Defining the Technical Delivery Manager role

Practitioners of organisation design are often divided by a common language. In my experience, the role of Technical Delivery Manager (TDM) is one of the most used and least commonly agreed.

By **Avi Sinharay**, M:nd Rocket Serv:ces

The Technical Delivery Manager (TDM) is responsible for overseeing the successful delivery of technical projects and ensuring they meet agreed-upon scope, timelines, and quality standards. This role requires a blend of technical knowledge, project management skills, and strong communication abilities. This blend varies depending on the company's context, which is why few role definitions agree.

In this technical paper I attempt to describe the TDM role as would apply to a medium-sized company that delivers a lot of technology change.

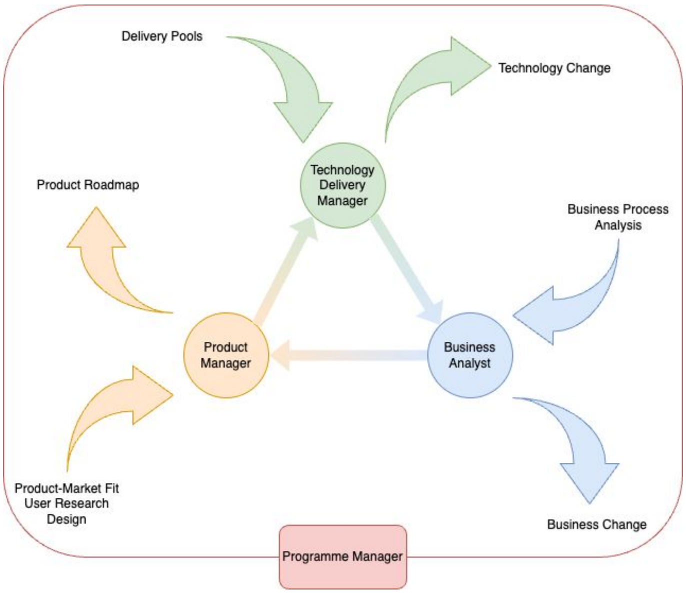
Role Context

The TDM works in concert with the Product Manager (PdM) and Business Analyst (BA) to successfully enact change in products and the wider business. Each role’s focus is similar but different.

Note that delivering technology change is one of many streams of work within a project and so within the purview of the Programme Manager.

Delivery roles

Technical Delivery Manager	Assures delivery of technology change by assuring work intake, coordinating technology development, assuring hand-off to project or business.
Product Manager	Defines and prioritises the work to be done by development teams. Assures that the work done meets the expectations set in the Product Roadmap.
Business Analyst	Defines the process change work to be done by the business in order to adopt the technology change. Assures the Business Change work. Will be coupled with a Project Manager to coordinate this business change work.



Responsibilities

The TDM’s responsibilities boil down to assuring 3 things:

1. Work Intake
2. Development execution
3. Work Output

1. Work Intake

In order for the work to be ready to be developed, it needs to be properly defined and prioritised. For customer-facing projects, this definition and prioritisation is done by the PdM. For other internally-facing projects, this definition and prioritisation is done by the BA. In either case, it is their responsibility to ensure that the right work is done at the right time, and to communicate this out to the rest of the business.

The work attempted is necessarily complex as demand is very reactive to market and business context. Furthermore most companies prefer to be unstructured in communication channels and ways of working in order to be more agile, so effective collaboration takes consistently high effort. The TDM takes on the collaboration burden of making sure this work is done to the quality that enables development to proceed through:

- Creating and assuring agreement for a **Quarterly Plan** by managing the planning process (following the PI Planning process as defined in SAFe) - but the project and business inputs are collated and prioritised by the PdM (or equivalent).
- Creating and assuring the **Sprint Plans** (and Release Plans) by ensuring there are good quality Jira tickets for each piece of work planned - but the PdM and the devs collaborate on writing the tickets.

2. Development

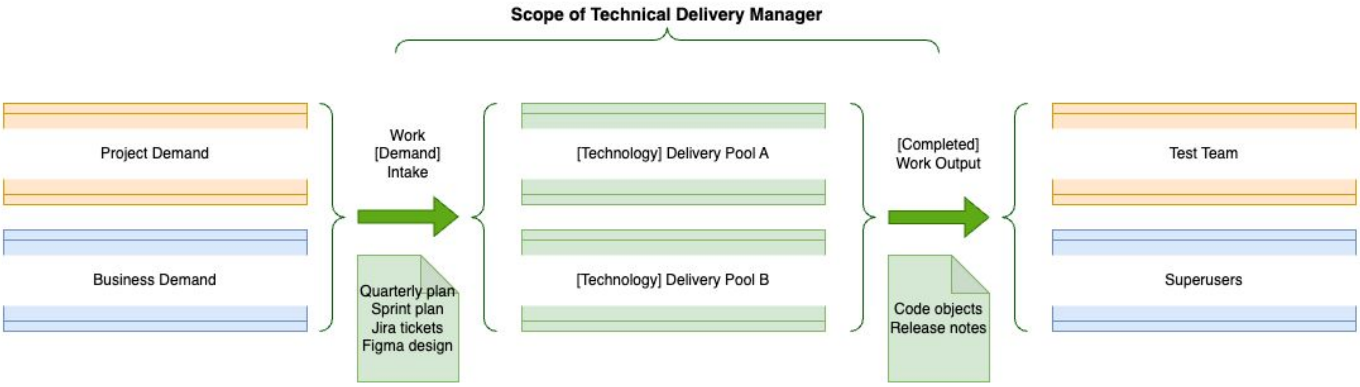
During the course of technology development, the TDM acts as an Agile scrum master facilitating the ceremonies of: daily stand-up, sprint planning, ticket refinement, sprint demo and retrospectives.

Technology development is generally one of many workstreams so the TDM will also represent all Technology development teams in the scrum of scrums.

3. Work Output

Once completed, technology change is handed over to the Test Team to conduct Acceptance Testing. Whilst this team is in a different reporting line, the TDM actively manages their work through:

- Ensuring that test **protocols** are written - the content is written by the Test Team in collaboration with developers;
- **Schedule** the Test Team's work to ensure timely completion of Acceptance Testing - without any view of their other responsibilities.



Conclusion: Managing the Dev Team API

Having a well-defined, well-understood input and output is analogous to an application programming interface. The TDM can apply the same principles of API design to managing the development team's hand-offs:

- **Clarity and simplicity**: the clearer and simpler the work, the less the collaboration overhead in extracting the design intent.
- **Consistency**: the dev team's output should be predictable, so that they can be confident in making delivery promises, and the business can rely on the delivery promises.
- **Robustness**: business needs change so the team should be able to react without being reactionary.
- **Performance**: there will always be pressure to go faster, but the TDM should make sure that this pressure doesn't reach the developers; instead the TDM should create the conditions for the team to perform as long as it takes without burnout.
- **Evolvable design**: continuous improvement will yield amazing results over time with the right focus, so the TDM should measure what really matters to the business and foster the safe environment needed to challenge each other constructively.

My favourite analogy for the role of TDM is as a gardener: they are responsible for making sure only plants get in and no weeds, they tend the garden diligently removing obstacles to growth, and they deliver the fruits to the rest of the business.