

# What To Expect When Working With Lineate





## Partnering on an ambitious project?

You likely have some questions.

After all, you're entrusting a high-stakes project to a bunch of people who you haven't worked with—that's enough to cause some trepidation in just about anyone.

Plus, the uncertainty wrapped up in working with a new partner isn't helped by the fact that the *actual process of work* tends to be shrouded in mystery...

**Which is why we're shining a light on the whole experience.**

Mapping out the journey. Calling out the elephants in the room.

And hopefully, replacing uncertainty with excitement.

### **Welcome to *What To Expect When Working With Lineate.***

This 15 minute read will briefly describe the hallmarks of our work before diving into our project methodology (and a few other goodies).

Ready? Right this way.



# What to expect from us

A quick word on what guides our work.





These days, nearly every engineering firm touts their “world-class talent”.

From every corner, you hear about “rapid development times”, “guaranteed quality”, and so on.

When these selling points all blend together into a beige slurry of marketing, **it can be hard to know what you’re really getting in a project development partner.**

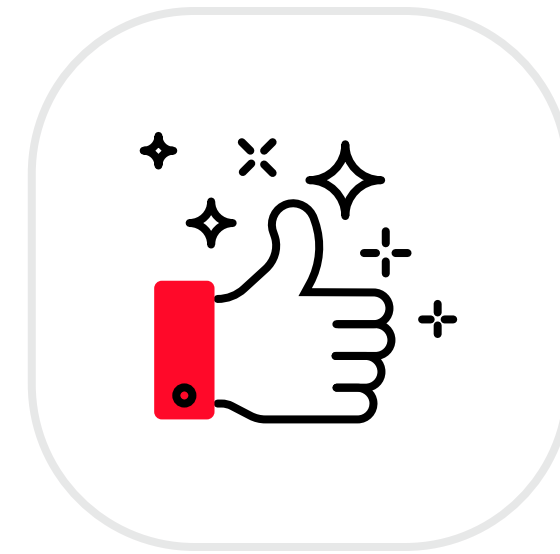
# So, what can you expect when working with Lineate?

---



## A team that genuinely enjoys solving for impossible.

While we hire for the love of code, we train for soft skills, business acumen, accountability, and a commitment to the latest platforms, frameworks and languages (without an attachment to any). This all adds up to a particular set of skills primed to turn the world’s most complex software problems into the biggest business opportunities.



## A “quick and clean” approach.

We design future-proof systems so that work doesn’t have to be redone as markets evolve, and we steer clear of “quick and dirty” methods to avoid inevitable issues. The upfront time we spend understanding your long-term goals means we can work fast without ever cutting a single corner.



## End goal alignment.

Even with our pre-project discovery and precision planning, we know circumstances can change. In the case of shifting project conditions, we’re easily adaptable—and always focused on the end goal.



## Flexibility as a foundation.

Simply put: we're easy to work with. That extends to work done in the trenches, where our clients benefit from the autonomy and accountability of our engineers—all the way up to negotiating project boundaries and costs. We strive to be the kind of partner that we would want to work with.



## Constant communication.

Even the most talented dev team can flounder in a project setting without the right communication skills—which is why we place such emphasis on cultivating them. And, as we detail in Chapter 3, we set up frequent, purposeful touchpoints to keep everyone on the same page.

And finally...



## Rock-solid code.

We don't consider code functional unless it's exemplary. Code isn't finished without a review, reviews aren't complete without tests, and tests aren't considered successful until the code runs reliably, fast, and we understand *why*.



**Writing excellent code is a joy in and of itself—and as engineers, we never want to lose sight of that.**

# Our Lockstep Method

“Discovery” turned up to 11





Our unique discovery phase is at the core of our service offering. This is where we dive deep into the operational nature of your software challenge, understand how it connects to your greater business goals, and where we begin building a vision for a solution.

From the very first conversation, we're immersing ourselves in learning about your company and your particular needs. We know that the optimal solution to your problem isn't just built with software particulars in mind—it's built to fit seamlessly within your business.



# Here's how we break down our Lockstep method:

## Stage 1: High-level inquiry

### Problem statement

We listen and ask questions in order to sketch out the dimensions of the problem.



### Gather requirements

Diving deeper into the technical implications of the challenge, we collect and consider requirements.



### Initial solution design

With requirements and success criteria in mind, we outline and justify a solution approach.



### Approach confirmation

We confirm our approach, gather and document any remaining requirements before crafting a proposal.

## Stage 2: Internal deliberations

### Basic team and time

We sketch out the initial timeline and team members (and project t-shirt sizes).



### Detailed team and time

We weigh exact team and time bandwidth against project requirements.



### Project plan draft

A detailed project plan is drafted including a GANTT plan, dependencies, and staffing plan.



### Complete baseline proposal

We complete our proposal and prepare for the initial presentation.

## Stage 3: Baseline proposal

### Initial presentation

Our team presents, gets customer approval, and addresses any necessary changes.



### Project plan finalized

Detailed estimates of team, time, phases, and deliverables are completed.



### SOW finalized

A SOW is sent to the customer for signature.



### Kickoff deck completed

We gear up for the last big milestone before work begins: the kickoff presentation.

On average, this three-stage process takes **22 days**.

# The Actual Work

The main course





Once the prep documentation is squared away and signed off, the project work gets whirring.

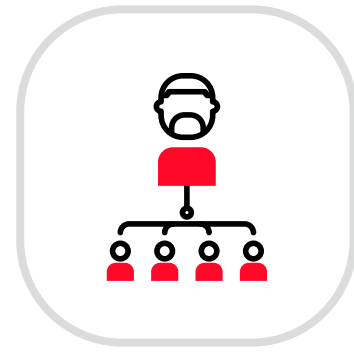
Core to Lineate is the ability to partner with our clients in a strategic way, enabling them to do things they couldn't do normally. Having our team be closely involved with your business is a key part of that.

## Here's how the team you'll be working with is structured out.

(This does, of course, vary per project.)

## The key players:

---



### Unit Director:

Manages the team assembled for the project and oversees the software development process.



### Project Administrator:

Tasked with keeping deliverables on time and on budget.



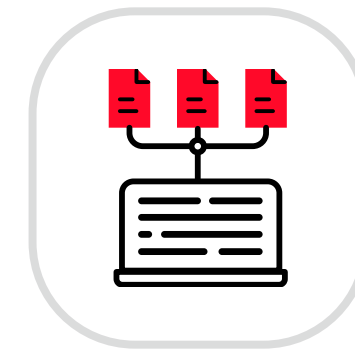
### Local Technical Advisor and Remote Solutions Architects:

Both roles bring their expertise to the solution design phase, and also see that it's carried out to best align with the client's business goals.



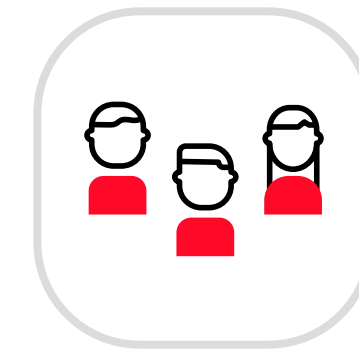
### Executive Sponsor:

Keeps a pulse on client communications, attends key meetings internally and externally, and is the most senior touchpoint, available when the client needs them.



### Client Partner:

A U.S. based role that brings cross-functional skills in consulting, customer success, and business management. Helps to translate client questions and requests into technical tasks for our engineering team.



### The Engineering Team:

We design our teams based on what you need to get the job done. Our Full-service teams include QA (designing test plans and conducting manual and automated testing), DevOps (implementing CI/CD deployment), and regular and senior engineers (frontend and backend, expert in dozens of languages and frameworks).

# An open dialogue

Long-distance collaboration across time zones—it's one of the banes of any cross-team project. That is, unless you adhere to a proven methodology to keep things running smoothly across asynchronous teams.

Communication is at the core of our process—here's a rough breakdown of the agile routines that we follow to ensure internal and external teams stay on the same page:



It's worth noting that our people make it a major point to work *with* your team. There's a reason our clients' development teams are often our biggest champions—we handle all the complex tasks that would take up hours of their time, so they can focus on what they do best.

Frequency	Topic	Description
4x a week <i>(Initially—we scale this down as we get into a groove.)</i>	<b>Stand-ups</b>	Time for internal and external teams to discuss updates and next steps.
Every 2 weeks	<b>Sprint Planning/ Backlog Grooming</b>	Time spent strategizing towards the next sprint, as well as prioritizing the backlog queue.
Every 2 weeks	<b>Sprint Demo</b>	Lineate team will demo updates and new features to stakeholders ahead of further testing.
Every 2 weeks	<b>Sprint Plan Report</b>	Created after sprint planning: contains expected velocity, total points, tickets to be done, and any blockers and planned resolutions.
Every 2 weeks (mid-sprint)	<b>Sprint Half-Time Report</b>	An update on current ticket state, bugs reported, plus any new tickets added.
Every 2 weeks (end of sprint)	<b>Sprint Finished Report</b>	Contains sprint burndown chart and status of all tickets (planned and not) in the sprint.
Weekly	<b>Project Status Report</b>	Team progress towards Epic / Milestone, # of tickets/points completed & remaining. Statistics on backlog, standard metrics (velocity, bug leakage rate), and comparison since last sprint. Statistics on any unestimated tickets or large tickets that need to be addressed.
Ongoing	<b>Chit-chat, Q&amp;A, and updates</b>	Day-to-day stuff.

# What's Next?

On loose ends





As a project draws to a close, clients tend to notice a few things.

The first is all of the structured documentation we've left in our wake. If the project ever needs further tweaking or building upon, we want whoever is doing it—even if it's not us—to have an easy time doing so. Usability is where it all begins and ends.

The second is that their development teams have a good sense about what we did, why we did it, and how. (Working transparently has its perks.)

Lastly, what we built *works*. “Future-proof” isn't just a buzzword for us. We build it *once*, the right way, so it doesn't need rebuilding in a year or two.

## And there you have it.

Curious to see if we can help you create  
company-defining,  
customer-wowing,  
shoot-for-the-stars  
software?

[Book a 15-minute fit call](#)