waydev

Cycle time: How to Optimize the Key Metric to Accelerate Software Delivery

In engineering organizations, growth comes with an increase in complexity of fast and reliable software delivery. And behind this rapid ascension to greatness lies a (not so) hidden weapon: optimized cycle time.

From developers and engineering managers, to C-level executives, almost everyone working in the software development business wants to ship faster, build better code, and deliver more value to end users.

But your job, as an Engineering Team Leader, means figuring out how to achieve these three specific objectives: gain visibility into potential roadblocks, reduce time spent creating reports, and increase delivery velocity. It’s a tough job, but someone’s gotta do it, and do it well.

Great team leaders help their development teams deliver high quality software by constantly improving *how* they work. Having a better overview of how to reduce cycle time will help you do just that. An optimized cycle time means you don’t have to choose between Time to Market, speed, or quality—instead, pick how you want to allot resources between the three.

With so many benefits on the line, you’re bound to have a lot of questions. *How do you calculate cycle time? Can you automate cycle time analysis? And what are the advantages of reducing cycle time?*

At Waydev, we think cycle time is the key to dedicating more time to innovations that deliver results, while making your team happier and more performant, and we're here to answer all of these questions and more.

This step-by-step guide will tell you exactly what to measure, why it matters, and how to fix it.

At the end of it, if you want to learn more about how we help engineering leaders drive their teams' productivity up, you'll be able to schedule a demo call and we'll show you how to use Waydev to optimize your own cycle time using automation. Let's dive in.

What does cycle time mean in 2021?

In software development, cycle time is a metric that measures process speed. In other words, it's the speed of your development time, how fast you can deliver a feature to the customer, from the moment work has started to the moment it's been delivered. At Waydev, we measure cycle time from the first commit up until that feature gets deployed to production and is made available to users.

The cycle time metric is an indicator of your organization's development velocity. It's all about the speed and quality of the execution.

Like most software companies out there, you're probably using a set of standard metrics to measure your development pipeline. These KPIs help you understand things like your current burn rate, bugs, throughput, and return on investment.

But without something to tie each of these valuable measurements together, your map to success is incomplete. A cycle time analysis lets you connect the dots between your KPIs and optimize your production process in order to stand out from the competition.

We know what you're thinking: where was this *years ago*? The truth is cycle time has been an essential metric for engineering organizations since the very beginnings of the Agile approach to software development.

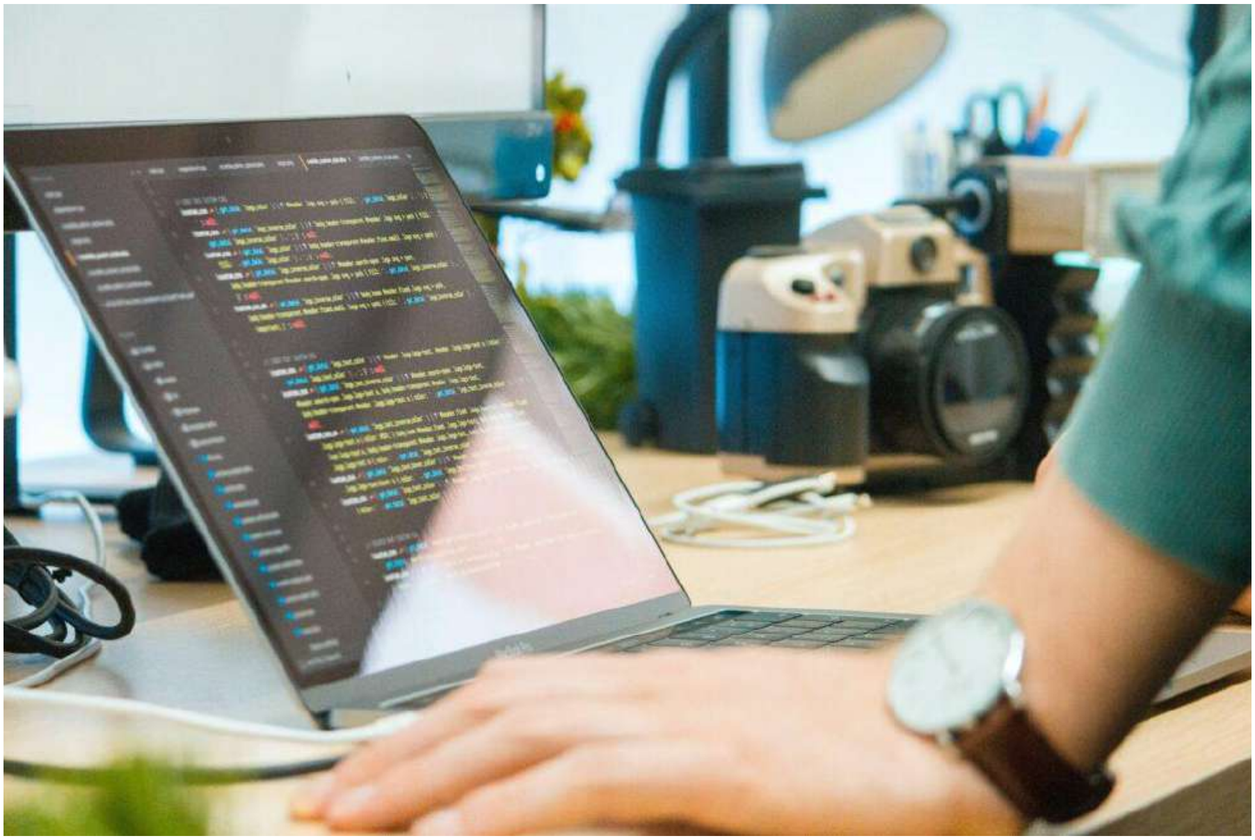
However, along with a recent change in perspective which shifts focus from basic productivity to impact and developer performance, there's been an epiphany into what an optimized cycle time can truly do for development teams.

In 2021, your main goal should be to reduce your production cycle time and our features at Waydev are ready to deliver the right Git analytics to do it.

An optimized cycle time benefits everyone—from engineers and managers to end users. It means that engineers no longer feel satisfied and less stuck, managers get to do their job better and faster and products get to market faster, where they receive user feedback which in turn, makes users happier.

How exactly does that happen? Let’s find out.

The importance of cycle time and the benefits of reducing it for software development



The importance of cycle time has been tried and tested by countless engineers and team leaders looking to increase velocity and transform their organization into a performant, well-oiled machine.

Reducing cycle time is one of the few things that will actually have an impact on your software development times and process. Amongst other important benefits in reducing it there are:

1. Your engineers will feel ownership from shipping frequently and your product team will be able to design better features, with more confidence in their execution.
2. Shorter cycle times make your development team's progress faster and more efficient. And that translates into the holy triad of software production—a reduced time to market, a better product, and more satisfied customers.
3. Your executive team will be able to focus on growth and expansion, knowing they have a highly functional development team to back it up.

How optimizing cycle times helps team leaders

Any software manager or team leader who wants to help their team see their work out in the world quicker—and at higher quality—should start by looking at their process cycle time.

A cycle time analysis gives you a clear picture of what's working (and what's not) in your development pipeline. When done right, it can indicate how efficient your software development team really is, while also letting you:

- identify bottlenecks and proactively resolve them
- ensure your teams ship faster
- enable your developers to build better code.

This type of quantitative metrics will shine a light on what you've known through intuition gathered from years on the job: faster ship cycles lead to more reliable product delivery and a better work environment.

How is cycle time measured?

Now that you've got a better look at the *why*, let's take a closer look at *how* to measure cycle time.

We've noticed that, at times, the cycle time definition can be easily confused with lead time. Before we go on, here's a detailed article to help you understand the difference between lead time vs cycle time.

If a cycle time formula is what you're looking for, try calculating the difference of time between the start and end date of the development process. Or, as the definition states, the difference in time between the developer's start date to the ship date, in no. of days. This will give you a rough estimate on how to calculate cycle time.

At Waydev, we believe that the current Agile methodology needs some tweaking. Our goal is to shift focus, from unending planning and documenting their software development cycles to a more flexible approach that uses real time data.

So, in an effort to make the cycle time picture clearer and easier to read, we’ve broken down the cycle time metric into the four stages of the software development process. And it worked.

1. CODING: Time to Open – The activities that take place before a Pull Request (PR) is opened have a higher impact on cycle time than those that take place after. That’s why optimizations at this stage have the greatest impact on overall cycle time.
2. PICKUP: Time to First Review – This stage deals with the first bottleneck in your PR workflow. Time to First Review is often overlooked because it doesn’t represent individual work, nor does it reflect the efficiency of a Code Review process. Instead, it represents the effectiveness of the first handoff in your PR workflow.
3. REVIEW: Time to Approve – Code Reviews are best suited for improving code maintainability and increasing shared understanding throughout the team. That doesn’t mean they can’t be improved upon themselves.
4. DEPLOY: Time to Deploy – At this stage, you’re deploying working software continuously to users to mitigate risks, respond to change quickly, and get the best version of your product into the hands of customers.

Cycle time may be a great metric to gauge success, but it isn’t a final diagnostic. To understand why your cycle time is high or low, you’ll want to look at its four constituents and see what optimizations you can implement at each stage.

A step-by-step guide to reducing and optimizing cycle time in software development

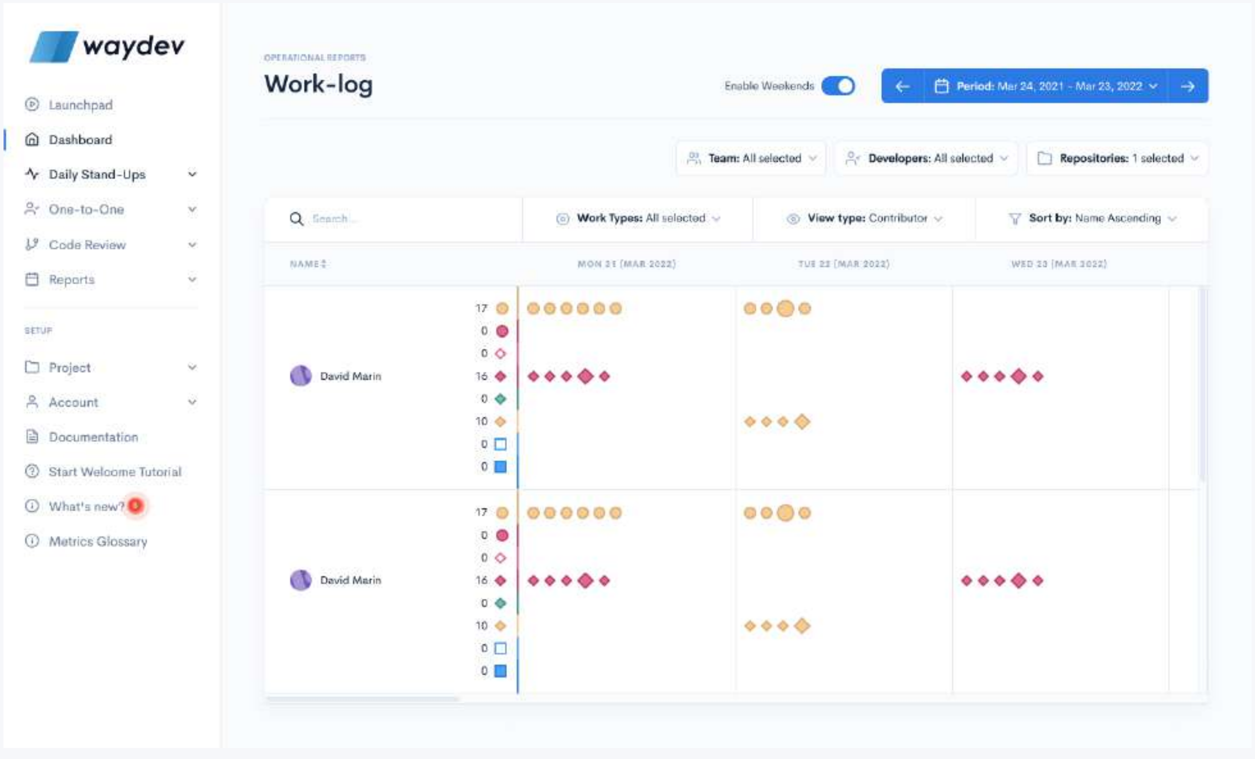
CODING: Time to Open

One of the most important principles of Continuous Delivery is that keeping batch sizes small can have significant effects on the entire team's velocity. That's why optimizing the Time to Open stage, which corresponds to coding time, and is the time elapsed from the first commit to creating a PR, is so important. It can have downstream effects that impact your development roadmap and business goals.

Get your team into the habit of opening PRs early and often. A good way to do this is to reduce the size of your PRs, by zooming into your team's output and work habits and looking at commit and Pull Request activity across the organization, by team and by individual. This lets you avoid batch sizes that are too large, an unnecessary increase in code churn, or a lot of multi-tasking and task-switching.

Speaking of task-switching, to really keep it at a minimum, try to limit unclear or conflicting priorities. These are bound to make your engineers constantly switch back and forth between projects, which is the last thing you want to deal with.

Last but not least, keeping an eye on code churn can help identify trends and help you as a manager do your job better, by maintaining deadlines and helping team members become more productive.



Period: Mar 24, 2021 - Mar 23, 2022

Team: All selected

Developers: All selected

Repositories: 1 selected

Work Types: All selected

View type: Contributor

Sort by: Name Ascending

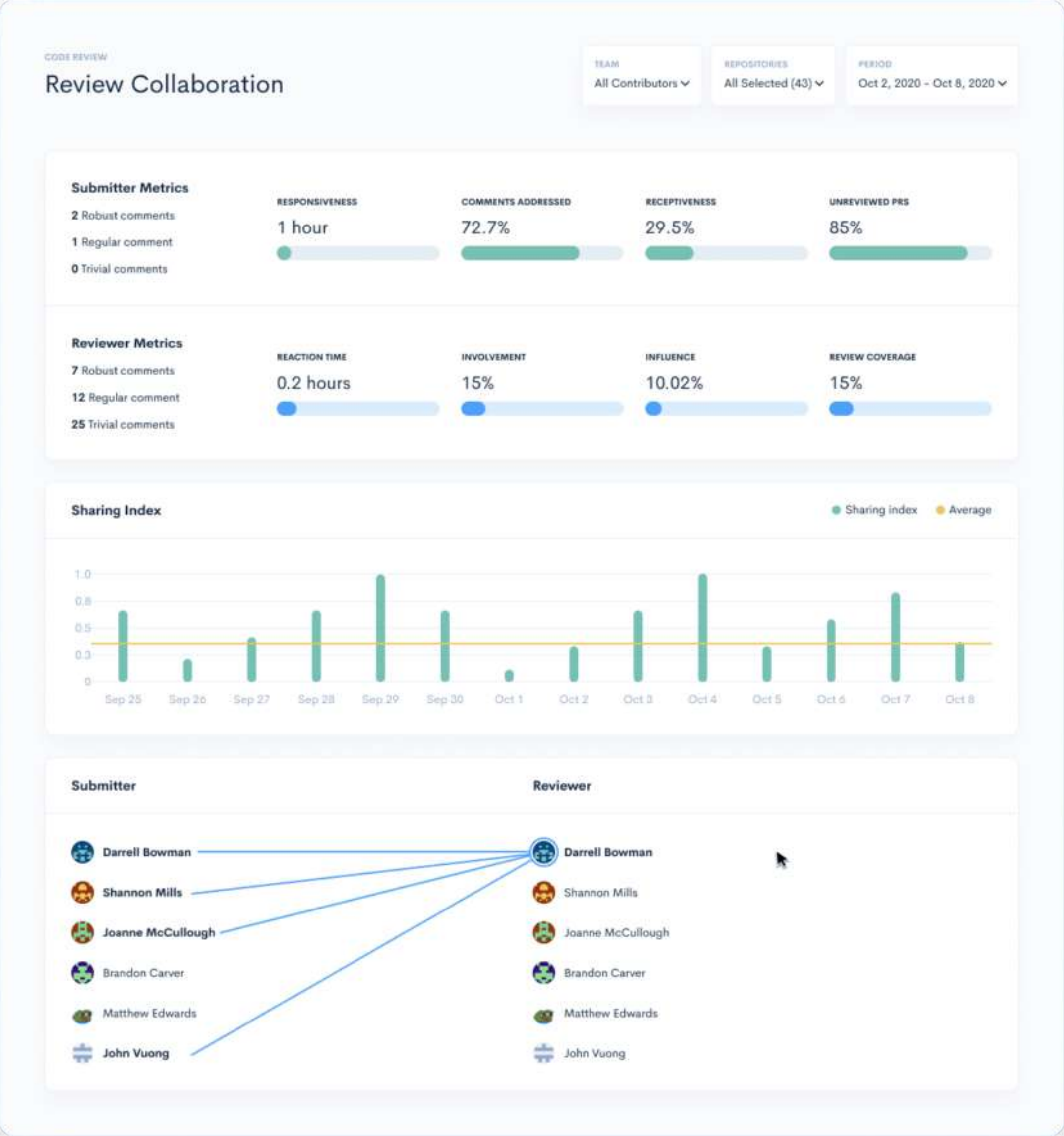
The Waydev Work Log features helps managers gain visibility into their onboarding process, understand where their team is focusing their attention, or notice when there's an unexpected spike in activity.

PICKUP: Time to First Review

This pickup metric deals with how fast reviewers pick up their peers' PRs for review. Essentially, it measures the time between when a PR is opened and the first time an engineer reviews that PR. What you want to do is optimize them both.

Why, you ask? Because not picking up PR for review fast enough means your team will be dealing with bottleneck after bottleneck, which limits your team's speed and increases your average cycle time.

Your goal here is to limit multi-tasking and minimize the time a PR is left waiting for review. To do that, you can distribute code reviews across your team and increase review collaboration. This will help you effectively communicate the healthy tension between speed and thoroughness in code review. It will also help improve your code review workflow and see how well reviews are prioritized across the organization.



CODE REVIEW

Review Collaboration

TEAM: All Contributors

REPOSITORIES: All Selected (43)

PERIOD: Oct 2, 2020 - Oct 8, 2020

Submitter Metrics

2 Robust comments
1 Regular comment
0 Trivial comments

RESPONSIVENESS	COMMENTS ADDRESSED	RECEPTIVENESS	UNREVIEWED PRS
1 hour	72.7%	29.5%	85%

Reviewer Metrics

7 Robust comments
12 Regular comment
25 Trivial comments

REACTION TIME	INVOLVEMENT	INFLUENCE	REVIEW COVERAGE
0.2 hours	15%	10.02%	15%

Sharing Index

Submitter	Reviewer
Darrell Bowman	**Darrell Bowman**
Shannon Mills	Shannon Mills
Joanne McCullough	Joanne McCullough
Brandon Carver	Brandon Carver
Matthew Edwards	Matthew Edwards
John Vuong	John Vuong

The Waydev Review Collaboration feature offers a unified view of submitter and reviewer metrics of the PR process.

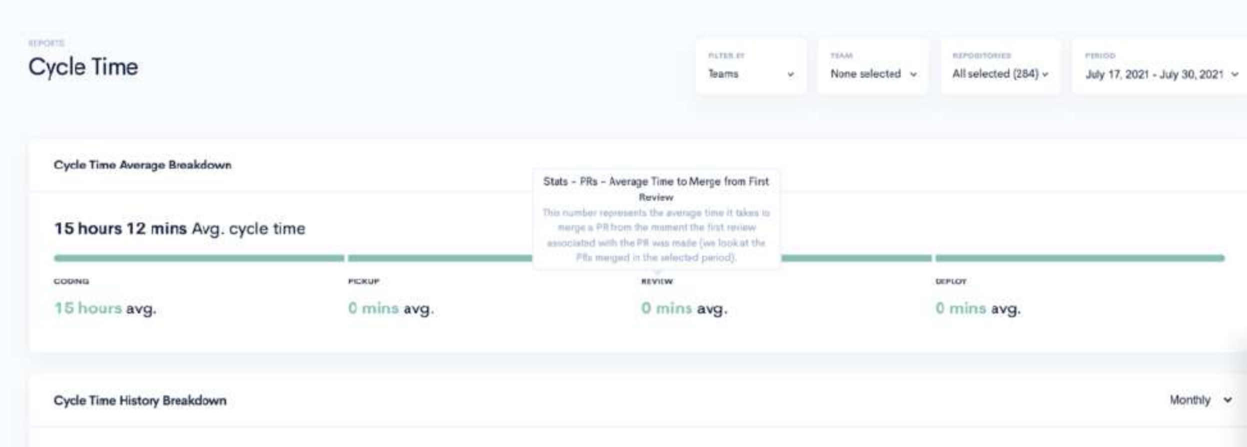
REVIEW: Time to Merge

The Time to Merge from First Review looks at how fast submitters incorporate feedback from their peers in code review. It's the time from a PR's first review to that PR being merged and it can greatly affect your bottom line.

The key here is finding a balance for code reviews that are both useful and efficient. Find the review coverage sweet spot. This is a thoroughness indicator that can tell you how effective your code review process is. Code reviews shouldn't be too thorough, or too superficial. Getting reviews and approvals right can decrease frustration for your team and help them use their engineering time in the most effective way.

To optimize this stage, try not to downplay the importance of code reviews, but to make the process of reviewing as efficient as possible, while still getting great reviews out of it. Make sure your reviews are both succinct and impactful. Strive for more valuable reviews that have clear comments that yield real action.

In the end, try to limit review cycles with better internal alignment. Make sure everyone agrees with what the criteria for a good code review is: does it make the work better?



REPORTS

Cycle Time

FILTER BY: Teams

TEAM: None selected

REPOSITORIES: All selected (284)

PERIOD: July 17, 2021 - July 30, 2021

Cycle Time Average Breakdown

15 hours 12 mins Avg. cycle time

Stats - PRs - Average Time to Merge from First Review

This number represents the average time it takes to merge a PR from the moment the first review associated with the PR was made (we look at the PRs merged in the selected period).

CODING: 15 hours avg.

PICKUP: 0 mins avg.

REVIEW: 0 mins avg.

DEPLOY: 0 mins avg.

Cycle Time History Breakdown

Monthly

An example of cycle time stats from Waydev's Cycle Time Average Breakdown report

DEPLOY: Time to Deploy

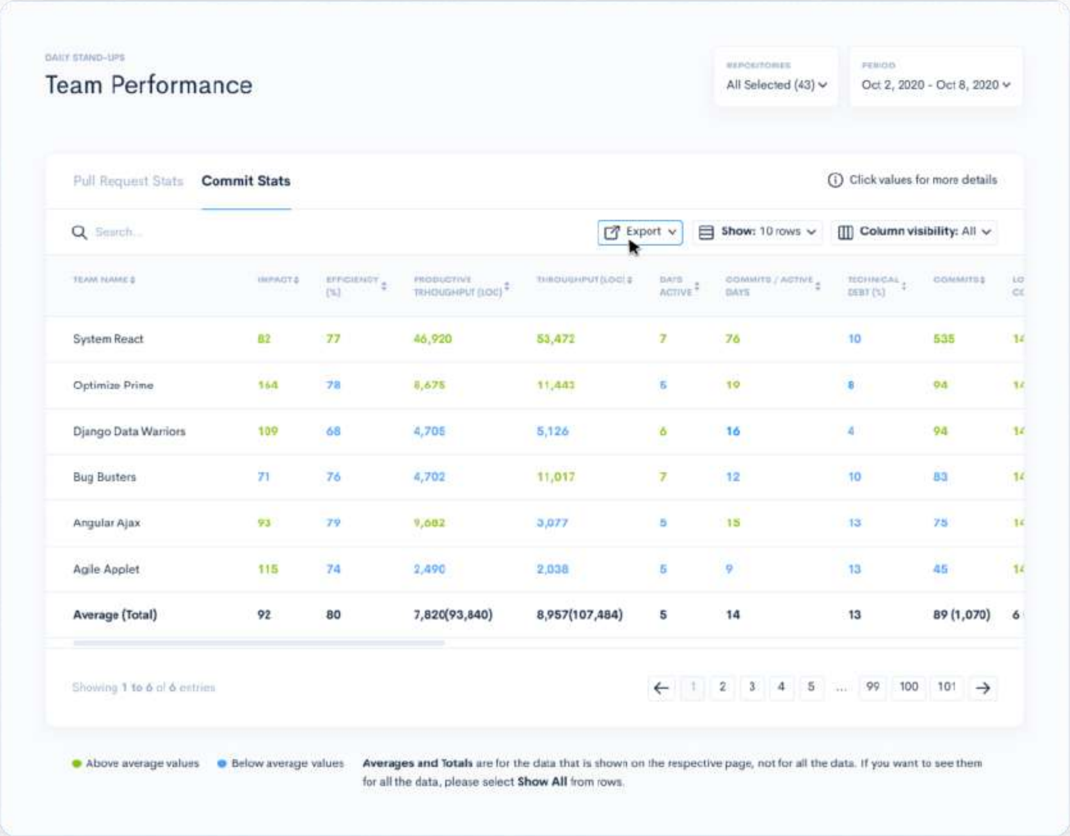
The final step in your cycle time optimization journey is the Time to Deploy stage. This metric is an indicator for how fast code gets deployed into production, and is the time between when a PR is merged to when it gets released into production.

Getting batch size and the review and approval process right can do wonders for your team's performance. But unless you're actually shipping to production, you're not practicing Continuous Delivery or reaping all its benefits. A healthy culture of Continuous Delivery can reduce the behavioral impediments to faster and more frequent deployments, such as lack of confidence and frustrations with the process.

Deploying continuously is a great way of shortening your feedback loops between your engineers and end users. Ultimately, the goal is to get the best version of your product into the hands of customers.

A robust technical tool-kit that’s tuned to your internal processes is key. Make sure to check your continuous integration suite, deployment pipeline, and existing monitoring production system to see if there are any areas that could use an upgrade.

You can also use automation to make sure every person on your development team spends their time strategically. This helps you avoid manual blocks in deployment.



DAILY STAND-UPS

Team Performance

REPOSITORIES All Selected (43)

PERIOD Oct 2, 2020 - Oct 8, 2020

TEAM NAME	IMPACT	EFFICIENCY (%)	PRODUCTIVE THROUGHPUT (LOC)	THROUGHPUT (LOC)	DAYS ACTIVE	COMMITS / ACTIVE DAYS	TECHNICAL DEBT (%)	COMMITS	LO CC
System React	82	77	46,920	53,472	7	76	10	535	14
Optimize Prime	164	78	8,675	11,443	5	19	8	94	14
Django Data Warriors	109	68	4,705	5,126	6	16	4	94	14
Bug Busters	71	76	4,702	11,017	7	12	10	83	14
Angular Ajax	93	79	9,682	3,077	5	15	13	75	14
Agile Applet	115	74	2,490	2,038	5	9	13	45	14
Average (Total)	**92**	**80**	**7,820(93,840)**	**8,957(107,484)**	**5**	**14**	**13**	**89 (1,070)**	**6**

Showing 1 to 6 of 6 entries

🟢 Above average values 🔵 Below average values **Averages and Totals** are for the data that is shown on the respective page, not for all the data. If you want to see them for all the data, please select **Show All** from rows.

Waydev's Daily Update feature is most commonly used in daily standups and can quickly help your team surface discussion around blockers in the code they are working in.

Leverage automation in cycle development using Waydev

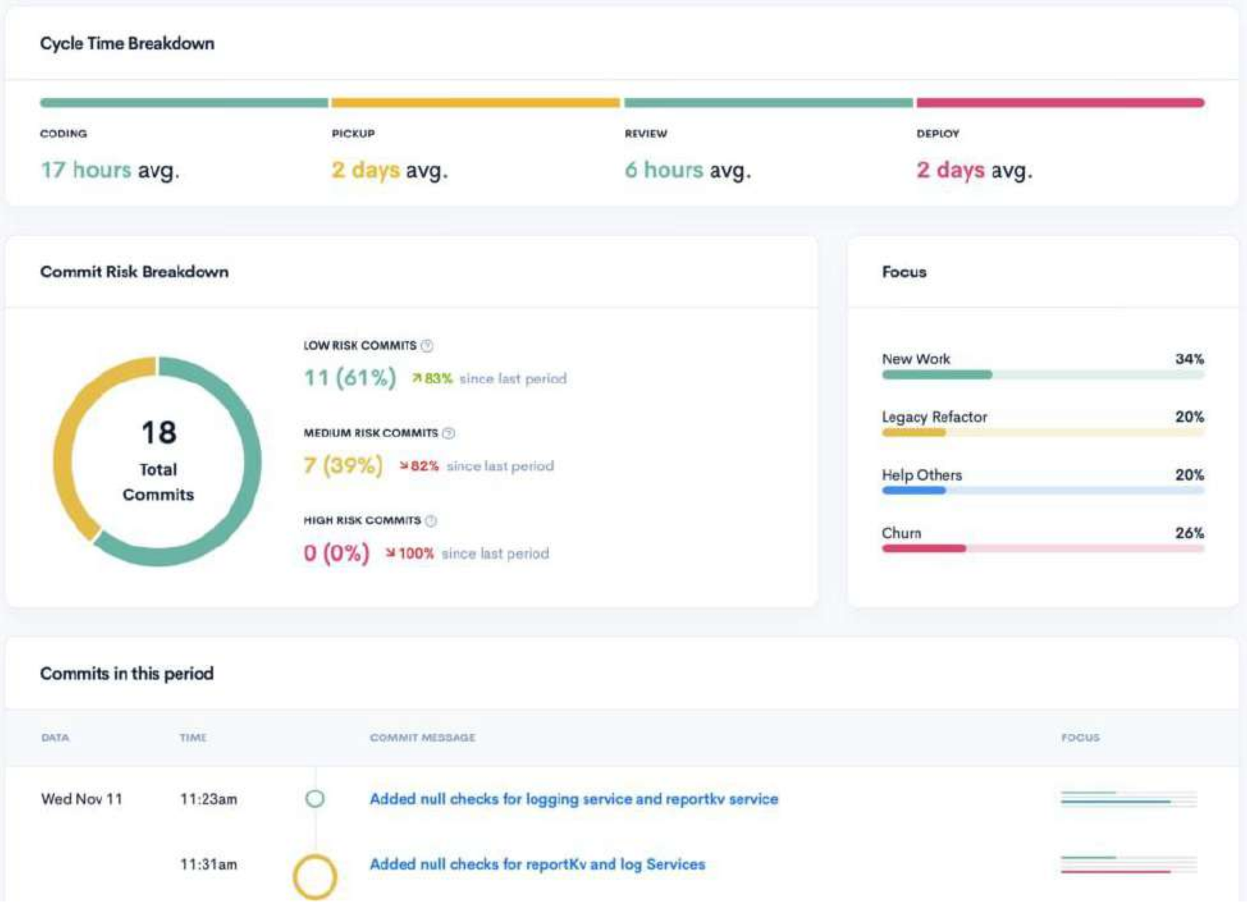
Reducing cycle time isn't that difficult if you've got the right tools and the discipline to do it. The hard part comes when you realize you need an automated way of doing the measuring, since a software development team like yours usually works on a large number of issues and tasks.

The alternative? Crunching numbers, estimating and relying on manual input, spending hours scouring GitHub, scanning Jira tickets, or pinging developers, making sure you have enough valid data to be useful.

With tools like Waydev, there's no reason to spend valuable time doing this. Instead, our automated cycle time measuring and reporting gives you all the data you need to optimize processes and improve productivity.

Engineering Managers have instant access to their team's progress without anybody's manual input. Waydev spots impediments, inefficiencies and turns this data into insights that drive performance and output.

That way, you can enjoy all the cycle time reduction benefits, while also keeping up with your development roadmap and business objectives.



Cycle Time Breakdown

CODING	PICKUP	REVIEW	DEPLOY
17 hours avg.	2 days avg.	6 hours avg.	2 days avg.

Commit Risk Breakdown

18 Total Commits

LOW RISK COMMITS
11 (61%) ↗83% since last period

MEDIUM RISK COMMITS
7 (39%) ↘82% since last period

HIGH RISK COMMITS
0 (0%) ↘100% since last period

Focus

New Work	34%
Legacy Refactor	20%
Help Others	20%
Churn	26%

Commits in this period

DATA	TIME	COMMIT MESSAGE	FOCUS
Wed Nov 11	11:23am	Added null checks for logging service and reportkv service	
	11:31am	Added null checks for reportKv and log Services	

This type of cycle time visualization is a key component of the Agile Data-Driven approach to Scrum that aims to bring visibility into engineering teams' development workflow.

Software development analytics tools like Waydev help you automate tracking across a range of performance metrics—including cycle time.

Our data-backed approach can help you identify factors contributing to inefficiency and measure the effectiveness of code reviews and training programs. All so you can focus on what really matters: transforming your organization into the success it was meant to be.

You can optimize your cycle time by using Waydev to:

- Track, measure and report on cycle time, making it more effective
- Benchmark and visualize cycle times
- Accelerate velocity and increase speed of delivery
- Remove roadblocks from the development process
- Reduce sources of frustration for developers

Ready to see how your team's work is progressing at a glance? Book a demo and see for yourself how reducing cycle time can boost engineering productivity and efficiency—we can't wait to see what your team achieves.

Visit waydev.co to learn more

waydev

TRUSTED BY **FORTUNE 500** COMPANIES

genpact Capgemini Carrier TATA HEALTH NOWCOM LIDL B2B BANK

BlueCross BlueShield rey SOVOS WOM Yellowbrick PLAID northpass

Leader SPRING 2022 Users Love Us Vanta SOC 3 AICPA SOC