Linear

# **Priority micro-adjust PRD**

# Context

## Customer Messaging

Make fine-grained adjustments to priority levels within basic categories. If you implement a stack-ranking system for your tasks and projects, this is specifically made for you!

## Problem

A very common use case we encounter is global stack-ranking: customers want to form a stable, opinionated order of priority across all of their backlog items. Today, we provide a global manual ordering feature, which we recommend for these cases.

There are several key issues with the current manual ordering system:

1. **Instability.** Manual sort is often unstable. Users often do not treat this ordering with care because it feels so low-stakes. They often make changes without realizing someone is relying on the ordering.
2. **Local vs. Global Sorting.** In other apps, manual sort is often local. Users are often surprised we have global manual ordering.
3. **Redundancy with Priority Field.** We currently have both manual sort and a priority field. If the goal is to create a global priority stack rank, it's awkward to have a priority field that doesn't agree with your sort.

When users express these problems, it's often through indirect feature requests.

1. We often get the request for "custom priority levels" because the 4 basic buckets don't offer enough granularity
2. Or we get the request for custom fields, for the purpose of adding a "ranking" column, that's just a numerical stack-rank

## Solution

Priority levels are a natural fit for this, but today they are just broad buckets (low, med, high, urgent) and don't offer enough control for customers that want to literally rank every single item.

We're simply going to allow users to adjust and rearrange the relative priority of their issues ***within*** a priority bucket, through drag-and-drop.

Once we have this, we will strongly recommend to users that all stack-ranking use-cases be achieved through priority + micro-adjust.

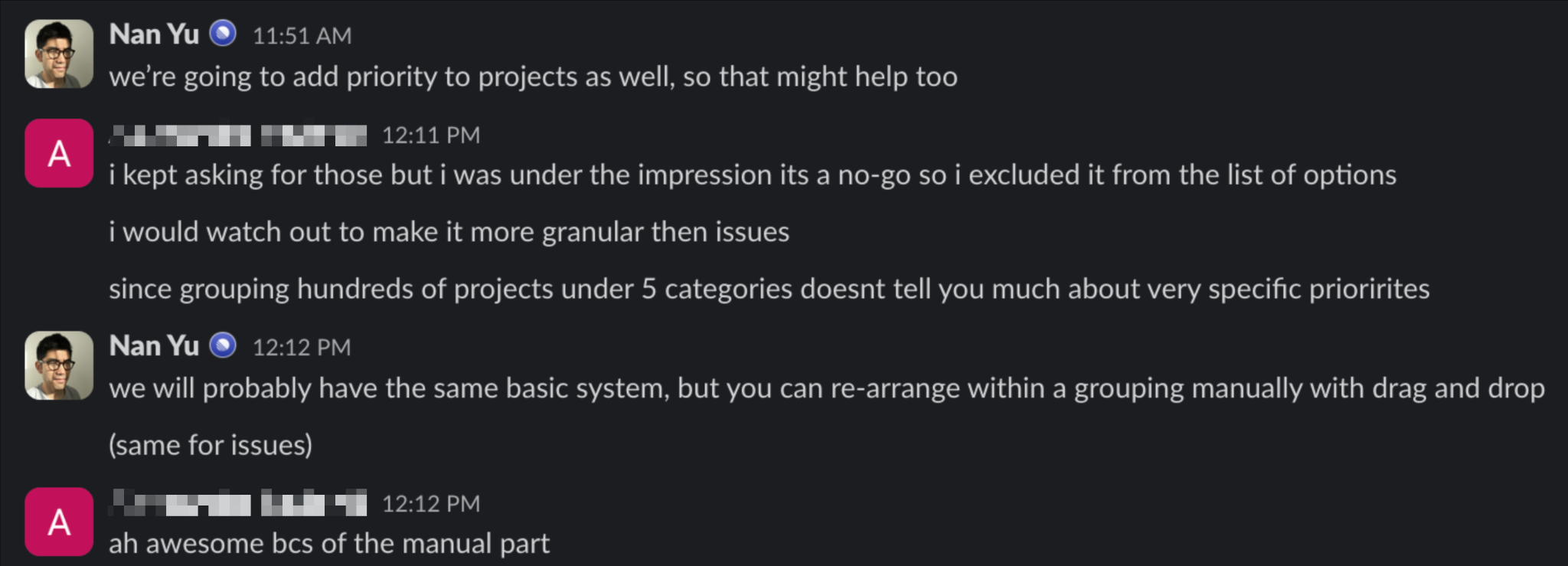
### Options we considered but decided against

1. **Custom priority levels.** Doesn't really solve for stack ranking, and would encourage stack-rankers to make way too many distinct categories
2. **Custom fields.** Would solve for stack-ranking, and many other asks, but it’s out of character for Linear. Our ease-of-use comes from having opinionated defaults that reliably work the same everywhere

### Prior art

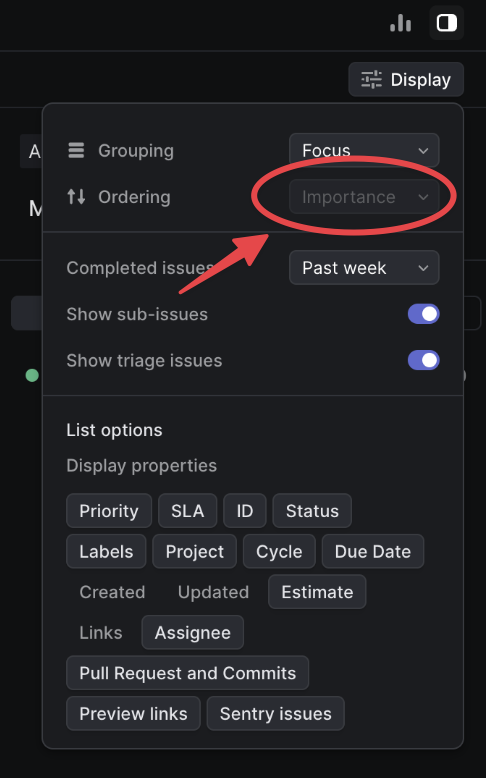
We know of no other tool that offers manual adjustment within priorities; they all take one of the options we decided against, so this is an opportunity to differentiate. We offer a more compelling story by integrating priority and stack-ranking together into a single system.

# Usage scenarios

**User A asks for priority on projects (which we are planning to ship) and more priority categories**

**With priority micro-adjust**: when User A wants to organize according to very specific priorities, she can simply drag issues and projects around to the order she wants, within the same priority level. This lets her avoid the complexity of defining new priority levels and educating her team on what those levels mean exactly. She’s just able to say “this issue is higher priority than the other high-priority issues”

**User B proposes a mockup of issues ordered by a custom field they call "importance", which they proposed to be an integer value that can be defined on each issue**

****

**With priority micro-adjust**: when User B wants to be specific about the relative importance of an issue, he can just re-arrange the order visually. This is less mentally taxing than trying to think of an arbitrary integer or having to explain what such a number means to the members of his team.

# Milestones

## MS1: Internal

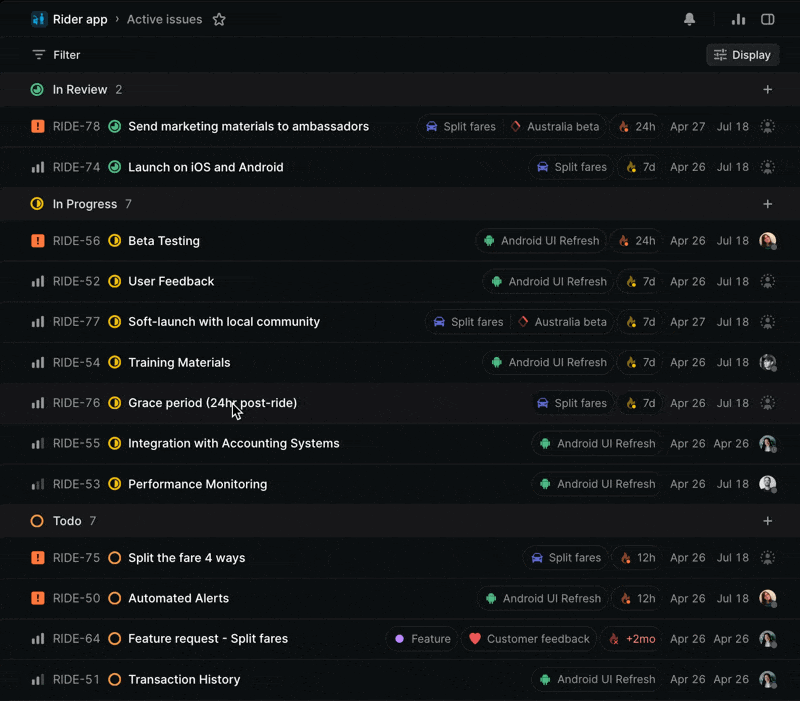
### Specs

When ordering by priority, allow drag-and-drop within a priority to determine fine-grained ordering

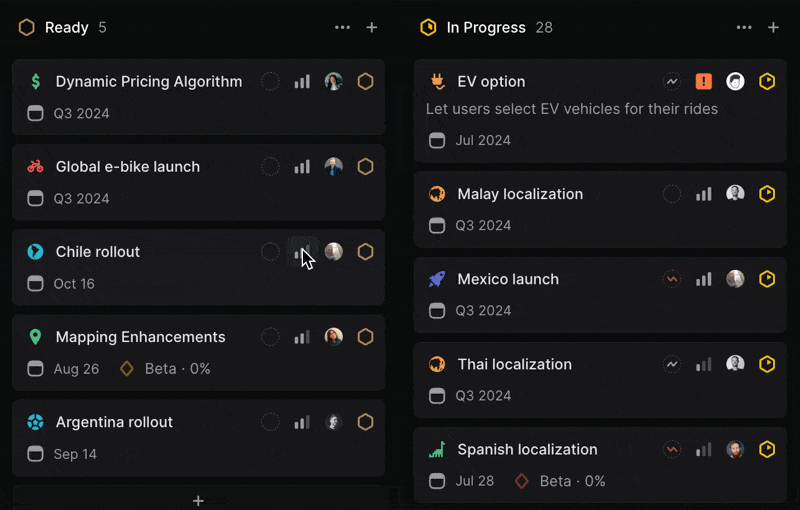
* When dropping an item into a set of items with a different priority, change its priority to match
* When dropping in between 2 sets of priorities
  + If the movement was upwards, take the lower priority
  + If the movement was downwards, take the higher priority

### Mockups & prototypes

**Adjust issues within and between priorities**

****

**Changing priority buckets through selection and drag-and-drop**

****

## MS2: Beta

### Change management

**Initialize using the manual sort index.** When releasing this to actual users, we need to bootstrap their initial microadjust ordering. We can use the existing manual sort index as a starting point because this has been our recommended solution for stack-ranking issues.

## MS3: GA + Changelog

### Announce alongside project priorities

We will release **priority micro-adjust** and **project-level priorities** in the same changelog to tell a more cohesive story about the priority system within Linear.

[Resulting changelog](https://linear.app/changelog/2024-07-25-priority-for-projects-and-micro-adjust)

