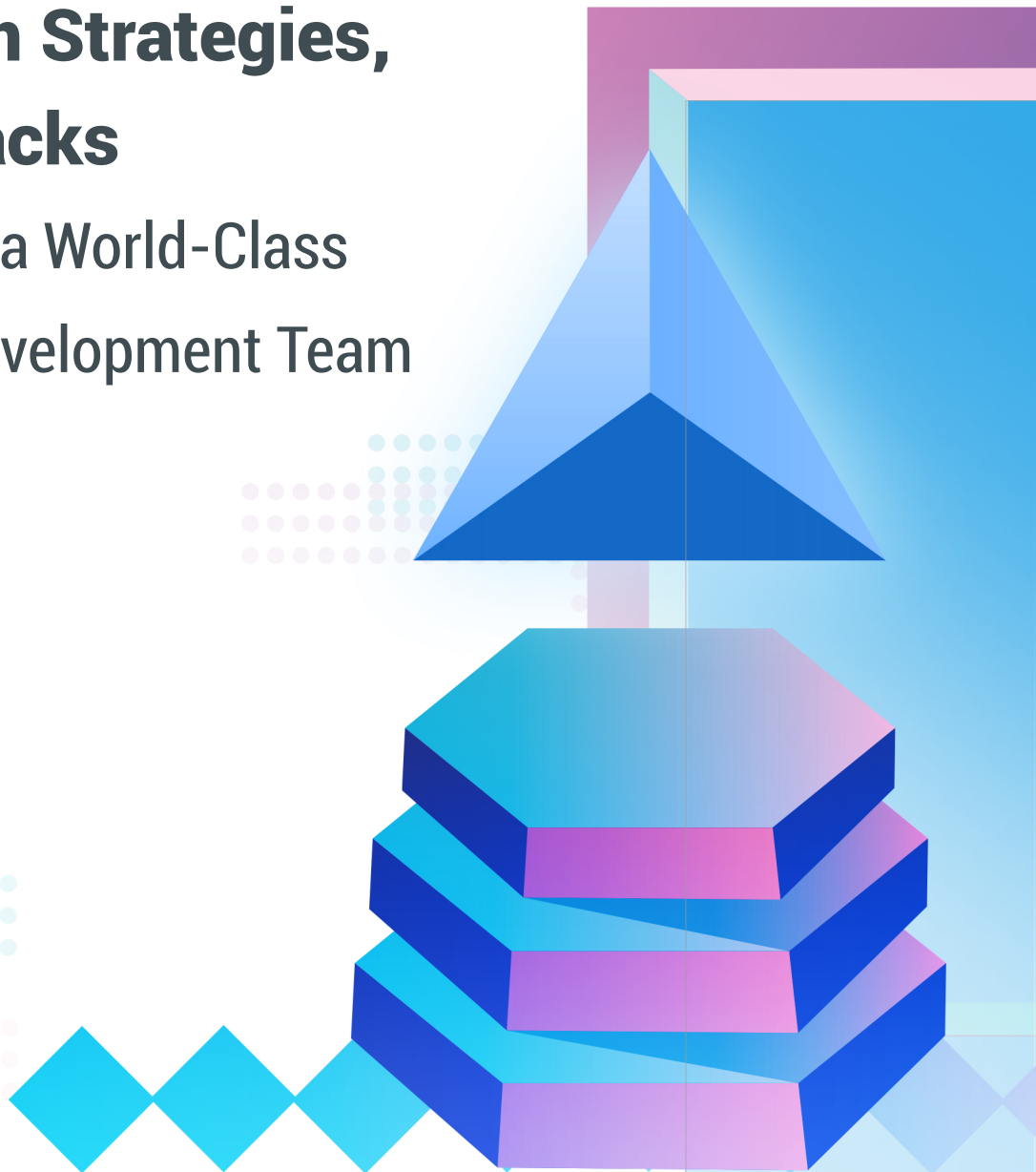


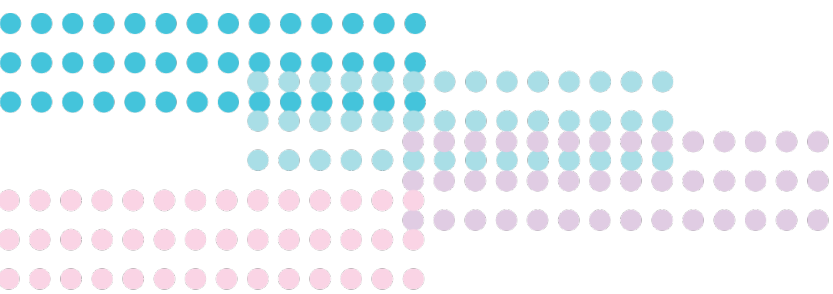


Build & Ship

**39 Proven Strategies,
Tips & Hacks**

for Building a World-Class
Software Development Team





Most of us want to be our best selves at work. We want to be more productive, more efficient, and work effectively. We want to spend more time doing the tasks we enjoy and less time doing the ones that are boring or tedious.

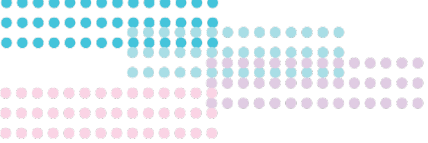
Unfortunately, many teams and managers don't take the time to consider how to improve and optimize the performance of their team. They put specific policies in place and let them go--forever--without considering how they could be made better.

It's clear to most people that the most effective teams--especially in engineering--are those that are engaged, challenged, and motivated. Teams that feel they have autonomy and direction tend to perform much better than those that are given strict, top-down directives.

In order to accomplish that, some companies need to change their thinking to embrace new practices that inspire and motivate team members to do their best work.

This guide is meant to serve as a reference for team leads, project managers, executives, and developers themselves. It's a reminder of how even small changes to the operations and culture within an organization can dramatically shift the happiness, motivation, and effectiveness of employees and teams.





TEAMWORK

Forming a great team is about finding people who are able to work together in a way that allows them to accomplish more as a group than they could as individuals. There are many factors that go into fostering a positive team environment and harboring teamwork and collaboration.

In this section, we will look at some ways that you can make your teams better at working together and with other teams.

1. Build stable teams

Science shows that teams that stay together tend to function better over time. This means it's important to build stable teams that will be able to work together of a long period of time. As they become more comfortable, they become more effective and efficient. ([Read more](#))

2. Set expectations about risk and failure

Key to encouraging risk-taking and innovation is to have a clear and upfront discussion about the role of risk and failure inside a team. This can be a difficult discussion to have, but it should be clear that in order to pursue new and exciting ideas, the team will have to feel that they have the freedom and ability to take some risks without the threat of punishment if things don't work as expected.

3. Institute policies that allow all team members a chance to express their ideas

One common issue within teams is that more vocal members tend to emerge with their ideas and others may be overshadowed or never heard.

Meetings, discussions, and planning sessions should be structured in a way to give all team members a chance to contribute their ideas. Not only will this give the team more ideas to work with (and likely improve the overall quality of ideas), but it will also help everyone on the team feel more comfortable about speaking up and sharing their thoughts.

4. Enforce “no stupid questions” policies and encourage questions and clarifications

Any team where people are afraid to ask questions is one that will struggle to grow and improve. This kind of fear breeds insecurity and animosity. Asking questions is healthy and will foster better communication and collaboration.

5. Focus on team fit above raw technical abilities

It can be tempting for companies to look for “rockstar” developers in hopes that they will elevate the performance of the entire team. But, studies have shown that a great team that works well together will vastly outperform a group of incredibly talented engineers who do not work well as a team. ([Read more](#))

6. Manage for team familiarity to increase speed to optimal performance

Studies show that teams that have a higher level of familiarity (measured by the amount of time spent working together previously) will perform better, deliver faster, and stay within the budget. This is largely driven by the fact that most teams undergo a bonding process characterized by four stages: Forming, Norming, Storming, and Performing. Teams that have already gone through the early stages (i.e., they are familiar with one another) will more quickly achieve optimal performance in the final stage. ([Read more](#))

7. Set aside time to recognize people's good work

It's probably obvious that your team should praise and recognize individuals whenever possible for the work that they've done. But, to take it a step further, you can create a standing time each week to recognize the work done by individuals in the week prior. This will help to boost morale and make team members feel the importance of their contributions. ([Read more](#))

8. Use a broader set of team metrics and individual metrics

Teams often measure group metrics of performance like velocity or burndown and individual metrics around time spent, bugs resolved, or other. But these metrics are only indicative of performance and aren't targets in and of themselves. For example, if the team's velocity is suffering, the solution is not to simply "improve the velocity"--you need additional metrics to tell you what is causing the slowdown and understand what it means.

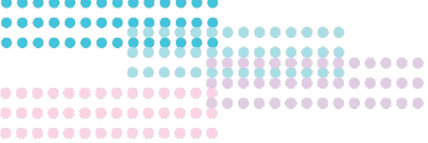
It's best for your team to establish a broad set of metrics that measure your process, performance, outcomes, and teamwork. [\(Read more\)](#)

9. Enforce transparency

One of the most important aspects of any team is trust. And one of the best ways to foster trust is to be transparent--open and honest. Sharing transparent details about progress, goals, and metrics goes a long way toward helping the team feel unified. [\(Read more\)](#)

10. Use the stepladder technique to encourage collaborative problem solving

The stepladder technique is a simple way to ensure that a broad range of ideas are integrated and considered within a discussion. This really simple method can help your team uncover better ideas and make sure more voices are heard. [\(Read more\)](#)



TIME MANAGEMENT

One of the most important factors for any team is how they manage their time. There are a finite number of hours and minutes in any given day, week, or sprint. So your team must learn to manage that time effectively and efficiently--both as individuals and as a group.

This doesn't necessitate strict, top-down time management policies. But, rather, it focuses on smart use of time and insight into how each individual member of a team can best master their own schedule.

11. Use time tracking as a learning tool, not a performance metric

Managers often use time tracking as a way to control their developers and measure their performance. But, time in versus code out is not a good way to measure or monitor performance. Time tracking can, however, be a useful tool for allowing individual developers to measure their own skills and performance and allow your team to make better estimates on future projects

Because of this, time tracking should be used as a learning tool and not a control mechanism. ([Read more](#))

12. Minimize/automate reporting

One thing that most software teams have in common: They hate reporting and tracking.

It's seen as unnecessary, superfluous, and often wasteful of the team's time to take hours out of the week in order to record what they have accomplished. Even so, the data (when used correctly) can be very valuable and insightful. Because of this, your team should always find ways to automate the tracking and reporting process. Minimize the actual time spent on these "overhead" activities and the team will be happier. ([Read more](#))

13. Create time for trials, risk, and experiments

Within the confines of a normal sprint or development cycle, there is usually not a lot of opportunity to experiment or try out a risky trial that may or may not pay off.

But these kinds of activities are critical for keeping your team engaged and finding ways to innovate and improve the software you're building.

14. Implement "flow" tracking to help team members understand their peak work times

We all tend to have specific patterns of when we are most focused and productive. Some of us may work best on logic-intensive projects in the morning and do better at small, quick tasks in the afternoon--or vice versa.

Your team can use tools to measure "flow" in order to help individual teams maximize their own performance by understanding their work patterns. This will also allow your team to work together more effectively as they are able to understand how others work best.

15. Review time tracking as a team

If your team uses time tracking, it shouldn't be an individual effort

16. Focus on output, not strict time controls

One major mistake that many companies make--and why many software teams hate tracking time--is to focus on the strict control of time (e.g., how many hours were "wasted" on specific tasks) rather than the actual result.

This is problematic for any creative profession that centers around problem-solving because tasks are almost never linear. They require analysis, incubation, and experimentation.

To an untrained manager, a timesheet that has 7 hours billed to "brainstorming" and 30 minutes spent writing actual code will seem like a colossal waste of time. But, in reality, this may be the necessary process for engineers attempting to solve difficult problems. So it's important to focus on the output (deliverable) rather than how or how much time was spent on which tasks.

17. Allow time for social conversation, discussion, and debate

One of the best ways to improve employee motivation and productivity is to actually allow for more time for interaction and conversation.

While it may seem counterintuitive that less time spent at a computer can lead to more work being done, there are several studies that show it to be true. The fact is that no employee can work effectively all of the time, and having time for social interaction will help foster more trust and comfort that leads to better teamwork all around.

18. Mandate time off for all team members

To go even a step further, you may want to consider instituting manual time off windows for members of your team.

Time away from work and from the office will allow your team members to rest and recharge. While it may have a marginal impact on your team's productivity during the time when they are gone, it will make people more productive and creative when they come back to work. ([Read more](#))

19. Use hours in addition to story points to understand changing schedules

One of the great things about Agile methodologies is the transition from strict, time-based estimates to relative systems like story points. This change has been great for many teams.

But, for internal planning, it's beneficial to also correlate those story points to estimated time. This way, if you know, for example, that the team will be taking 20 total hours of vacation over the course of the upcoming 2-week sprint, then you can adjust the scope of the sprint to accommodate for the change.

20. Identify and stick with a time-management philosophy

There are many approaches to time management and productivity. One of the most popular is the [Pomodoro technique](#). But there are also others like the [Getting Things Done philosophy](#). Your team should explore these and encourage team members to adopt one that works best for them. As with any routine, the key to adopting one of these strategies is consistency. So, don't jump back and forth too often. Give one approach time to take root.

21. Use a tool like Slack rather than email

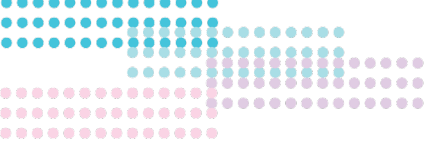
Most modern teams have already moved away from emails to instant messages and chat apps. But, it's worth noting that there are some valuable advantages to having quick, real-time communication with the entire team.

Your team can be more productive and spend less time managing and reading email. In addition, anyone involved in any particular project will have full transparency into the communications between other members of the team.

22. Institute blackout/snooze time for team communications

Although a tool like Slack may increase communication and save time, it can also be a huge distraction throughout the day.

Encourage your team to use the "snooze" feature or disable notifications during work periods. Then, check in periodically throughout the day.



PROJECT MANAGEMENT /OPERATIONS

When it comes to actually moving a project forward and making sure that all of the pieces are in the right place, things can quickly fall apart without

23. Build cross-functional teams

Many companies still form functional teams. This is almost always less effective and efficient than having multiple cross-functional teams that can have autonomy. In an ideal world, your team should be able to build and ship something from start to finish without relying on any other teams or departments. ([Read more](#))

24. Always assign a point person

It may be obvious to many organizations, but every engineering team should have a single point person. This person is in charge of communicating with outside teams and they are ultimately responsible for the work that gets done. It could be an engineer manager, product owner, or other position—just make sure it's clear who it is. ([Read more](#))

25. Shorten your iterations

In agile development methodologies, the length of the iteration is often a contentious discussion. It's impossible to say exactly what is ideal for every team, but, as a general rule, shorter iterations have two main advantages. First, they allow you to remain firmly within a predictable time horizon. There is less chance for things to change or shift between the start and finish of the iteration, which means you can deliver what's needed at that moment in time.

Secondly—and maybe most importantly—shorter iterations mean that your team will have more frequent opportunities to learn from the results and improve the process.

If you have a 4-week iteration, then you can only have 12 times through the loop in a single year. On the other hand, even cutting that down a 3-week cycle will bump you up to 17 cycles per year—a 41% increase. ([Read more](#))

26. Create a “failure metric” to measure and normalize risk taking

It may seem counterintuitive to “measure” failure—but it's an interesting hack that encourages risk-taking and experimentation.

Having a metric specifically designed to measure failures (and learning!) will allow your team to control and understand those risks. It will normalize the failure and make it easier to accept when it occurs, which makes it easier to take risks that may result in such failure. ([Read more](#))

27. Focus on intrinsic forms for motivation over extrinsic

Everyone is motivated by a series of intrinsic and extrinsic factors. Intrinsic are those that are shaped by our desire for achievement. Extrinsic are the external factors that we pursue largely for social reasons.

Cash bonuses and even praise and recognition—while sometimes effective—should not be the main forms of motivation for your team members. Instead, focus on fostering autonomy, mastery, and purpose. Make people feel like their work truly matters and they'll be motivated to do great work.

28. Minimize handoffs between teams

Hand-offs and kick-offs are often bottlenecks for software development. Any time there is a transfer of knowledge from one group to the next, there is the opportunity for error, oversight, and mismanagement.

To reduce this risk, you should try to minimize the instances of handing off a project between people or teams whenever possible and adjust your operations flow. ([Read more](#))

29. Improve your onboarding

Onboarding can be a difficult process for many companies. There is often a lot of knowledge that needs to be transferred. There are new process to learn. Not to mention new people to meet.

So, focus a lot of time and effort on analyzing and improving your onboarding process. Extra effort spent during this critical time can pay off immensely for a team member's ability, motivation, and fit later on down the road.

30. Remind people “why” they’re doing their work

Too often, management and executives will dictate work as simply something that needs to be done. They don't offer much in terms of the motivation behind the task or even the mission of the company.

For any project, reminding people how it ties into the larger vision and how that relates to the overall mission of the organization can give them extra motivation to deliver. [\(Read more\)](#)

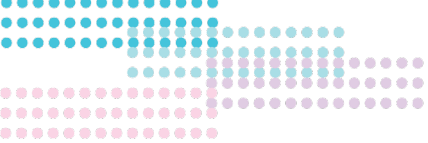
31. Ensure that everyone on the team agrees to the goal/vision at the start of a project

Leading up to the kickoff of any project or the start of any iteration, there should be discussion and debate about what is upcoming. But, once the process moves forward, it's incredibly important that all members of the team are 100% on board with the goal and vision.

If there's animosity or unresolved disputes, it may lead to members of the team feeling resentment or even sandbagging the whole project. Be sure to talk through any disputes and resolve issues during the planning process to avoid problems later on. [\(Read more\)](#)

32. Focus on “pulling” the team rather than “pushing” them

In order to improve productivity, some team leaders will try to “push” their team. But, this is usually not an effective strategy. As noted in other tips from this guide, creative problem solving is about motivation. The leader's job is to motivate the team—“pull” them toward success—rather than expecting to be able to control their work directly. [\(Read more\)](#)



CULTURAL & SOCIAL FACTORS

Work is almost never just work. Given that most of us spend at least 40 hours per week with our teams and coworkers, we often develop social relationships. Because of that, it's important to give careful consideration to the kind of social culture that exists within your team.

Not every team will be full of best friends, but the more people feel comfortable working together as well as trust and respect for one another, the better they're able to achieve their goals.

33. Control for burnout

Once someone is burned out, there's usually no coming back. So be incredibly careful how far you push your engineers and for how long. If you expect everyone to work at 80-100% capacity at all times for months on end, they will grow tired. People can only work at this optimal level for short bursts of time. [\(Read more\)](#)

34. Do more frequent reviews

Many teams have performance reviews on a yearly basis. But that's not optimal. It can be incredibly difficult to discuss all of the work done over a 12-month period. Instead, most reviews end up focusing on the last 2-4 months. So, it would make sense to then hold reviews more frequently in order to discuss a shorter period in more detail.

For much the same reason that software teams prefer shorter iterations with frequent retrospectives, having a periodic performance review is beneficial for everyone. [\(Read more\)](#)

35. Clarify the mission

As a team leader, your job is—in part—sales. You're selling the vision and the mission of each product (and the entire business, in some cases) to the members of your team.

It can be easy for engineers to get lost in the minutiae of any given project or task. So, make sure that you regularly communicate the larger missions and what your team is trying to accomplish. [\(Read more\)](#)

36. Have a “culture interview” when hiring

Culture interviews (or “beer interviews”) are when a team will go out for a beer, coffee, or lunch with a potential hire before they're given a job offer. It gives the team a chance to meet the candidate and interact with them in a social setting.

This is usually used as a final step in the interview process, but it's a good way for the team to assess the fit of that person in a way that can't be easily communicated in a normal interview setting or on a resume. [\(Read more\)](#)

37. Use “now-that” rewards, not “if-then” rewards

Many leaders try to use “carrots” or “if-then” rewards to incentivize people to take action. Things like bonuses that are contingent upon meeting a certain goal or extra vacation time for hitting a deadline are common.

But, instead, consider using “now-that” rewards. In other words, don't offer the reward as an incentive, but provide it as an unexpected reward for a job well done. Studies have shown that this will boost morale while contingency based rewards can actually decrease performance. [\(Read more\)](#)

38. Encourage and provide time for “side projects”

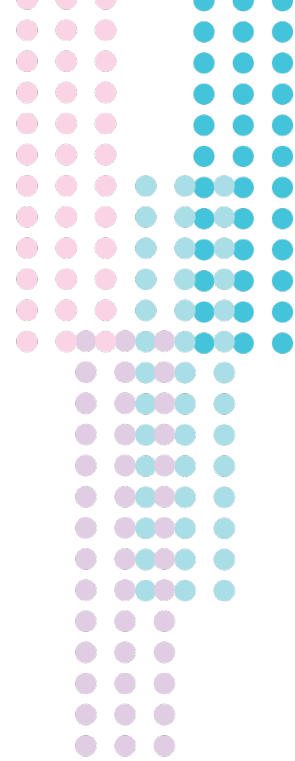
Google’s 20% rule (where employees are told to spend 20% of their time on outside work or projects) has become famous and credited with the invention of products like Gmail. In addition to that, 3M instituted a similar policy, which, in the 1980’s, resulted in the creation of the Post-it note.

Carving out time for your team to focus on projects that are of personal or professional interest to them gives them an opportunity to keep their skills sharp, keeps them engaged in the work, and could also lead to the creation of innovative products or services.

39. Treat “hygiene factors” as a baseline

So-called “hygiene factors” are things like pay, benefits, and vacation. Many times, employers think of these as the most important aspects of a job. But, many studies have concluded that they are in fact just a baseline for employee satisfaction.

Offering a competitive salary can prevent some employee dissatisfaction, but it won’t actively promote employee satisfaction on its own. Other traits of the position--autonomy, mastery, and purpose--are what will actually increase employee engagement and motivation. ([Read more](#))



Time to Code

Seamless, almost automatic time tracking for teams using Microsoft VSTS and TFS.

[LEARN MORE](#)