

Technical Program Manager (TPM) Ladder Guide: A Cover Letter

Author: *Straker Carryer*

Executive Summary

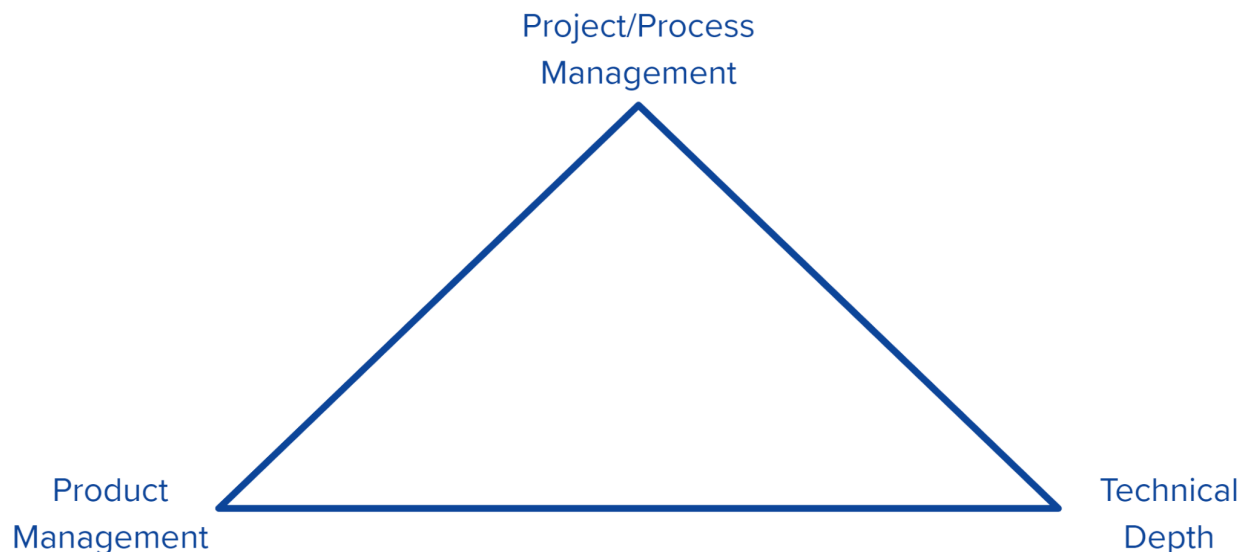
This cover letter introduces a ladder guide for the Technical Program Manager (TPM) role.

This guide follows [the open format laid out in the series on strakercarryer.com](https://strakercarryer.com). This cover letter explains the role at a high level and is meant to be used for recruiting and understanding of the role by others not in the role. It is paired with [a matrix](#) for clarity of in-role expectations and promotions.

Role Description

The TPM role is a unique one, with lots of different uses in the industry across different companies. However, it can easily be summarized as a “jack or jill of all trades.” They are interdisciplinary by nature, creating a one-stop shop for cross-functional and cross-organizational projects and programs. As part of their interdisciplinary background, they have the technical knowledge to participate directly in architectural conversations with engineering teams.

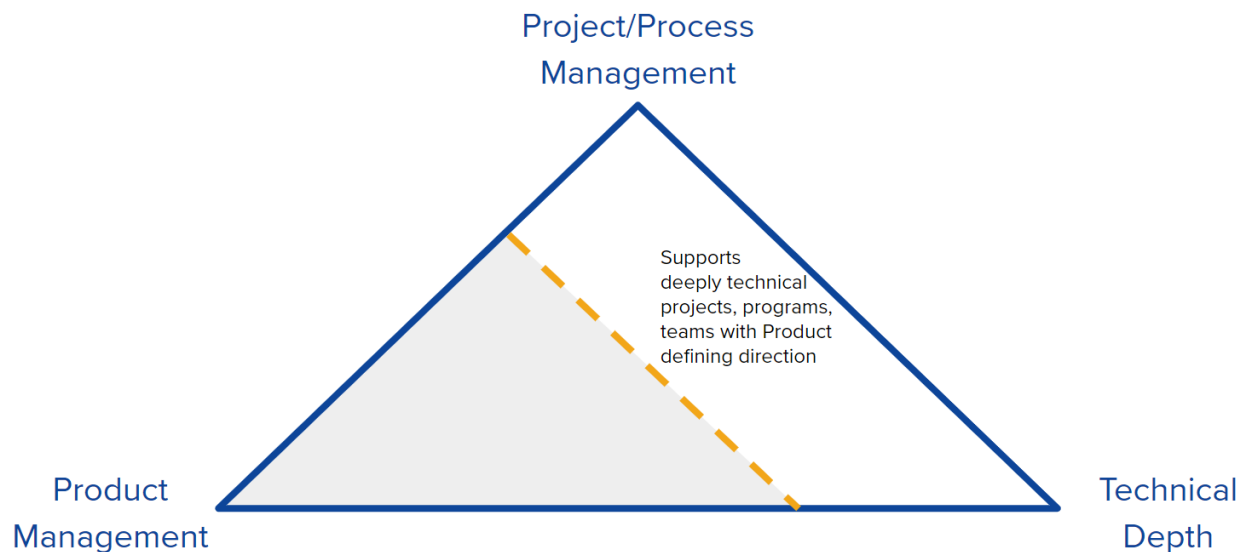
Given the broad range of skills necessary to cover a TPM’s scope, it is unreasonable to expect them to demonstrate all skills at all times. Rather than a laundry list of skills, the role is best defined via the following infographic, “the TPM triangle.”



If we break down the TPM skillset into the above vertices, a TPM must be able to do all three, but rarely all at once. Rather, they are expected to demonstrate two of these skill sets simultaneously. However, they must be capable across all three, as they must be flexible to change which two they are focusing on as work progresses from strategy through execution.

Depending on an organization's need, it is fair to have multiple TPMs in the same area to ensure proper coverage. Therefore, TPMs are also expected to work together as needed. However, if they do so by splitting up the vertices in the TPM triangle, there are likely more TPMs in the space than are necessary and they will no longer be meeting the expectations of this guide.

In the same manner, the TPM triangle explains how TPMs are different from being a Product Manager in isolation. TPMs are expected to work with other Product and Program Managers, understanding that some skills will overlap. Depending on the group working together, you may see a TPM flex from one area of the triangle to another to ensure a project or program is being served as needed. For example, if a TPM is paired directly with a PM, you might see the TPM shift more to the “technical executor” archetype on the right side of the triangle to complement and balance the PM:



For more details, see [the introduction to the role and the TPM triangle on strakercarryer.com](https://strakercarryer.com).

Career Progression Summary

This guide recommends 5 levels for the TPM role, mostly [following Google's naming convention of title adjectives](#):

1. TPM
2. Senior TPM

3. Staff TPM
4. Senior Staff TPM
5. Distinguished TPM

While additional levels can be added to support Associate or Fellow TPMs to add levels earlier or later in a TPM's career, respectively, this is only necessary for a small percent of larger companies that tend to already have ladder guides for their roles defined. Note that this guide follows Senior Staff with "Distinguished" instead of "Principal" like Google given the large number of companies that use Principal instead of Staff (e.g. Amazon).

The percent of TPMs at a particular level is expected to be significantly less than the level before it.

TPMs are expected to reach the Senior level, but advancement beyond that level is not required; some companies call this a "terminal level."

Unlike engineers which have [a significant shift in the job at the Staff level](#), TPMs must demonstrate leadership from the beginning of their career. Instead, the main factors that change across levels are the scope of their work, as evidenced by their key stakeholders and the level of ambiguity with which they operate. At the start of their career, a TPM's scope is likely a single engineering team or small set of teams that report to the same manager (usually director level or below). However, this scope increases quickly, moving away from a count of engineering teams and instead focusing on an organization or line of business.

See [the Impact Range of the paired matrix](#) for more details:

	Impact Range return to worksheet				
	L2 - Mid-Level	L3 - Senior	L4 - Staff	L5 - Sr. Staff	L6 - Distinguished
Direction	Receives frequent direction. Sometimes gives cross-functional direction within their scope.	Often self-directed. Often gives cross-functional direction within their scope.	Mostly self-directed. Often gives cross-functional direction within their scope.	Self-directed. Gives cross-functional direction across an entire line of business, potentially spanning multiple organizations.	Self-directed. Gives cross-functional direction across the entire company.

Scope Approximate; will vary based on Program requirements	Typically directly supports a small number of teams (1-3) within a single line of business. Might drive projects/programs of a broader scope than their directly supported teams require, but these are usually standardized across a larger number of teams.	Typically directly supports an organization (director level) and/or 1-2 programs or large projects required by that organization. Often drives projects/programs of a broader scope that their directly supported organization requires.	Typically manages programs and/or large projects required by an organization (VP level) rather than directly supporting teams.	Typically manages programs and/or large projects required by a line of business (SVP level) rather than directly supporting teams.	Typically manages programs and/or large projects required by the company (C-suite level) rather than directly supporting teams.
Context Based on the “Problem, Solution, How, Execution” framework	Is usually assigned projects/programs where goals are clear. Solves problems and executes solutions using direct precedents.	May be assigned projects/programs where scope and goals are less clear. Proactively identifies some problems. Solves problems and executes solutions using some precedents.	Often proactively identifies problems. If assigned a project/program, high levels of ambiguity are expected. Solves problems and executes solutions using limited precedents.	Usually proactively identifies problems. If assigned a project/program, high levels of ambiguity are expected. Solves problems and executes solutions without precedents.	Proactively identifies problems. If assigned a project/program, the highest possible levels of ambiguity are expected. Solves problems and executes solutions without precedents.