What Happens When You Lack Senior Engineering Leadership

Allan Wintersieck
Feb 25 · 4 min read ★

I don’t think that a single “perfect” organizational structure exists. Every company has different needs and therefore has different talent requirements. Some team structures are flat, some are deep. No matter how you set up your technical team, if there are processes for shared ownership of your product, you can succeed.

However, there is one position that every company/team should prioritize: senior engineers.

No matter your industry, size, or tech stack, you need senior engineering leadership to guide your software products and internal processes. Without this senior leadership, unfortunate things can happen, some more damaging than others.

Here are some of the most common occurrences that can happen when you lack senior engineering leadership.

Fall Into a Rewrite Trap

20 years ago, Joel Spolsky wrote an excellent article that I still refer people to today. It highlights one of the worst decisions a software company team can make: rewriting code from scratch.

While I recommend reading the full post, the general concept is this.

All engineers have a desire to throw out their predecessor’s code and build something shiny and new. Generally speaking, when engineers are first introduced to another developer’s code, they think they can write something better.

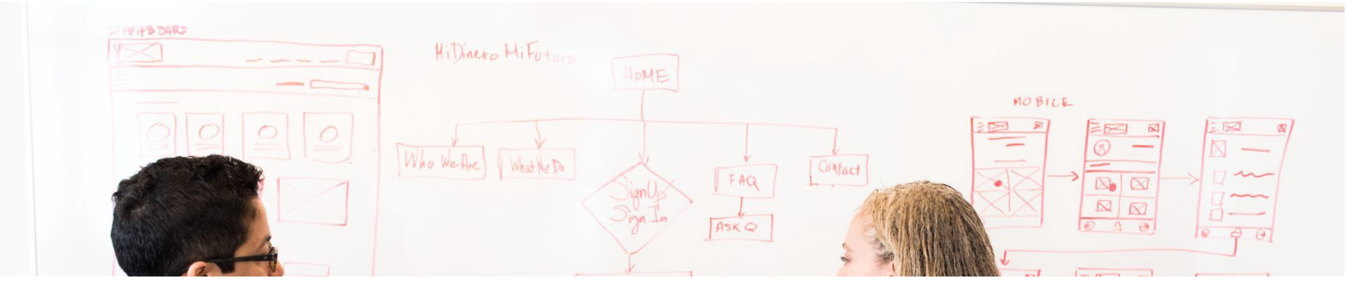
Senior engineers can relate to this, but from their years of experience, they realize that this isn’t a good idea.

They understand that a codebase might be “messy.” But software is most useful when it accurately models the world, and the world is decidedly messy. Some of that code may be ugly because it needs to capture the essential complexity of the domain.

In the words of Joel, “The idea that new code is better than old is patently absurd. Old code has been used. It has been tested. Lots of bugs have been found, and they’ve been fixed.”

Your senior engineers understand this concept and can reel in junior developers who, while talented, sometimes lack this discipline.

You Get What You Asked For, But Not What You Wanted



You also need senior engineering leadership to ask essential business-minded questions before any developmental work happens. Even the most thorough product roadmap will likely have some holes that senior engineers will need to fill in. Without this experience and knowledge, these holes will go unfilled.

In other words, you get exactly what you asked for–but not what you wanted.

I see this scenario happen frequently when clients of ours work with offshore development teams. The developers take everything too literally and don't consider human logic.

An example of this might be an online clothing store.

The product owner says, "Our users should be able to filter the clothes by color."

It's easy to get a list of all the named colors of clothes from a database ("wine," "pearl," "plum") and throw them in an alphabetic dropdown list, but this is a complex feature!

A senior engineer will probably spend some time thinking about the best way to choose colors, and also implement a "nearest matching color" algorithm so that denim, admiral, and lapis all appear when users select "blue."

Having a senior leader to ask questions, clarify features, communicate with developers, and then consistently check their work is essential for any complex product.

Junior Engineers are (Sometimes) More Susceptible to Trends

Your junior engineers may have recently completed programming boot camps or degree programs.

They were exposed to the latest trends and programming languages, which is, generally speaking, a great thing. However, just because a language or tool is new doesn't mean it's better.

Today, new language and tools have stellar marketing; junior developers might think they're the best thing since sliced bread. But as we all know, marketing campaigns don't always correlate to usage and reliability.

An example of this is MongoDB vs. PostgreSQL. To a newbie, MongoDB looks wondrous. It's a new(er) database program marketed as, "the most popular database for modern apps."

That sounds great. Maybe we should start building our databases in MongoDB.

This is when a senior engineer needs to step in. With their extensive experience across a variety of database programs, they will be able to consider the product's needs, budget, and framework and then select a database that achieves the product's business goals.

Engineers Can Snow Non-Technical Leadership

Odds are, your leadership team is not acutely technical.

Odds also are, your developers are not in tune with your five-year business strategy.

There will be times when parties disagree on features, priorities, timelines, frameworks–you name it. If your leadership team doesn't speak dev, you can easily get snowed.

In other words, developers might claim it has to be this way, using purposely head-scratching jargon to make you concede.

To avoid getting duped by a developer, there needs to be technical leadership to filter and translate each team's underlying needs.

Final Note

This post isn't an attack on junior developers. After all, your junior developer will learn and grow into senior engineer positions relatively fast. Instead, it's a case for hiring strong senior engineering leadership that has technical chops but can also:

- Communicate with non-technical team members
- Help solve issues across a variety of technologies
- Understand the business side of a product
- Lend their years of hands-on experience to their team members

Getting this hire right will ensure that your product is planned, built, launched, and maintained at the highest possible level.

. . .

Want to learn more about software development and product management? Sign up for my email list here. I'll send you an article about once per week.

Software Engineering Software Development Leadership Software

Senior Software Engineer