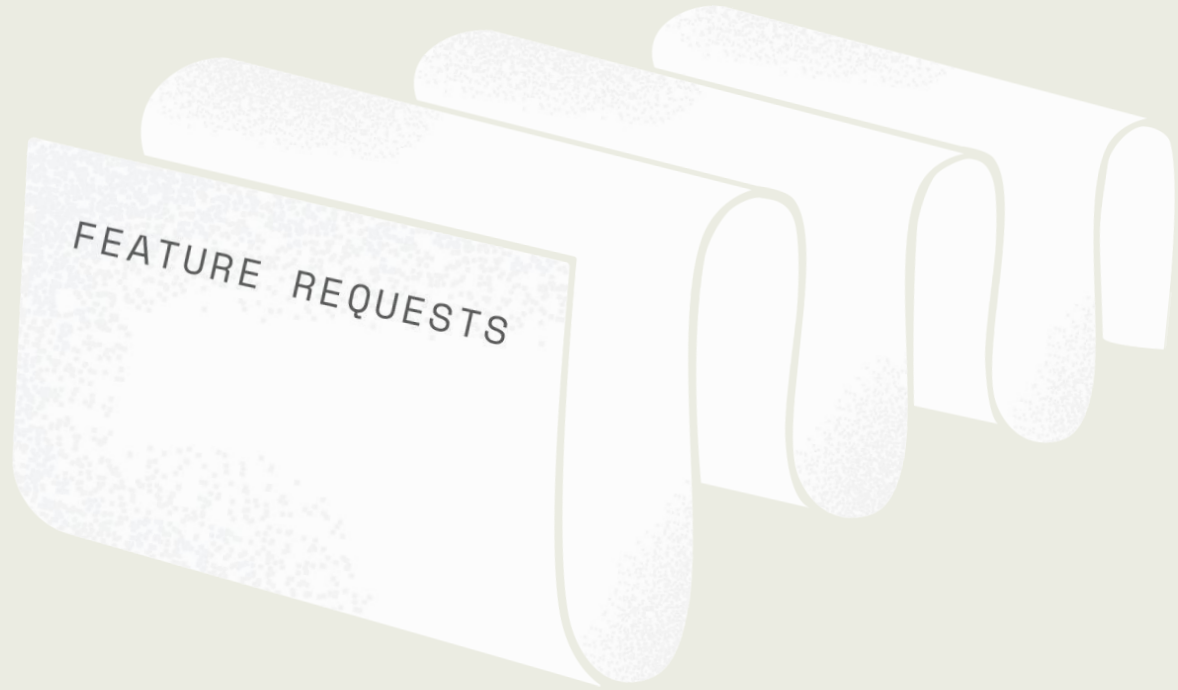


# PRACTICAL TECH DEBT PRIORITIZATION

How to get quality work on the roadmap  
and turn your team's morale around.



FEATURE REQUESTS

# Retro

## Customer value

We deliver great stuff, but we pretend it's not our stakeholders are really happy.

We don't deliver what our customer want. Our stakeholders don't value what we build.

## Fun

We have going to work, and these great fun working together.

Work is boring.



## Suitable process

Our way of working fits us perfectly. Our team has an effective decision-making process.

Our way of working leaves a bit to be desired. We fail to effectively make decisions.

## Learning

We're learning lots of interesting stuff all the time!

We aren't learning anything and don't have time to learn.

## Tech quality (code base health)

We're proud of the quality of our code! It's clean, easy to read, easy to test, easy to change.

Our code is desperately low quality, and technical debt is raging out of control.

## Support

We always get great support & help when we ask for it! I'll make a mistake on our team, it's not held against me.

We keep getting stuck because we can't get the support & help that we need. I feel personally attacked when I make a mistake.

## Speed

We get stuff done really quickly, no waiting on us.

We never seem to get done with anything. We keep getting stuck or interrupted. We keep getting stuck on dependencies.

## Influence

We have influence over what we build and how we build it.

We have no influence over what we build or how we build it.

## Team mission

We know exactly why we are here, and we are really excited about it.

We have no idea why we are here, there is no high-level picture or focus. Our mission is completely unclear and ambiguous.

## Dependability

When my teammates say they'll do something, they always follow through with it.

People on the team aren't really following up on what they say they are going to do.

## Tech quality (code base health)



We're proud of the quality of our code! It is clean, easy to read, and has great test coverage.



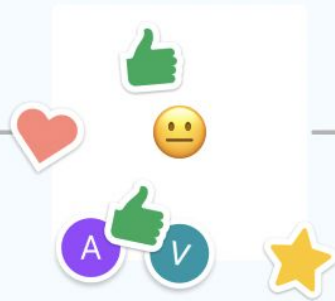
Our code is dangerously low quality, and technical debt is raging out of control.



## Tech quality (code base health)



We're proud of the quality of our code! It is clean, easy to read, and has great test coverage.



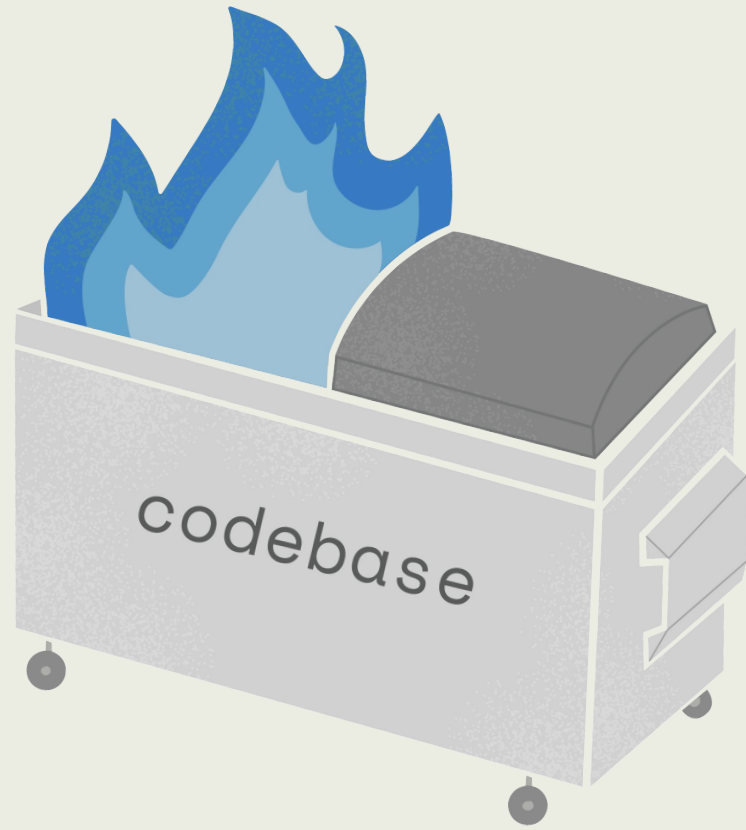
Our code is dangerously low quality, and technical debt is raging out of control.

# ABOUT ME

Engineering Manager @ Zip

Prev founder and foundering engineer

From Oakland, California



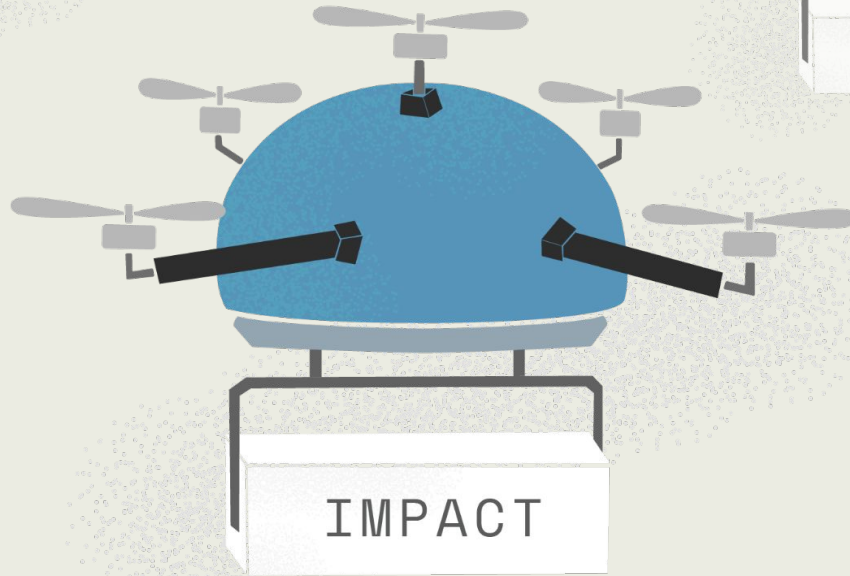
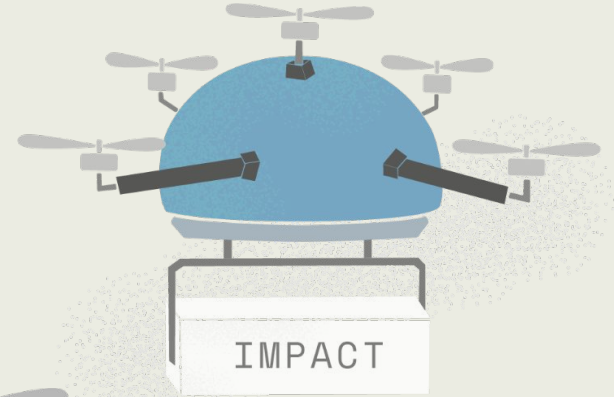
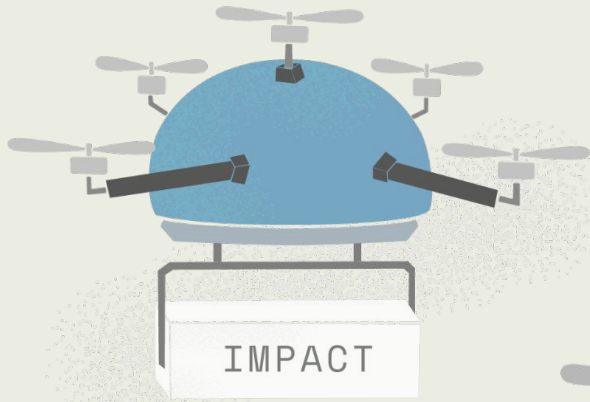
## > **EMPOWER YOUR TEAM TO DECIDE WHAT TO WORK ON**

1. Start with impact
2. Set expectations on resourcing
3. The team decides
4. Continuous improvement

## > **GET THE WORK ON THE ROADMAP**

1. Incept early
2. Craft your impact-first message
3. Use data and team sentiment to track







Your PM

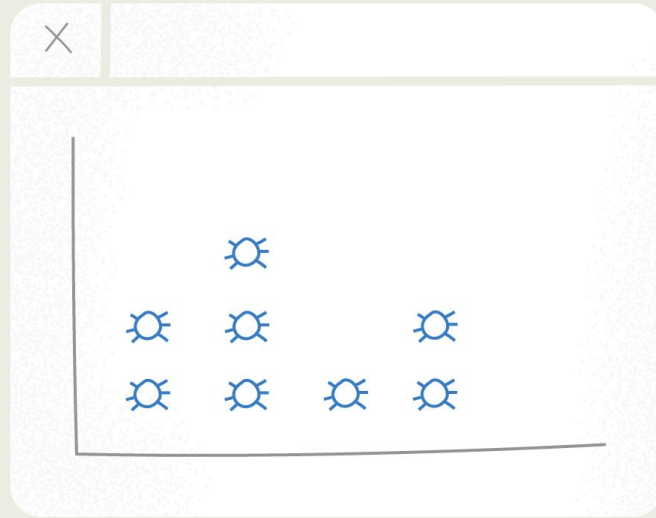
The code for our ABC product has so much tech debt. It makes working in this codebase feel icky and everything takes longer than it should. Building new products on the stack is incredibly difficult so we'd like to invest in re-architecting it.

We have a vision for the future of the ABC product and we want to invest in a feature our customers are eager for. We have a plan to incrementally make improvements to the codebase as we work on this project. Investing in this will also unblock future work in this area.

**START** 

**INCEPTING YOUR**

**TEAM'S NEEDS**



1. Use data
2. Start early

Sometime you'll need to:

**MAKE DEMANDS**



• •

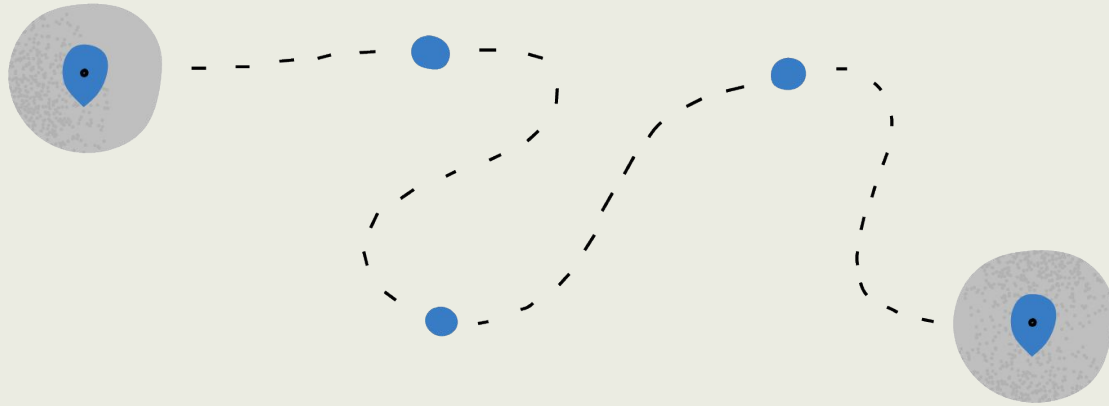
# ADDRESSING TECH DEBT

NEXT **4** WEEKS

• •

# **PREPARE YOUR TEAM**





1. Set expectations on resourcing
2. Teach them to talk about impact first



# Quality Projects



















Table +

Filter Sort ⚡ 🔍 ... New ▾

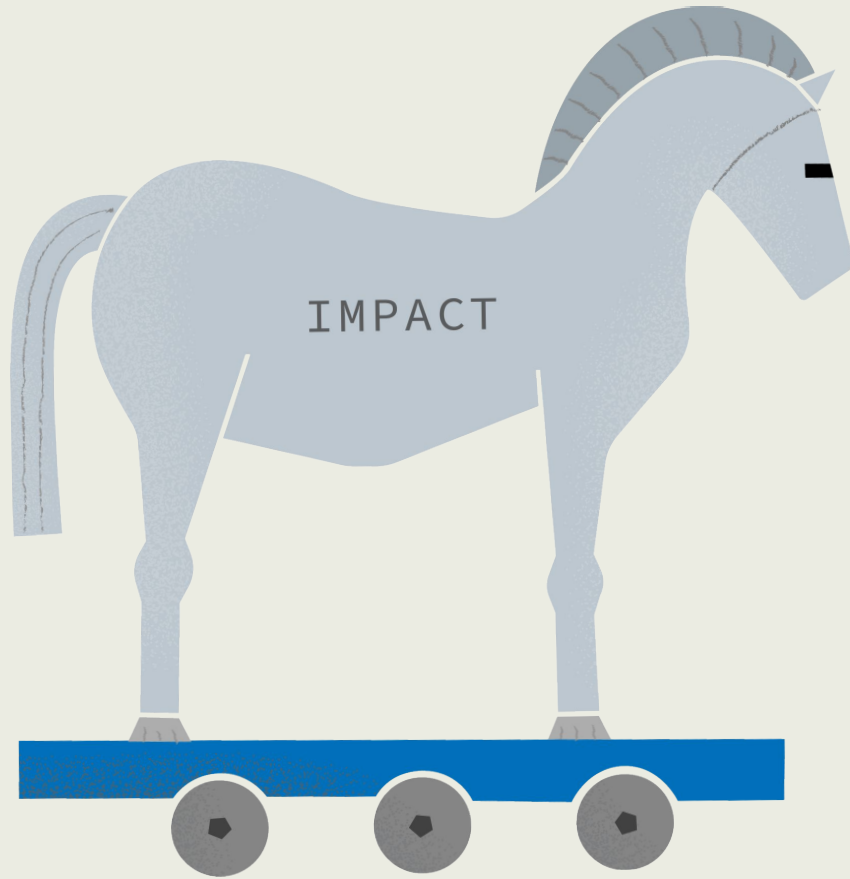
↓ Priority Order ▾ | + Add filter

Reset Save for everyone ▾

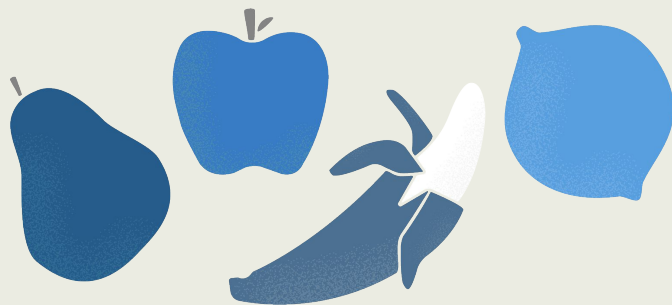
Aa Name	# Priority Ord...	Area	Eng Cost	Impact	Impact Areas	Confidence	+ ...
I&E Mutation Backend Coverage	7	WFC	XS (1-5 days)	M	Essential Maintenance	High	
Migrate CSV upload admin tool to use new GraphQL APIs	6	CSV	S (1-2 weeks)	L	Essential Maintenance Development Velocity	High	
More graceful global question creation	5	Company Settings	XS (1-5 days)	L	Development Velocity Bugs	High	
Misc Tao → Ent P0s and P1s	4	Insights Dashboard	M (2-4 weeks)	L	Essential Maintenance Development Velocity	High	
Category page migrating to GraphQL	3	Company Settings	XS (1-5 days)	L	Essential Maintenance Development Velocity	High	
Company cloning rewrite	2	Company Settings	L (4-8 weeks)	L	Unblock Product Bugs Development Velocity	High	
CSV: Tao → Ent	1	CSV	L (4-8 weeks)	L	Essential Maintenance Development Velocity Bugs	High	
WFC: Enforced CI to prevent lack of Audit Trail support		WFC	M (2-4 weeks)	M	Bugs	Med	
WFC: Tao → Ent		WFC	XL (8-16 weeks)	M	Development Velocity	Med	
CSV Perf Improvements		CSV	S (1-2 weeks)		Essential Maintenance	Med	
Category page performance + TS		Company Settings	S (1-2 weeks)	M	Essential Maintenance	Med	

	 Migrate CSV upload admin tool to use new GraphQL APIs	CSV	S (1-2 weeks)	L	Essential Maintenance Development Velocity	High
	 CSV: Tao → Ent	CSV	L (4-8 weeks)	L	Essential Maintenance Development Velocity Bugs	High
	 Misc Tao → Ent P0s and P1s		M (2-4 weeks)	L	Essential Maintenance Development Velocity	High
	 More graceful global question creation	Company Settings	XS (1-5 days)	L	Development Velocity Bugs	High
	 Company cloning rewrite	Company Settings	L (4-8 weeks)	L	Unblock Product Bugs Development Velocity	High
	 Category page migrating to GraphQL	Company Settings	XS (1-5 days)	L	Essential Maintenance Development Velocity	High
	 WFC: Enforced CI to prevent lack of Audit Trail support	WFC	M (2-4 weeks)	M	Bugs	Med
	 WFC: Tao → Ent	WFC	XL (8-16 weeks)	M	Development Velocity	Med
	Category page performance + TS	Company Settings	S (1-2 weeks)	M	Essential Maintenance	Med
	 Parallelized CSV Export	CSV	L (4-8 weeks)	M	Essential Maintenance	High
	 I&E Mutation Backend Coverage	WFC	XS (1-5 days)	M	Essential Maintenance	High
	 Core_data table for data export job	CSV	S (1-2 weeks)	M	One-offs	High

# **CRAFT YOUR MESSAGE**



# **CULTURE OF CONTINUOUS IMPROVEMENT**



## Quality Fruits

1. 1-2 day efforts
2. Bug corrective actions
3. Nice-to-have quality

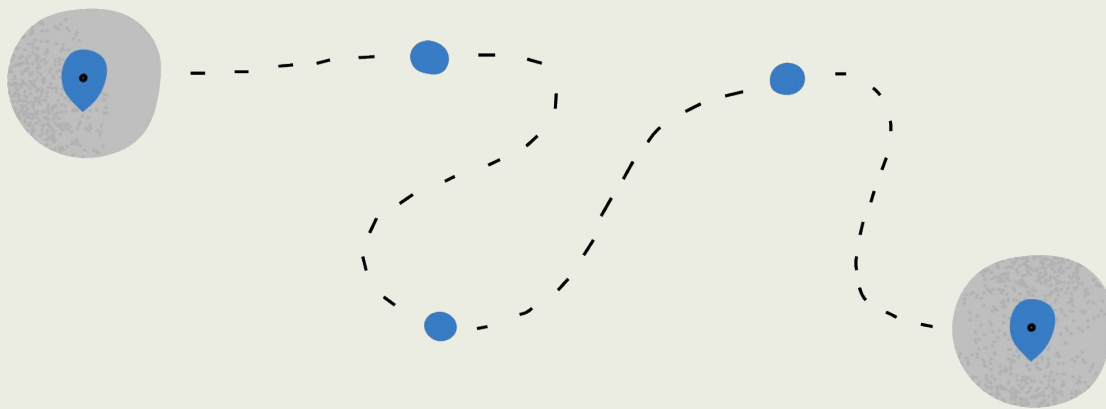
## > **EMPOWER YOUR TEAM TO DECIDE WHAT TO WORK ON**

1. Start with impact
2. Set expectations on resourcing
3. The team decides
4. Continuous improvement

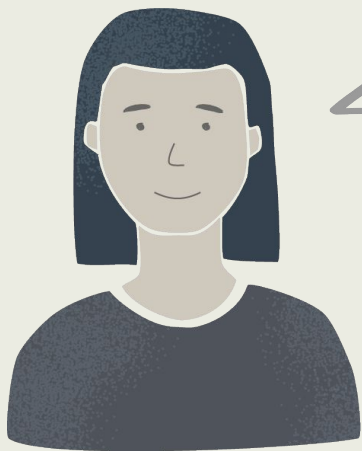
## > **GET THE WORK ON THE ROADMAP**

1. Incept early
2. Craft your impact-first message
3. Use data and team sentiment to track





1. Endpoint performance
2. Significant progress to migrate and modernize our stack
3. Pyright type coverage
4. 32 Quality Fruits



You have somehow communicated with product/design and leadership in a way that makes our lives so much better.



**THANK  
YOU**