7pace

Rethinking Timekeeping for Developers: Turning a Timesuck Into Time Well Spent

Christina Marfice

Time tracking doesn’t have to be a timesuck.

In fact, it can be time well spent.

But traditional timekeeping doesn’t work for developers. They see it as a waste of time, and afterthought — or worse: A tool their managers can use to hold their work over their heads.

But what if the problem isn’t actually the time tracking itself? What if the problem is that software teams are just tracking their time all wrong?

Time tracking can actually be valuable for your team and your organization. But first, you and all your team members need a complete shift in the way you frame time tracking as part of your work.

Here’s the key: Stop viewing time tracking as a timesuck. Instead, see it as a way to collect data that can make everyone on your team better at their job.

Table of Contents

Time tracking doesn't have to be a timesuck.

In fact, it can be time well spent.

Most Developers Hate Time Tracking

- It's Hard to Track Creative Work
- Managers Can Use Time Tracking as a Control Mechanism
- Time Tracking Feels Like a Timesuck
- Time Tracking Doesn't Add Value

Why is it so challenging for developers to track time while they work?

Most Teams Suck at Time Tracking

- Teams Turn Time Tracking Into Carrots and Sticks
- Traditional Time Tracking Tools Are Annoying
- Tracking Time is an Afterthought, so the Data is Unreliable
- Teams Don't Use Time Tracking Data to their Advantage

Why aren't traditional time tracking tools good fits for software teams?

Reframing Time Tracking as a Valuable Process

- **Implementation**

  Give Developers Multiple Options for How to Track Their Time
  Automate the Tedious Parts
  Link Time to What it was Spent on
  Keep Time Where You Keep Work
- **Process**

  Use Time Tracking to Plan and Forecast
  Be Proactive About Identifying Unrealistic Scoping
  Help Individual Team Members Use Data to Improve
- **Deployment**

  Know the Relative Nature of Pace
  Never Use Timekeeping As a Control Mechanism
  Use Historical Data to Inform, Not Enforce

What Makes 7pace Timetracker Different?

You've Reframed the Process. Now You Just Need the Right Tool.

Most Developers Hate Time Tracking

And there are a lot of reasons for that. For many developers, tracking time doesn't feel like a valuable part of the job — it feels like, at best, a waste of time and an afterthought, and at worst, a way for managers to control their teams' time and work.

Maxim Lutsan

7pace CTO

“Tracking time is something that comes on top of daily work and requires additional efforts,” 7pace CTO Maxim Lutsan explains. “Just consider why kids (and even adults) forget to wash their hands. Everyone knows that it is healthy but still many people forget to do it or do not do it properly. The same applies for time tracking: One should not forget and always do it; otherwise it does not work. The time tracking solution should help developers by being as unobtrusive as possible and helping them track time quickly.”

It might be a tough sell to convince engineers that time tracking can actually be a valuable part of their job, but it can be done.

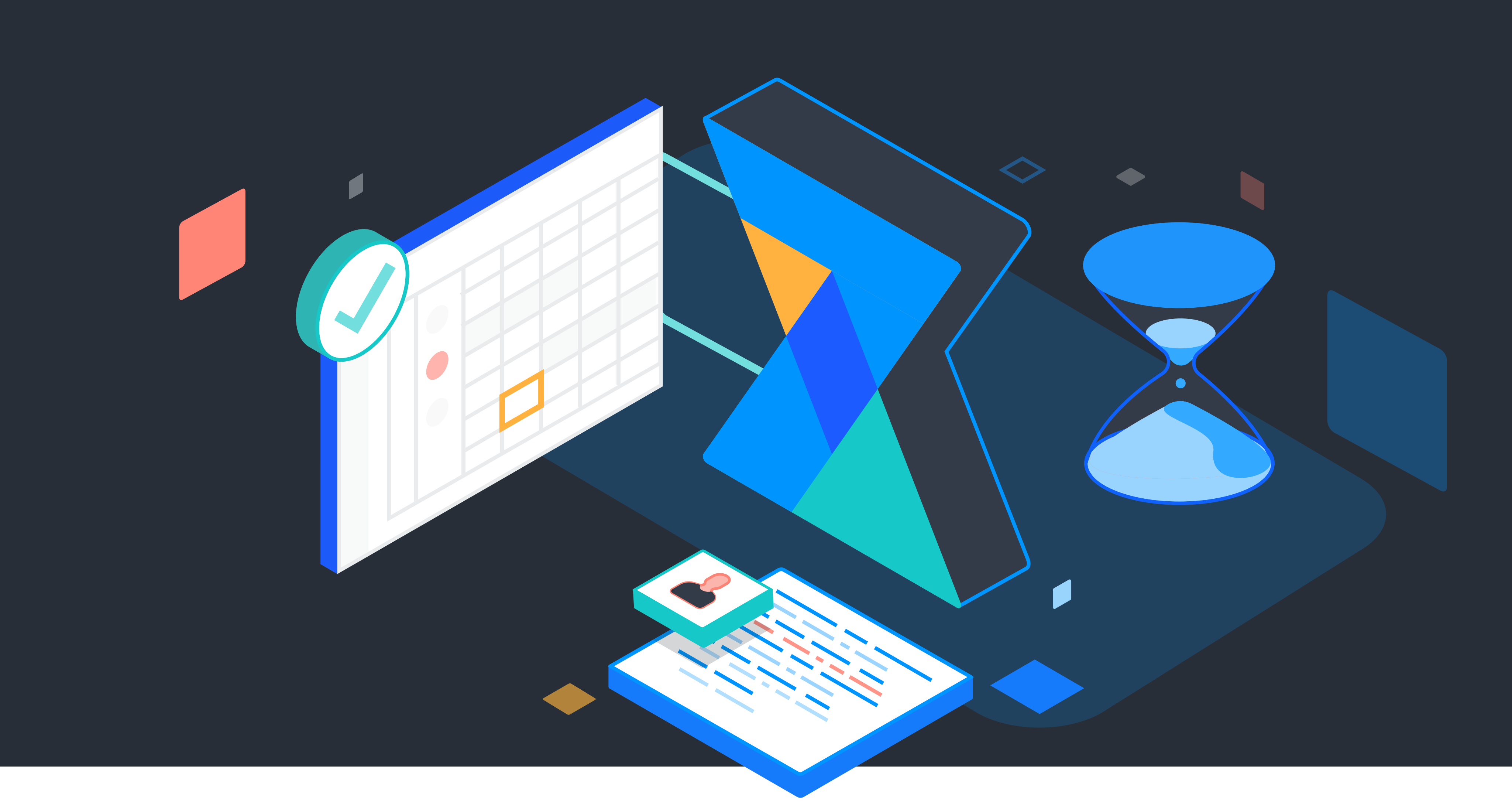
First, though, it’s important to understand why so many developers are against traditional time tracking methods. Let’s look at some of the most common reasons developers hate tracking their time, from the practical to the philosophical.

It’s Hard to Track the Kind of Work Devs Do

As anyone in a problem-solving profession knows, time tracking for that kind of work is just not as simple as sitting down, doing a task, and recording how long it took.

Think about how you accomplish a job as a dev. You sit down at the start of your day and make a prioritized task list for what you’d like to get done. You start on the first task and work until you get stumped by a problem, so you move on to another task, or you take a coffee break to rest your brain and come at the problem from a new direction.

You might try one solution to the problem that ends up not working out. It might be the next day before you land on a solution that allows you to make some progress.



Now — how do you track your time for that workday?

That's the struggle developers have when they try to track their time in traditional ways. There's no clocking in and out when you take breaks or pivot to different tasks, because the mental work continues even when you're "off the clock."

That brings us to the next reason developers hate time tracking: It can be used to control them and their work.

Managers Use Time Tracking to Control People

This is the big one: Time tracking has an image problem.
That's because a lot of managers use it as a way to control their teams.

While an Adobe survey revealed that 77 percent of workers want full control over their work and the time they spend doing it, another study revealed that 18 percent of employees say their managers track time for them.

As we already know, the kind of work software teams do is rarely linear. It doesn't fit into the confines of neatly tracked time per task. And when developers feel the pressure of having to perform to meet a manager's expectations for how they use their time at work — when they feel like they're being micromanaged and controlled — they're just not likely to do their best work.

But there's more to why developers hate time tracking than this. Some of the problems lie in how we've been taught to look at time tracking as a part of our work.

Time Tracking Feels Like a Timesuck

So many developers hate tracking time because they feel like it's a huge timesuck.

Are they wrong? Only partially.

Time tracking is actually a waste of time for a lot of teams, because they're doing it all wrong.

Whether time tracking is a waste of time depends on why your team members are tracking their time. If they're doing it so managers can control every moment they spend at work, that is a waste of time. But if they're doing it so they can master their own time and productivity, that's not a waste at all.

It also depends on how your team tracks time. If they jot down hours as an afterthought in a Google doc, that's a waste of time. But if they use a program designed to help software teams draw insights from how they spend their time at work (like their average pace) and get better because of it, that's definitely time well spent.

And that brings us to the next reason many developers can't stand traditional time tracking.

Time Tracking Doesn’t Add Value

Not the way most teams and organizations use it.

Time tracking is something developers should see as inherently valuable because of how it can help them manage themselves, their time, and their work, upping their productivity and making them better engineers.

But that’s not how we’ve been conditioned to view time tracking. For a long time, the practice has been dismissed as simply “punching the clock,” tracking the time employees spend at their desks so they can collect their paychecks.

What’s worse is that almost no one stops to think about how we can reframe the entire concept of time tracking to make it into a worthy, valuable part of any job.

So what if we did just that? What if we completely reframed the conversation around time tracking and treated it as an important tool software teams can use for individual improvement and mastery?

That’s our goal. Keep reading to find out how we can get there.

Why is it so challenging for developers to track time while they work?

„You don’t want your developers thinking about the clock while they are solving tricky problems, and neither do they. Development is not a clock-in clock-out career and often the best work comes from the insights while off the clock. I don’t think enough attention is paid to the truly creative nature of development work, which does not lend itself to ‘I spent 8 hours on this.’ Development work was probably ‘I spent 2 hours on this, and then I walked away from frustration, and then in the middle of my run I solved the problem so I jumped right on the computer and fixed it in 20 minutes.’ Time tracking assumes a linear workflow whereas a lot of development may be haphazard.”

ALEX H., DIRECTOR OF BUSINESS ARCHITECTURE AT MTX GROUP

„Frankly, I have never been of the opinion that time tracking is for the developers, as much as it's for management and their ability to plan and allocate resources. Depending on how the task estimates are achieved, it may be a useful metric to see how you're performing based on that. But I don't find an environment that directly compares velocities between employees to foster a healthy work environment, so generally don't draw much attention to them."

JOEL K., IT MANAGER WITH MORE THAN 12 YEARS OF EXPERIENCE IN SOFTWARE ENGINEERING

Most Teams Suck at Time Tracking

The first step in reframing the way we look at time tracking is understanding how teams are fundamentally approaching the subject the wrong way.

These are the reasons software teams suck at time tracking. Solving these problems is Step 1 toward making time tracking a valuable part of your work.

Teams Turn Time Tracking Into Carrots and Sticks

Traditional time tracking methods — and managers who use them as tools to control their teams — put a lot of emphasis on speed, productivity, and efficiency.

But developers' work isn't like that. It takes longer. It requires more methodical, calculated approaches.

Unlike other types of work that strive for maximizing time, the best coding might simply take longer. Quality is greater than quantity in this field.

Sure, there are some developers who will rush to create a solution as quickly as they can. On the other hand, there are developers who take their time, fully analyse problems, and then build solutions. Guess which type of developer tends to do better work?

While the developer who works with speed in mind might be the first to arrive at a solution, we all know quick fixes just create more problems down the line. The problem then becomes self-perpetuating — as developers put out fires by using more quick fixes to address bugs, their quick and haphazard solutions create more bugs, and on and on.

Traditional Time Tracking Tools Are Annoying

Traditional time tracking tools were definitely not created with developers (and making their lives easier) in mind.

They're easy to forget about. They're annoying. They cause engineers to just forget to track their time altogether.

"If you're in the mind-set of logging...every activity you're doing, you're actually going to be really hurting your productivity," Art Miesel, a productivity expert and the author of Less Doing, More Living, told Fast Company. "That touch-in, touch-out mentality is very different from the mentality you need to do creative work, or research, or make an effective phone call."

That pretty much sums up how developers feel about a lot of the time tracking tools that are out there. Putting time and effort into logging tasks and activities feels like a waste, especially when it pulls your mind out of the space it needs to be in to solve complex problems — or, in other words, do your job.

Plus, tools that make it an extra step in your day to log time are easy to forget about, which leads us to the next way software teams are failing at tracking their time.

Tracking Time is an Afterthought, so the Data is Unreliable

According to Les Johnson from Function Point, a Canadian time tracking software company, it's not unusual for teams to attempt to track their time — but to do so totally halfheartedly.

"If you're lucky, time is entered on a daily basis. If not, it's probably a last minute activity at the end of the day on Friday and likely less accurate," Johnson wrote. "Needless to say, the time entered is often a best guess. And if a few extra hours are needed to reach 40, staff tend to inaccurately spread the remaining hours out across the active projects."

The only thing an approach like that accomplishes is making sure team members get paid for all 40 hours they spend at work each week. This halfhearted, narrow view of what time tracking is good for ignores what should be the true purpose of tracking your time at work: The data.

When your team isn't accurately tracking their time, they're creating data about their work that just isn't good for anything. On the flip side, accurate time tracking creates data that teams can use in extremely valuable ways.

And that brings us to the final way software teams suck at tracking time.

Teams Don't Use Time Tracking Data to their Advantage

Keeping accurate time as you work through projects and tasks creates a wealth of data.

Imagine if your time tracker could create reports about your work, and then help you forecast future project times based on historical time data. Imagine if it could give you insights that you could use to improve your work — on an individual basis, not because a manager is breathing down your neck to get projects done faster.

That data can be put to use in all sorts of ways that — far from the carrots and sticks created by overbearing managers who don't know how to use time tracking as a valuable tool — actually make your team better and more productive.

For this, your team needs two things:

To reframe the way it thinks about time tracking, and
A tool that will help it track time while also harnessing the potential of all the data that comes with good time tracking.

We'll start with the first step: How to reframe the way your team thinks about time tracking, so everyone on the team starts to see it for the valuable process it can be.

Why aren‘t traditional time tracking tools good fits for software teams?

> “Many traditional time tracking tools expect us to remember the hours we spent on tasks. I don’t think that way, and I don’t think other people think that way either. When it’s the end of the week and I have to submit my timecard I am taking guesses on where that time went. I have absolutely no idea. Traditional time tracking tools are best estimates, imprecise, and cumbersome.”

ALEX H., DIRECTOR OF BUSINESS ARCHITECTURE AT MTX GROUP

> "I've worked where I only punched in and out, I tracked no time, I tracked time on a project code level, and where I tracked time on a task level. Of all of those, I actually preferred tracking no time, obviously. But if I was tracking, I preferred to just do it on the project code level once a week. Filling out tasks every day is just monotonous. I'm never going to remember to log ALL the time every time I do a 15 minute bug fix, so you're always working off estimates."

JOEL K., IT MANAGER WITH MORE THAN 12 YEARS OF EXPERIENCE IN SOFTWARE ENGINEERING

Reframing Time Tracking as a Valuable Process

The first step in changing how your team thinks about time tracking — turning the process from a time suck to time well spent — is reframing the value of tracking time.

There are three parts to reframing the time tracking process: Implementation, Process, and Deployment.

Implementation

Implementation is all about finding ways to integrate time tracking into the planning and development process.

At the implementation stage, the one thing to keep in mind is this: You're trying to remove the pain points from time tracking. Make it as easy as possible. Remove the tedious aspects of the process. Make time tracking a seamlessly integrated part of the work, so it's no longer an afterthought or a chore.

AUTOMATE THE TEDIOUS PARTS

We've heard from developers already in this book, and one of the reasons they hate tracking their time is because it's tedious, it takes them away from work they feel is more important, and they don't know how to track their time accurately.

So why not automate the process (or, at least, the most tedious parts of it).

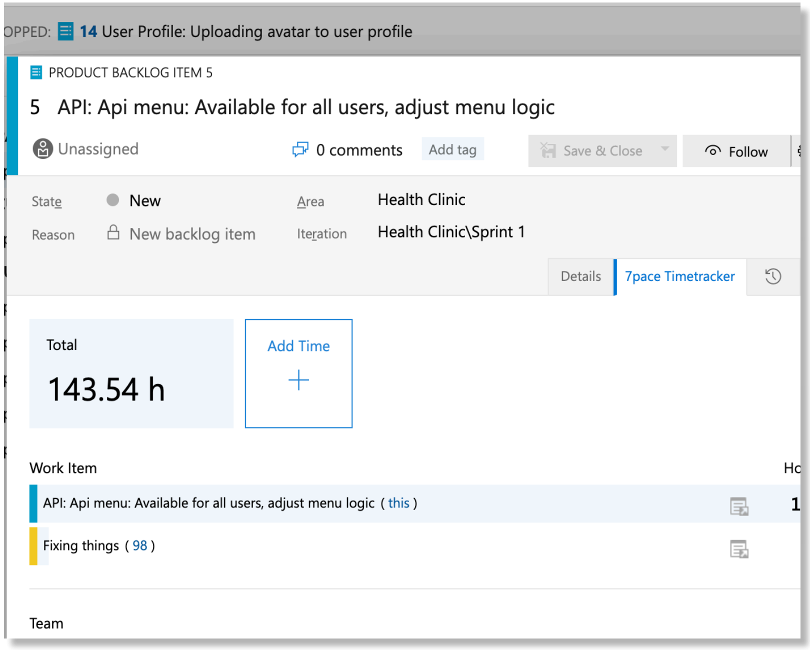
The right time tracking tool will have automation features that solve some of the pain points developers feel about time tracking. Look for those features when you're looking for a time tracker that will add value for your team — not another chore.

LINK TIME TO WHAT IT WAS SPENT ON

Implementing a time tracking process that will add value to your team's work means giving them the ability to link their tracked time to the specific tasks and projects they spent it on.

That means, if they're working in Azure DevOps, being able to track time by each work item they complete. In GitHub, it means being able to link time to each issue they solve. No matter what platform your team uses, their time tracking solution needs to integrate with that platform.

And that brings us to the final part of time tracking implementation.


OPPED: 14 User Profile: Uploading avatar to user profile

PRODUCT BACKLOG ITEM 5

5 API: Api menu: Available for all users, adjust menu logic

State	New	Area	Health Clinic
Reason	New backlog item	Iteration	Health Clinic\Sprint 1

Total 143.54 h

Work Item	Hc
API: Api menu: Available for all users, adjust menu logic (this)	1
Fixing things (98)	

KEEP TIME WHERE YOU KEEP WORK

There's a huge difference between a software that just creates a list of time that's been worked, and a software that's integrated deeply into the tools you and your team already use.

We'll let you guess which one you should use.

This might be the most important part of time tracking implementation for your team — they need to be able to track their time within the same platform where they're doing their work. It keeps time tracking out of the way of the work they're doing, which relieves so many of the pain points that make developers reluctant about the entire concept of tracking their time.

With 7pace Timetracker, a time tab is added to each of your Azure DevOps projects automatically. You can click the time tab from a work item form to begin tracking time, or to see time spent by you and your team members — on that work item and all child items.

It integrates seamlessly and then stays out of the way so you and your team can do your jobs

Process

The next step: Integrate time tracking (and time data) into your team's existing processes. This is where time tracking begins to be valuable for your team — when the data it creates is put to good use.

USE TIME TRACKING TO PLAN AND FORECAST

Planning and forecasting should already be parts of your team's process.

But now, with a good time tracking system implemented, your team will have new tools at their disposal to make it even easier to plan and forecast their work.

Assess your development process and look for ways to incorporate the time track tracking mechanism into planning for sprints, retrospectives, annual reviews, and other steps in the process.

As you complete projects with your new time tracking process in place, you can begin to use historical time tracking data to make better, sharper time and effort estimates.

But keep in mind that time records should only be used as individual metrics to help team members measure and meet their personal goals. Never use them to set quotas on how much work needs to be created within time frames, or to creative competition about which developer is “fastest” or “most efficient.”

BE PROACTIVE ABOUT IDENTIFYING UNREALISTIC SCOPING

One of the biggest benefits to having access to historical time tracking data is the ability to apply it to the future. Knowing your team’s pace on past projects can help inform current and future projects — specifically, it can help your team more realistically scope their work for new features, fixes, and releases.

On the flip side, having historial time tracking data and better estimates for the time needed to complete future projects can help your team be more proactive about identifying and eliminating unrealistic scoping. No more missed deadlines? That’s what teams want to hear.

HELP INDIVIDUAL TEAM MEMBERS USE DATA TO IMPROVE

Far from being a tool for managers to use to itemize and scrutinize work, time tracking data should be used by individual programmers to learn more about their habits, create goals, and measure improvement.

As you make time tracking part of your team’s process, make sure to teach each team member how to use your time tracking mechanism and its data to drive and improve their own habits.

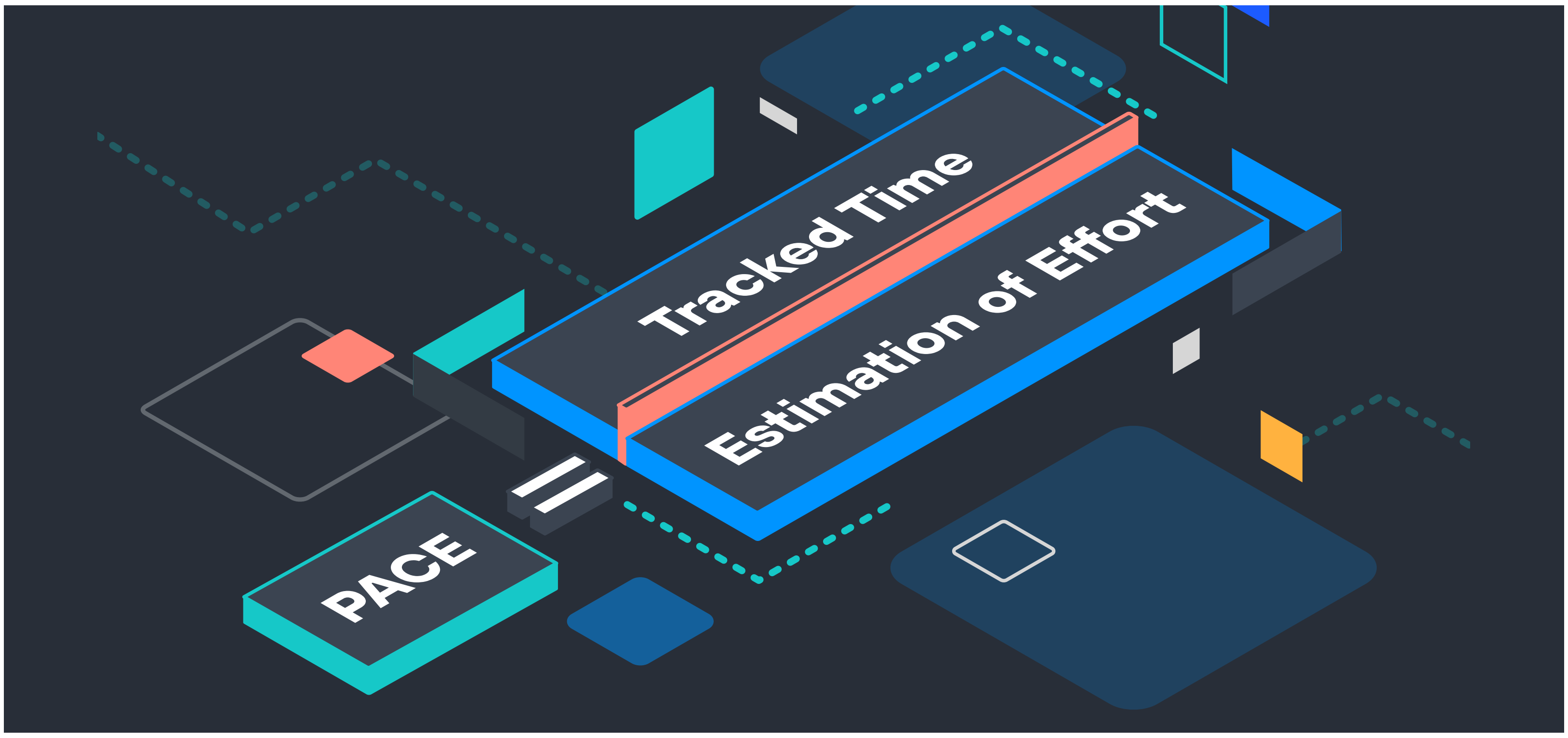
Deployment

Now that time tracking has been implemented and built into your process, it’s time to bring things full circle.

Time tracking deployment is where managers really have the opportunity to lead by example — to prove that time tracking isn’t a control mechanism for their own use, but a valuable tool that can help the entire team get better.

.

KNOW THE RELATIVE NATURE OF PACE



Source: https://www.7pace.com/blog/software-development-pace-calculation

One useful metric your team can derive from time tracking data is their pace, which is the value of tracked time divided by estimation of effort (or number of story points).

A team's average pace can provide a solid baseline for scoping future work, but it's important to remember that pace is relative, and can vary from project to project.

A better way to approach your team's pace is by asking if the team is happy with their average pace. If yes, it's a "good" pace for the team. If no, that's not a reflection of the team or the value of their work — it just means there's a problem to be solved.

NEVER USE TIMEKEEPING AS A CONTROL MECHANISM

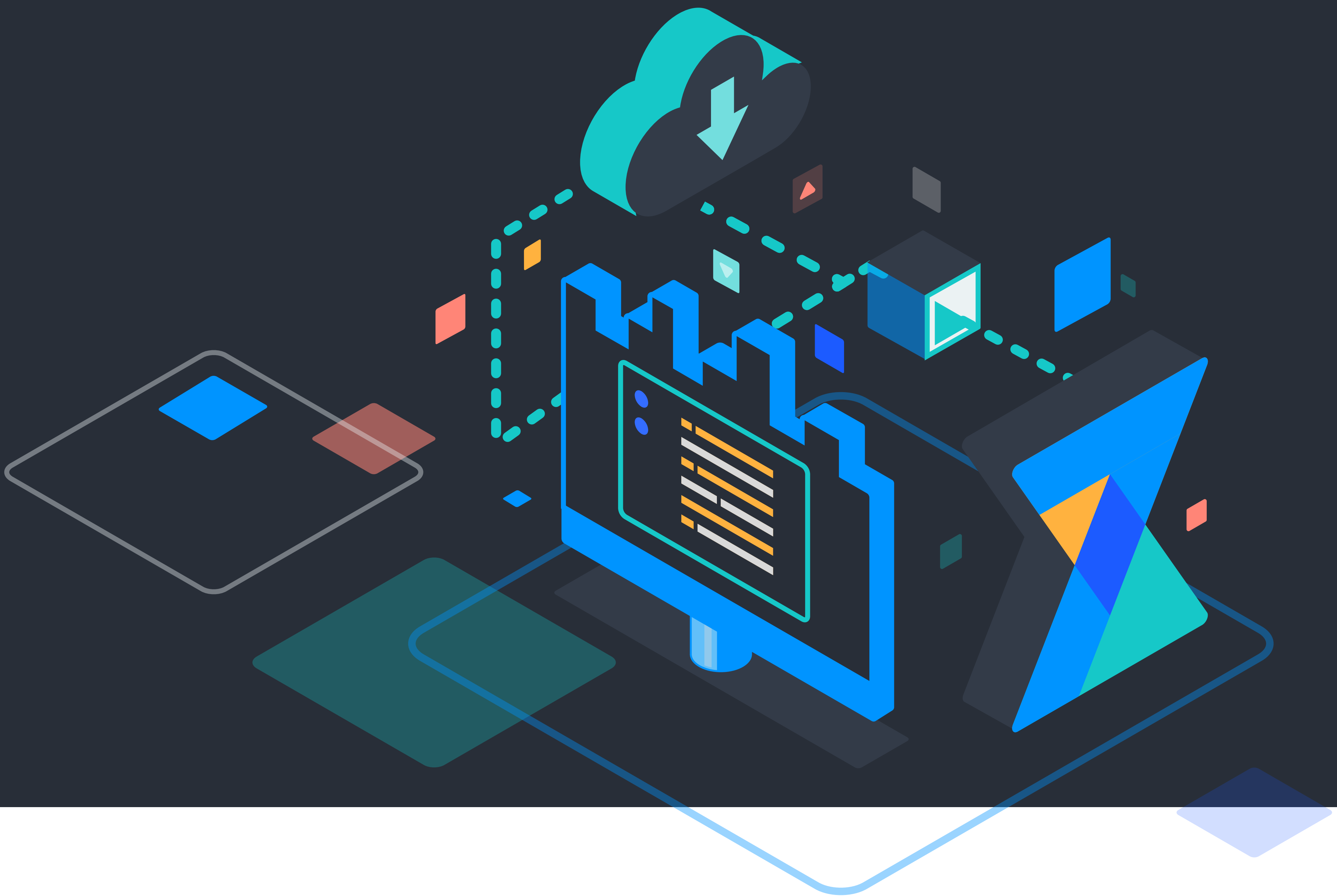
We've said it before and we'll say it again: Using timekeeping as a means to control developers and their work will never be effective.

The same goes for setting quotas or having unreasonable expectations for "time spent coding."

If time tracking means managers will pore over time sheets and scrutinize how developers are spending their time, go back to the beginning of this book and start over, because you haven't successfully reframed your thinking around time tracking for software teams.

USE HISTORICAL DATA TO INFORM, NOT ENFORCE

Instead, time tracking data is a tool, just like any other tool your team uses to measure goals, progress, and improvement. It can help them identify blockers and slowdowns, and it can help them increase their efficiency — but only if they're using the data to inform themselves and their own growth. Never because their managers are using time cards like they're carrots and sticks.



What Makes 7pace Timetracker Different?

"Our teams have used Timetracker for a while as they make it very easy to use. The rare occasion of needing support is always responded to very quickly and the service is great. Viewing the data in the application is easy, but if you also need to combine it with other data for analysis, you can always export the data to put it all together. Excellent product and service."

STEVE CHRISTIAN, 7PACE TIMETRACKER REVIEWER

"7pace Timetracker is inside of the system where developers do most of their planning and collaboration. Time tracking is literally inside their tasks: Just open the task, add time, done.

Managers like 7pace Timetracker because they see overviews on a higher level. Everyone is happy."

MAXIM LUTSAN, 7PACE CTO

"We have been using this Time Tracking application for the past 3 years and we really like it! The integration was easy and all the tools are simple to use.

The support team is incredibly responsive, anytime I have a question/problem I can count on them to help me. They are super friendly and really efficient!
Thank you!"

MARIE-NANCY CROS, 7PACE TIMETRACKER REVIEWER

“On top of integration, 7pace Timetracker has rich reporting tools. For managers, there are predefined reports and tools for complex overview of the team‘s performance. For individuals, 7pace Timetracker provides a rich overview of each team member’s productivity and summary of their work for long periods of time.”

EUGENE KOLOMYTSEV, 7PACE SOFTWARE ENGINEER

“We used the time tracker for 1 year. It was easy to use and allowed much more detailed tracking of activity, and provided excellent historical data. It integrated seamlessly into TFS.

The support team was great, and quickly resolved any issues we had.

We unfortunately had to cancel our subscription due to our organization restructuring, but I would highly recommend the Timetracker software.”

ZACHARY PREZIOSO, 7PACE TIMETRACKER REVIEWER

You’ve Reframed the Process. Now You Just Need the Right Tool.

Ready to get all the value from time tracking — with none of the wasted time?

What you need now is the right tool.

7pace Timetracker is built for development teams who want the data and value time tracing provides, but with integration that makes time tracking easy and seamless. It's fully integrated time tracking for Azure DevOps, TFS, VSTS, and Github that stays out of the way so you can do your work.

Ready to track time without the timesuck? Learn more at 7pace.com.