” project. The result is serious project churn. The solution is cyclical request cycles (at least quarterly) that consider all requests side-by-side and select those most important to the company. How many should pass the grade? See number 4: matching demand to supply.

8. Lack of executive support.

This one should be obvious, but it happens all the time. The executive team realizes there’s a problem with projects not executing properly, contending for too few resources, or just not delivering results. So they authorize a PMO, and hope it solves the problem. But come time to attend steering committee meetings they send lower-level functionaries and don’t give them decision-making authority. Or, even worse, they over-ride PMO decisions. Then, when projects are once again performing poorly due to resource contention and prioritization conflicts, they blame the PMO and disband it.

9. Implementing a tool without a process.

Another obvious, but often found, problem. Whether you are implementing a PPM solution or an ERP system, automating bad or non-existent processes simply makes your problems come at you faster! IT departments are particularly prone to thinking the tool is the whole solution, but it's surprising how many other organizations fall into this same trap. The usual excuse is that the tool already has best practices embedded, so why not just follow those? In our experience, every organization is different and requires their own tweak on "best practices."

10. Implementing a process without a tool.

Standard advice seems to be to get your process act together before automating with a tool. We've seen many PMOs who have followed this advice and run into problems. Complex processes run on spreadsheets and documents are very burdensome, and that extra burden can sabotage adoption. We've found it's far better to implement the process and tool at the same time. There are a few advantages. First, you can design the new process to take advantage of the chosen software tool's strengths. You would have to tweak them during an implementation anyway, so you might as well do it while you design the process. Second, users learn the new process in the new tool which means one learning curve instead of two. Finally, your new process has the best chance of adoption by streamlining and automating from the start. There is one serious downfall to beware of though – when something goes wrong, you really need to diagnose if it is a process or tool issue. Software is a great inanimate scapegoat, but often not the actual problem.

Tweet about this paper using the hashtag #daptiv