waydev

← **Back To All**

9-Step Guide: How to Plan Effective Software Development Projects

JULY 14TH, 2021

Software development is an extremely complex process. And way too often with tech industries today, the work is needed... yesterday.

It's something that happens whether you build software for clients around the world or develop your own software products.

When the client is non-technical, you may find it hard to explain why things don't go as planned, or motivate why you need more resources. And the client may get confused as to why a project can't be delivered on time and on budget, despite your developers' best efforts.

When you work on your own products, you may find it imperative to reduce cycle time, increase predictability and minimize frustrations among your developers to deliver projects faster.

That's why planning is so vital (otherwise, it'll look like this project timeline...).

Privacy - Terms

Because no matter how simple or complex the project is, you need **an effective project plan.** It's the solution for clear structure, documentation, and procedures to complete tasks efficiently, deliver quality software, and meet realistic deadlines.

But how do you plan effective software development projects nowadays?

How can you deliver better software, faster? Here's **your complete guide on planning in 2021, with the help of development analytics.**
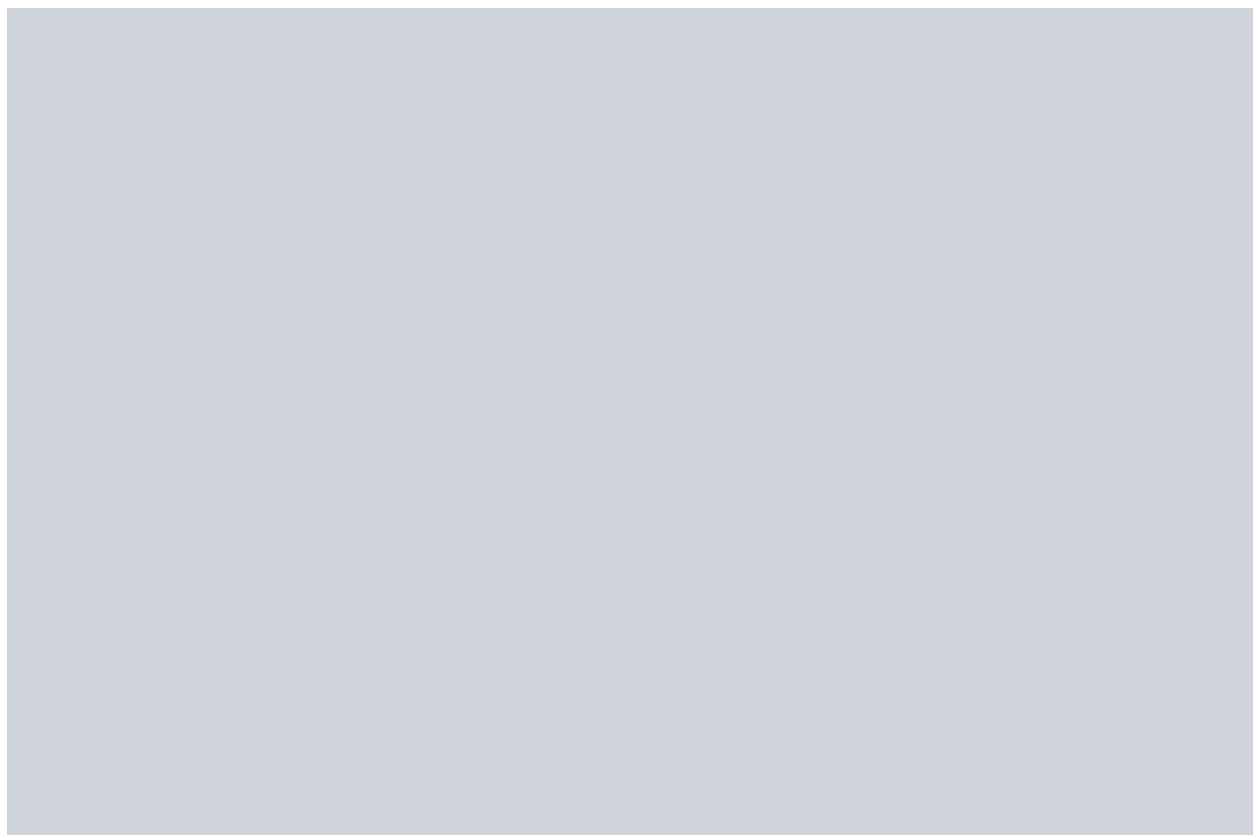
What is Software Development Planning?

A software development plan refers to the process of **organized management** for a project. It describes the activities and the resources needed to complete it successfully.

Software development planning encompasses **the timeline of a development project.** The plan describes all the phases of the process

Privacy - Terms

analysis and planning, ideation, design and development, deployment, production and launch, and maintenance.

Knowing **how to plan effective software development projects** means to document and cover everything from the concept, project description and requirements, to a clear schedule of the timeline, project monitoring, desired deliverables and potential revisions, assessment of costs and resources, risk management and anything in between.

That is why **the quality of the project largely depends on how well the software project plan is built.** It also allows for the optimal allocation of resources exactly where they're needed. Without planning, you'd risk having too many hurdles to tackle during the process, which would lead to delays, additional costs, customer dissatisfaction, low morale and low team productivity.

As a result, the effective planning of software development projects is essential to everyone – product owners, stakeholders, managers, development teams.

The planning is ideally done by a project manager together with the responsible team(s), in sync with the stakeholders and other departments that have a role in the project's success – such as testing or QA teams, or marketing specialists.

The benefits of modern software projects planning are huge: not only you'd have a clear process in place for everyone, but you'd also develop more time- and cost-efficient, reliable software in a much less stressful way, for both the provider and the client teams.

Among other benefits of planning and following the software development life cycle:

- Everyone "speaks" and understands the same language
- It sets clear workflows and communication channels between the parts involved
- It sets clear roles and responsibilities for everyone on the team
- Everyone understands the status and the definition of "done" for each step

Privacy - Terms

- It sets an initial agreement on how work would be handled post-production (bug fixing, new features, updates)

Setting processes to deliver digital projects started a while ago. In the nineties, there was the Waterfall model, with long deadlines and Gantt charts. We got Agile in the 2000s, with sprints and manual reports.

Now, you have **the Agile data-driven method** that measures **team performance automatically** and gives you real-time data on the development work, to make more data-driven decisions.

It's exactly what Waydev, the #1 development analytics platform, helps you with.

Waydev gives you practical and objective insights on engineering work, helping you visualize how many resources are spent on unplanned work, bug fixing and other initiatives. You can better understand the financial impact and project costs and you'll be able to use custom, automatic reports to answer complex questions about what's happening in the development process.

Try Waydev with your team for free.

Privacy - Terms
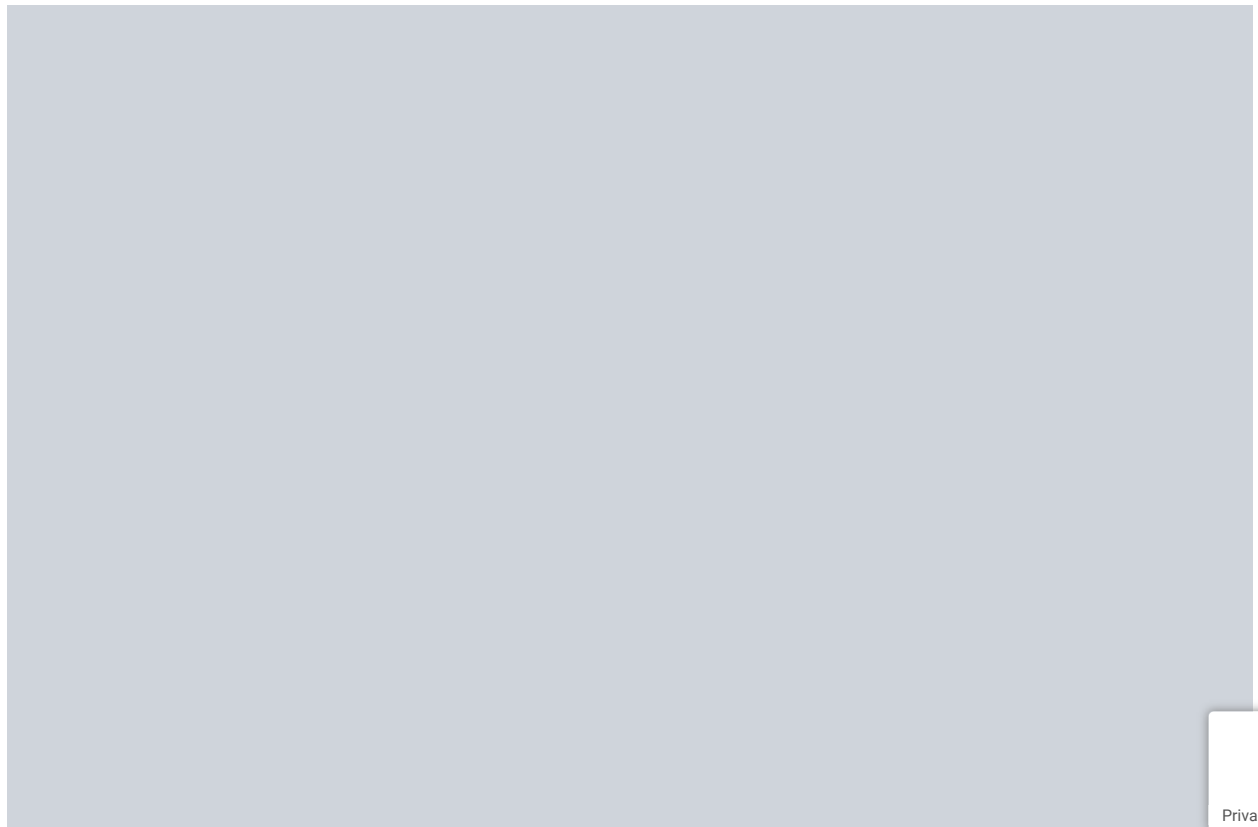
What Are the Goals of Planning a Software Project?

We can say that **the main goal of planning is organizing the process to make sure client and user requirements are met, and preventing issues and miscommunication.** How it's done will greatly influence the success or the failure of the project.

If we're to dive a bit deeper into the **purposes of planning a software development project**, it would be about **having a clear overview over:**

- The functional purpose of the software project
- The tasks and subtasks to complete each functional requirement
- The list of priorities regarding the features development
- The people involved and their clear roles and responsibilities
- Resources and realistic estimates
- The performance and process metrics that will measure efficiency and quality of work.

Among **the most important goals and requirements needed for effective planning** are:

- Having a clear and concise understanding of the goals and specs
- Finding time and cost efficient solutions to completing the project
- Having a complete documentation of the project roadmap
- Adherence to quality development standards,
- Delivering reliable, high-quality products to end users
- Ensuring the flexibility needed to accommodate changes and iterations without affecting the process.

As you can see, there are both project goals and business goals to achieve.

Having a well-thought-out development plan also helps when there's a problem or a roadblock. By following the process, it's easier to spot and clear them without affecting the timeline.

Privacy - Terms

[Check out how Waydev works](): our development analytics platform analyzes your codebase, PRs, tickets, and CI/CD activities, to help you bring out the best in your engineering teams' work.
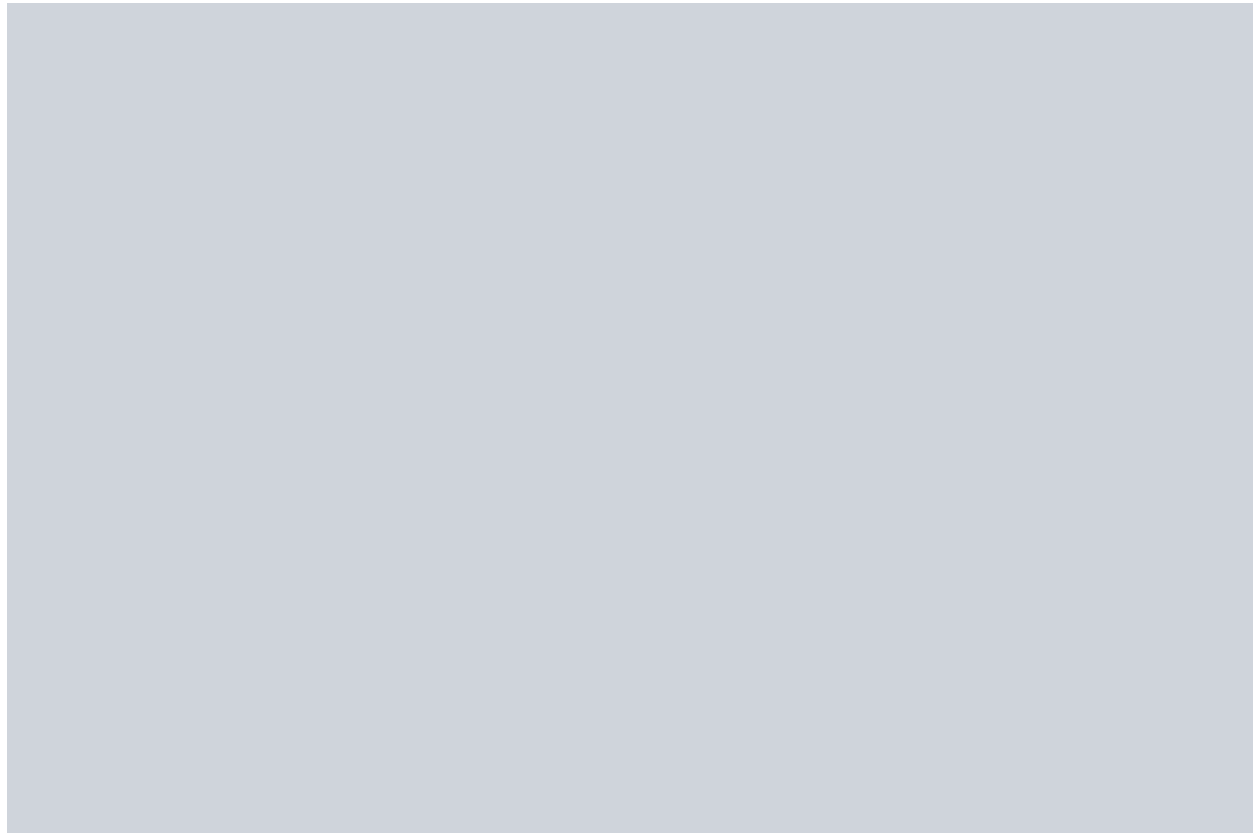
What is Included in a Software Development Plan?

Because software development is a complex technical process that requires a lot of steps, having a project plan is extremely helpful **to visualize and follow each step of its development,** according to a timeline for the team's work.

Fortunately, there are many tools – such as development and project analytics, as well as project management apps – to choose from. This choice usually depends on the size of the engineering team and the complexity of the software project.

What is generally included in a software development plan are the following stages:

Privacy - Terms

- **Planning**: when you define the project and its objectives;

- **Scoping**: when you analyze and set:
  - tasks and subtasks as specific as possible
  - Realistic time and cost estimates together with the developers – but account for unexpected delays: holidays, sick days, meetings
  - Milestones and their indicative dates (to know how the project is going)

- **Implementing**: when you develop and deploy the software.

Since data is your best friend in software development, it's always a great idea to use historical data and check your past processes – with Waydev's Project Timeline, it's possible. See whether you were inclined to overscore or underscore when planning a software project (although it's usually the latter).

Now, let's dive into what matters most – learn **how to plan effective software development projects**, by making sure you have these important steps (or phases) on your checklist.

Privacy - Terms
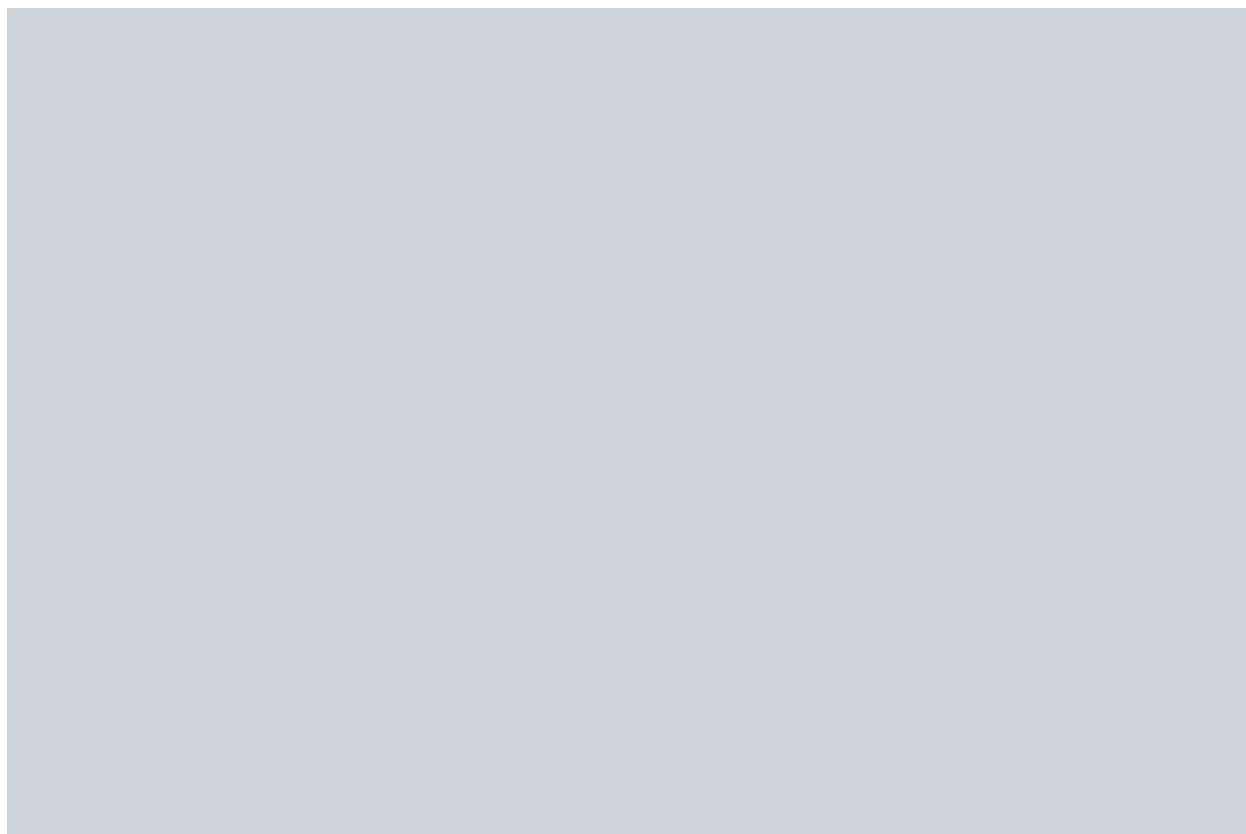
9-Step Planning Guide for Software Development Projects

As a tech lead or software engineering manager, you're already familiar with all these different stages of the **software development life cycle.** That's pretty much the same steps as those for any software product, no matter the chosen process.

After all, you're responsible for your team's performance, the quality and efficiency of the delivered software, and the communication with stakeholders and alignment of engineering goals with business initiatives.

So**, what are the steps to start a software project**? The first one – planning everything.

1. Project Analysis

This is the first step of the scoping stage of a software project requested by a customer. It has a lot to do with the feasibility of the project.

This **analysis** part is to make sure you're starting on the right foot, and it means looking into:

- A deeper understanding of the purpose of the project (concept and goals)
- Also, a deeper understanding the needs of the project
- The project specifications.

Most of the time, it takes place at the same time as the requirements and the main planning phase, since they're all connected.

Privacy - Terms

2. Requirements

As you move on to the third step in planning and continue to complete the scope of the project, your development team and you start to **document the technical requirements** needed to successfully deliver the software product.

In this stage, your tasks are to:

- Decide on the team responsible for the development

- Decide on the requirements

- Decide on the milestones that would work as checkpoints for testing and feedback rounds (they answer to: "how's the project going?")

Together with your team, you'll answer specific questions about the technical aspects of the project, such as: what issue does the project solve? Who's the end user? What sort of data is needed? Does it need certain integrations? How is security tackled?

By answering these questions, you will be able to describe and document the technical specifications and requirements, methods of

Privacy - Terms

testing, technology stacks, software and hardware, as well as the type of methodology (you might start sprint planning, if you're using Agile).

Part of this planning stage is also the risk assessment, making sure that you're also understanding the constraints or risks the project might have.
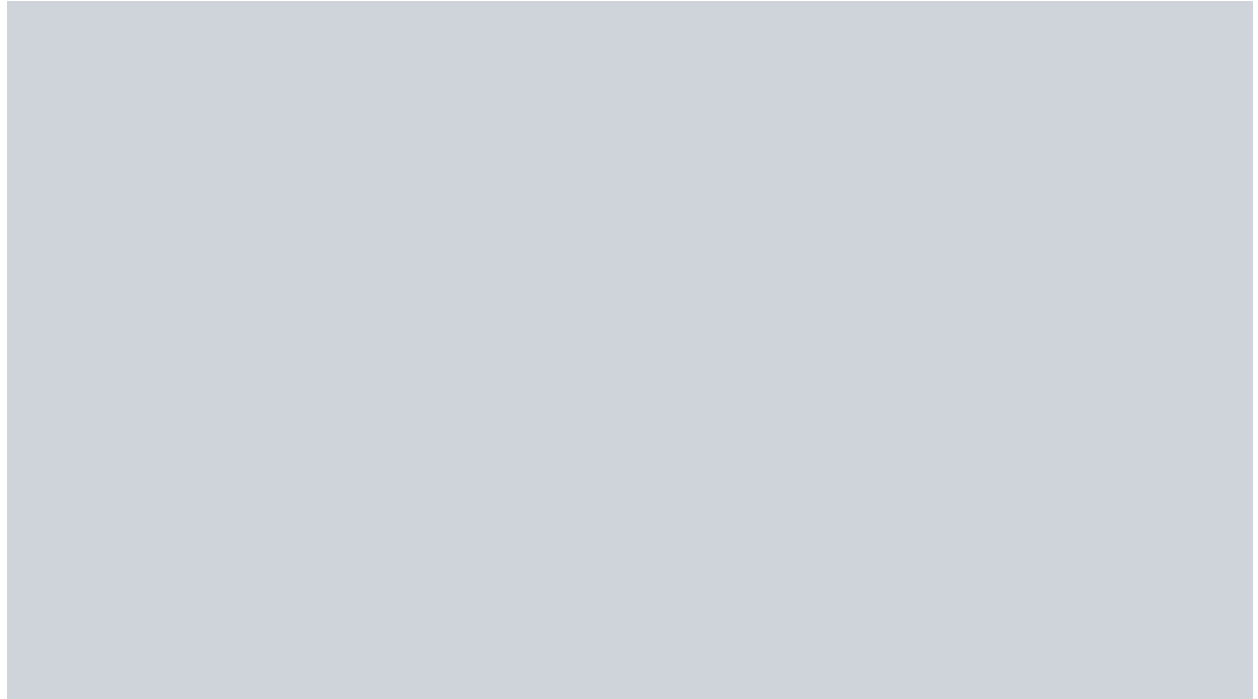
3. Planning

Planning is the most meticulous and critical part of the project. It's also part of the project scope that allows everyone a clear understanding of the timeline from ideation to deployment.

- A clear overview over the timeline ensures a better organization and synchronization of everyone involved.
- By breaking down work into tasks, subtasks and steps you will take and looking at the resources needed, you can provide a clearer estimate of the time and budget.
- You can use a Gantt chart, a list or, better yet, a software tool to visualize this timeline plan and track it smoothly.

If there's an issue, you can pinpoint and solve it more easily than having no planning at all.

Privacy - Terms

At the same time, this step determines the quality of your management – which, in turn, will determine the success of the project. The more **detailed and well-thought the planning** is, the easier it becomes to follow the schedule and prevent too many issues along the process.

As a result, the planning phase is all about researching, deciding and documenting all the aspects: goals, deliverables, tasks, costs, deadlines.

But don't forget to set meaningful engineering KPIs and metrics in place to quantify your projects' success for the concrete targets you set.

To give you a more specific structure, now's the time when you'd have to work on:

- An approximate timeline and main stages of the development
- Tasks and subtasks for each individual who works on the project
- Ways of communicating with the team, the client, stakeholders
- Time and cost estimates (more accurate when the requirements are set and analyzed)
- Release dates and types of releases
- **Documentation of everything** (even though you'll make iterations, keep this intact).

An ideal scenario is to have the software development team working together with the marketing and the testing team to align business and development goals – but that's not a rule. If they're part of an external team, testers, for example, can cover their part at a later stage.

The planning stage also describes the specific stages of the project development, work needed in each stage, and cost estimation.

The timeline should be frequently revisited. Nothing is set in stone when it comes to software development, since it's always a complex process to handle, so adaptability is key.

Waydev is trusted by over 1,000 engineering leaders worldwide. Discover how our complete Git Analytics tool features can also help you to increase velocity and reduce cycle time.

Privacy - Terms

4. Design & Prototyping

The next part of an effective software project plan is conceptual designing. This has nothing to do with the aesthetics, but with the project architecture and functionality.

As a result, it's done by software architects and engineers to make sure the flow, structure and the functionality meet the project requirements. It can include:

- The whole structure and its capabilities
- The functionality of the software solution
- Hardware and software components, if needed
- Software tools for future development
- Processes that meet the business needs and goals.

In this stage of the software development life cycle, it's expected to create either simple, easy to understand wireframes that show the functions and interactions of the product – or prototyping, if you need feedback from users on the proposed features.

Privacy - Terms

5. Measuring and Tracking Progress

Before moving on, we have to take a look at this step of the planning process checklist.

You need to set **software metrics to track and measure performance and code quality.** Setting these metrics should take place at the beginning of the project development, to secure a clear and steady workflow with as few roadblocks as possible

Without these measurements to collect and provide valuable data, you cannot manage the project effectively.

Project analytics, or development analytics, are essential tools that help you accurately spot and then clear out potential or existing obstacles during the software development life cycle.

As you track and measure these software metrics, it becomes easier to see work trends, as well as the level of code quality and deployment readiness, to name a few. Learn what metrics to track here: Software Metrics Leaders Can Measure to Track the Development Process.

Privacy - Terms

Our tool provides you with all the relevant developer metrics you need to track and measure work from start to finish – in real-time.

You have, for example, the Project Timeline report that shows you the most actionable metrics in code-based activity. It helps you see what your team is working on over a sprint or a custom time frame.

Privacy - Terms
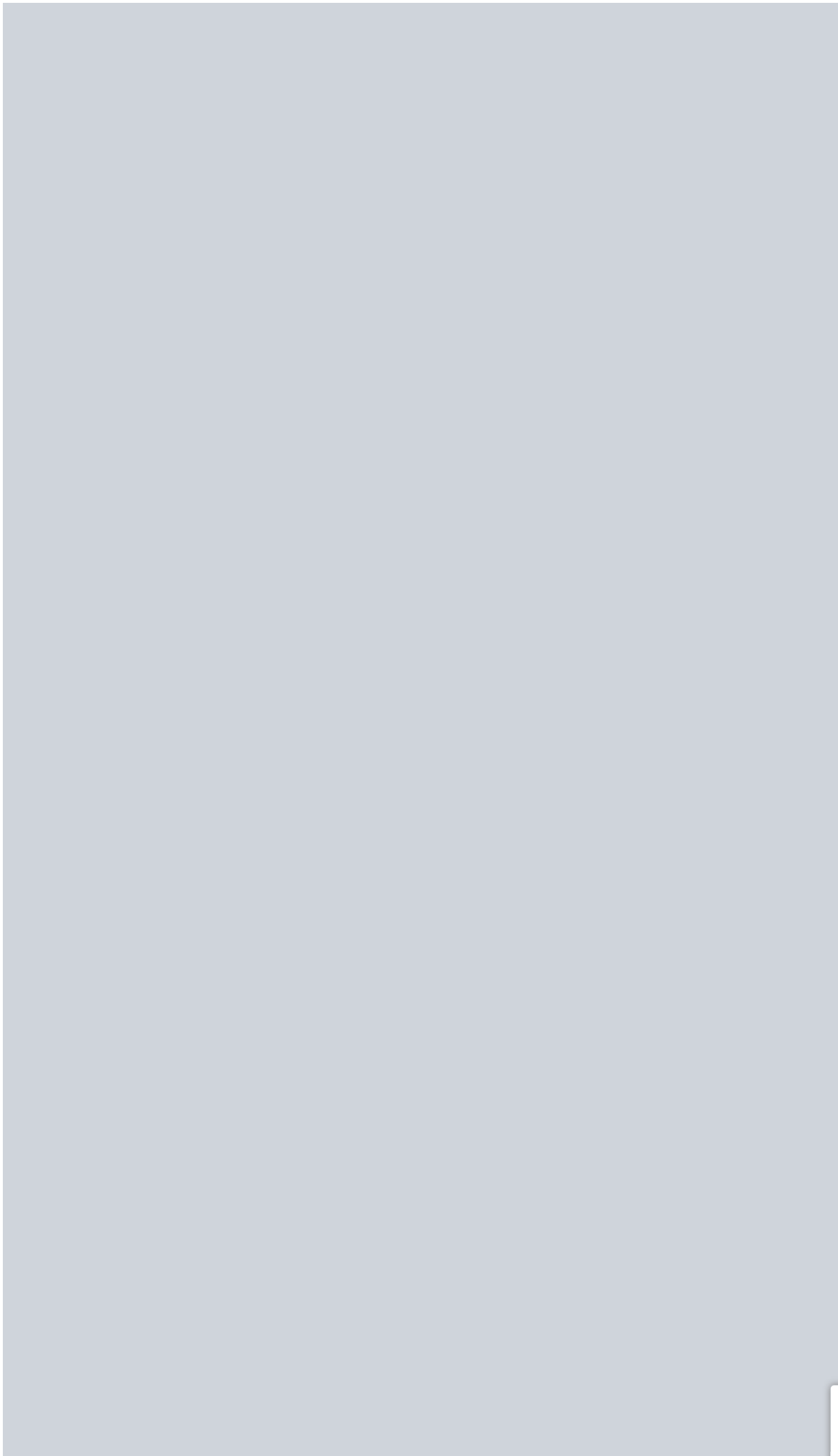
Depending on the project requirements, evaluating them will lead you to build better software faster, to ask for additional resources, or adjust the delivery dates based on objective data.

You get a crystal-clear overview of the Code Review Workflow metrics to learn what's going on in the review process to **optimize engineer collaboration.**

And you also get incredible insights with automated executive reports on Resource Planning and Project Costs to understand how resources are spent and to help teams ship on schedule.

Working remotely? Request a demo and get the Agile data-driven approach to software development no matter where your teams are.
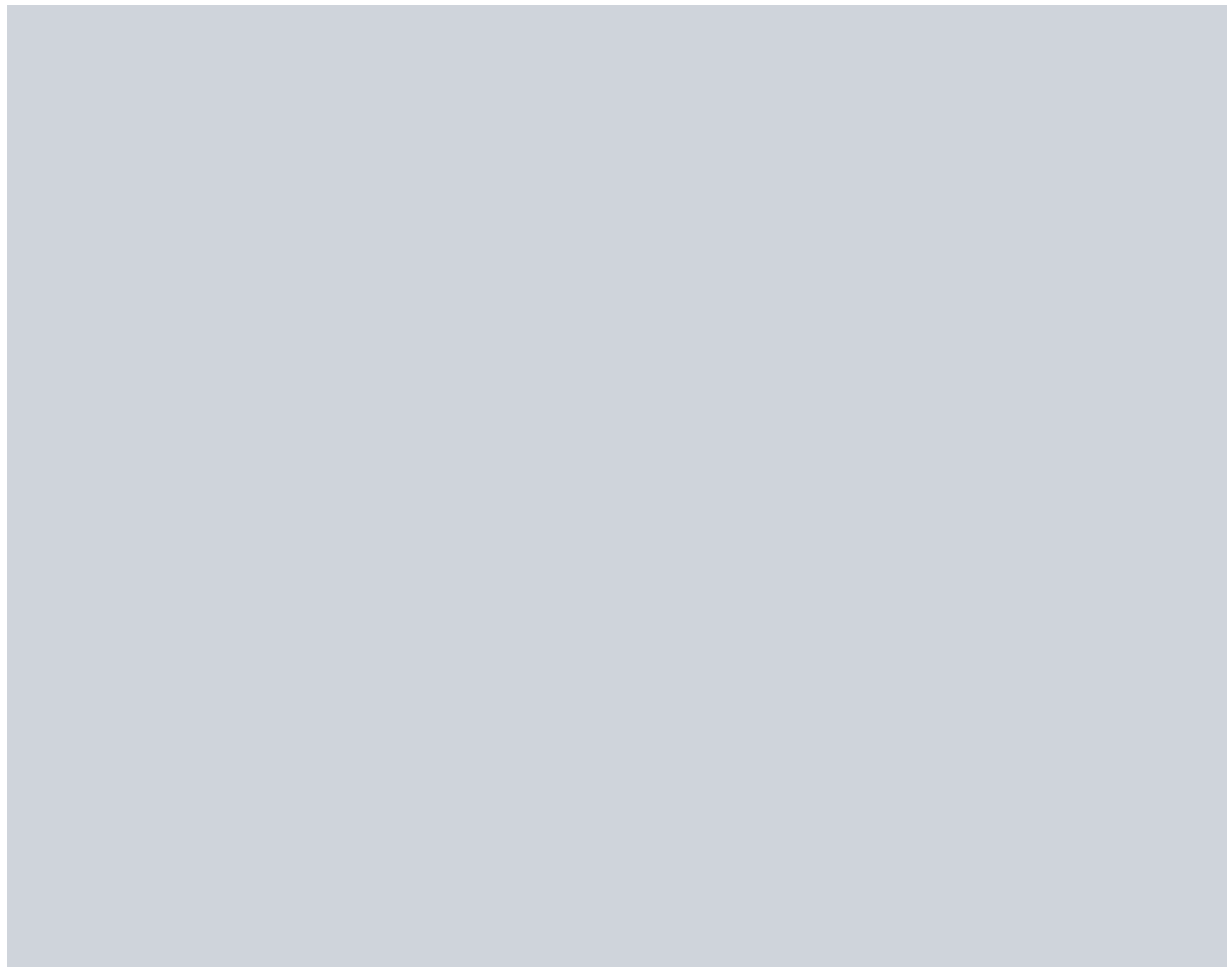
6. Development & Coding

Privacy - Terms

Once the design step is complete, it's time to move to implementing them through coding. So this next one in planning effective software development projects is one of the hardest, most complex and time-consuming one: **the software development stage.**

The software development team is now starting to build the solution according to the scope of the project, requirements and proposed functionalities previously agreed on. Depending on the size of the project, it can be a simple and quick process, or a more complex one that requires various integrations and functionalities.

Either way, the role of the planning work is to make sure you keep potential risks at a minimum and build a stable, clean and effective software product.
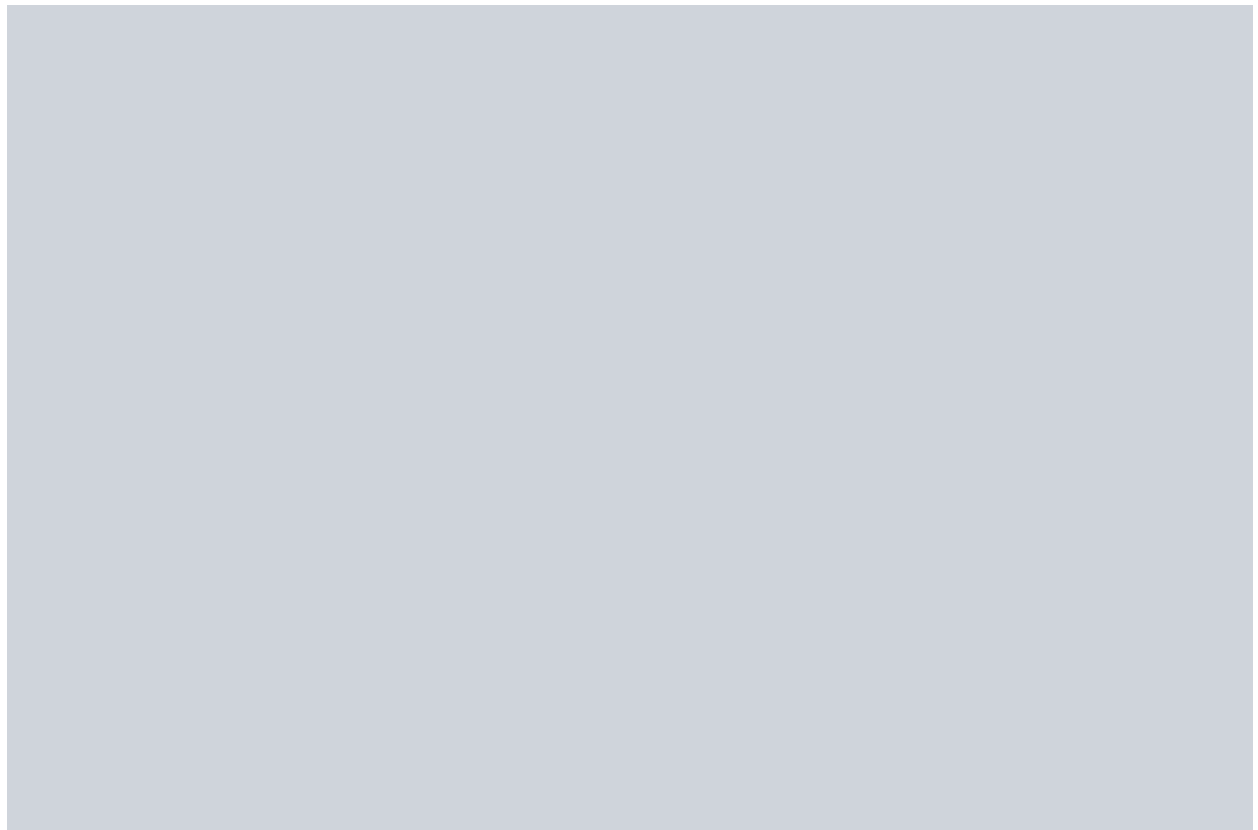
7. Testing & Quality Control

Still, the delivery of reliable software is not done without quality testing. You'll have to check:

- Compliance with project requirements

- Meeting the budget and time estimates

- Code quality and other relevant software metrics.

That is why this step of the planning process requires you to have these aspects set in place before starting the actual work.

Oftentimes, development and testing are conducted simultaneously to make sure each software feature your team is developing is bug-free and ready to go. However, once the software solution is completely developed, it still needs a full round of testing to make sure the final product is working as it should and it's bug-free and useful to end-users.

This is why we have this step separate – it's also a stage that takes more time to finish.

The testing and quality assurance phase can also be covered by an external team. If that's the case, they can provide a separate plan for this stage.

Privacy - Terms
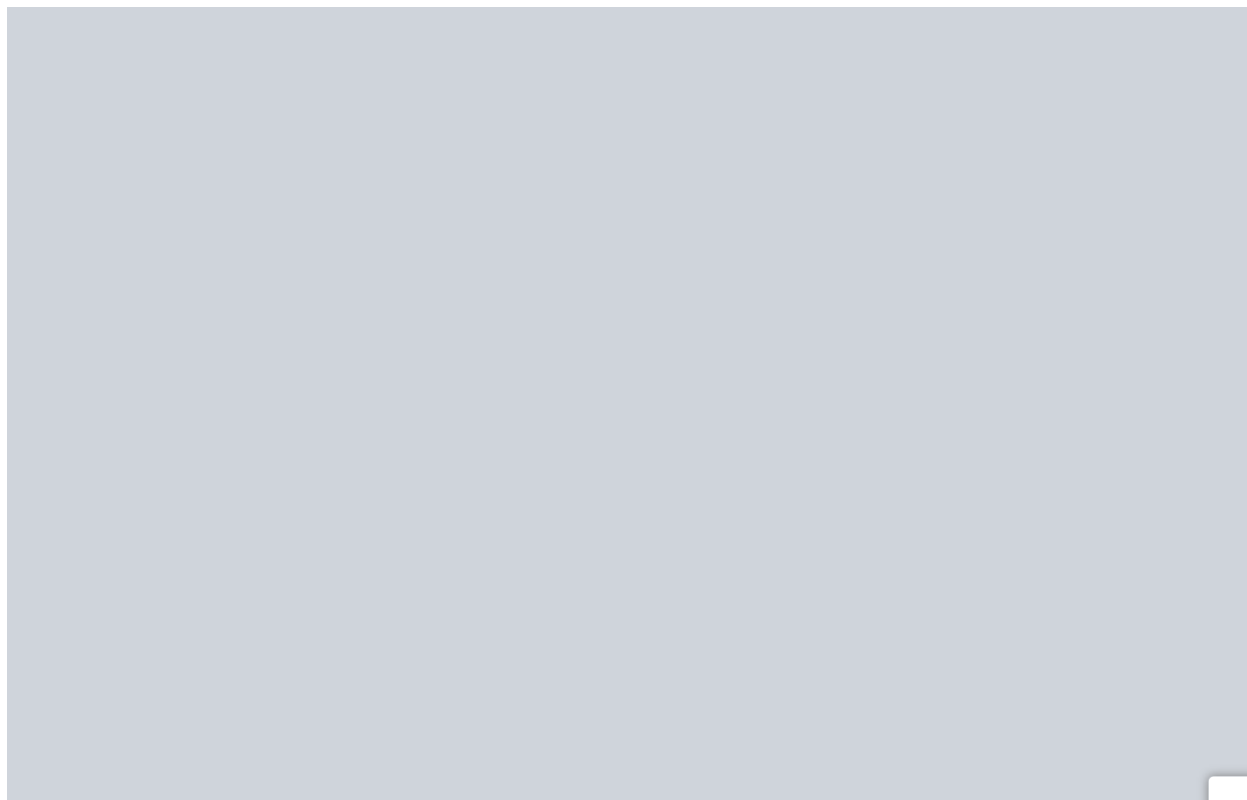
8. Deployment to Production

This is the (almost) final stage of every software project development process: deploying the code into production. It's the moment when all the hard work your developers put in is finally launched live to end-users.

Deploying is not necessarily taking place at the same time as the production stage. Software projects are, again, pretty complex. So you might perform several deployments during the development and testing stages, before having the final deployment to production.

Sometimes, the process looks like this **during the development, testing and production stages:**

- Your team builds, deploys and tests code until it is ready for release;
- That code becomes a release candidate;
- Your team deploys the release candidate for pre-release testing;
- Work on release candidates until one is stable enough as the final one;
- Name that stable release candidate as the release version
- Deploy the release version to production.

9. Support & Maintenance

Even though the planned development work is done, that doesn't mean the work on the software is also done. After launching it live, the software usually goes through other stages of its development.

The post-production stage refers to the support and maintenance of the live software. Almost all digital projects need some sort of basic maintenance or maybe more advanced updates – at some point or on a regular basis, hence the name of software development **life cycle.**

Just like customer needs are always evolving, so do the products and services they use.

Privacy - Terms

The goal of this stage is **to ensure the scalability and improvement of the initial software to better serve its users** through:

- Basic maintenance of the existing software

- Bug fixing – there will always be bugs to fix, once the people use the live software;

- New (or improved) features and functionalities to be added, depending on the user feedback

One important thing to remember is that the client should have everything about the project available to them. In your planning process, make sure that:

- The client can have technical assistance post-deployment (usually agreed initially);

- You have complete documentation of the software project from ideation to production;

- There is a documented plan on maintenance, support and documentation practices;

- There's a clear procedure when it comes to technical debt and code quality control (keeping developers accountable). Find out more here about measuring developer productivity with the right performance metrics.

Make sure you have all this data written in the project timeline and list of activities (tasks and subtasks) so that you'll know how to prioritize them accordingly.

Privacy - Terms

Wrapping up: How Can You Improve the Success of a Software Development Project?

Use the full potential of effective project planning to avoid (or minimize) these risks:

- Incomplete or inadequate specs

- Incomplete documentation

- Communication gaps between the client, the team, and stakeholders

- Scope creep (because of lack of an Agile data-driven methodology)

- Delivery delays

Regular progress checks are necessary at every stage. Answer these questions:

- Did we achieve the milestones and targets?

- Are we within the agreed timeframe and budget?

- Are the resources being used optimally?

Privacy - Terms

But all the steps in our guide will ensure you're planning successful software projects, no matter the size or complexity of the requested solution.

Fortunately, today you have development analytics at your fingertips – not just to plan and manage more effectively, but to boost team performance and morale, the client's satisfaction with the results and, of course, the alignment with your own organization's goals as well.

That being said, **we hope you now have a better understanding of how to plan effective projects with the help of this guide and a data-driven approach.**

Measure organization-level efficiencies, and start optimizing your development process with Waydev. Try it for FREE or Schedule a DEMO.
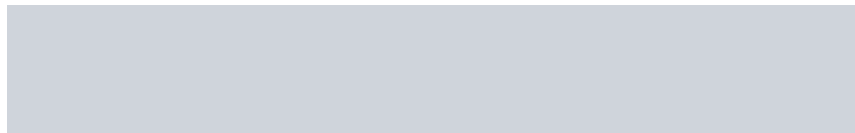
AUTHOR

Alex
Co-founder at Waydev
Get in touch
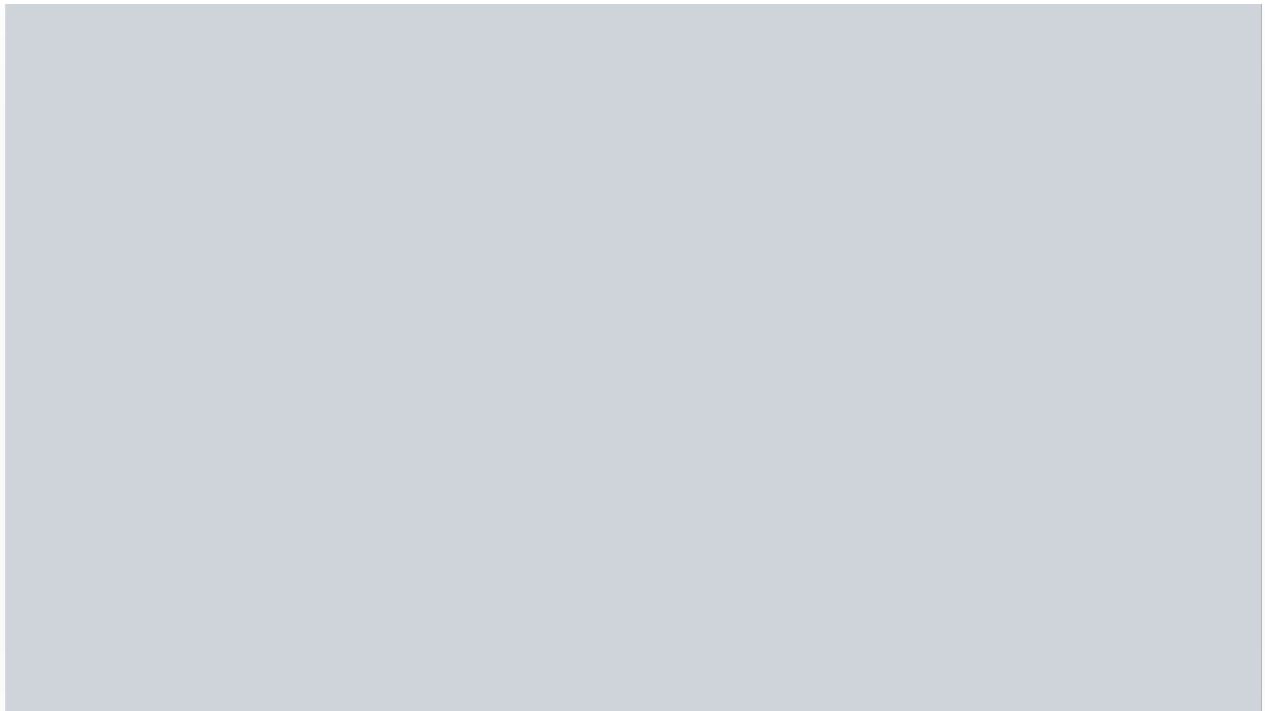
TOPICS

Data-Driven Decisions   Developer Progress   Development Analytics   Development Project
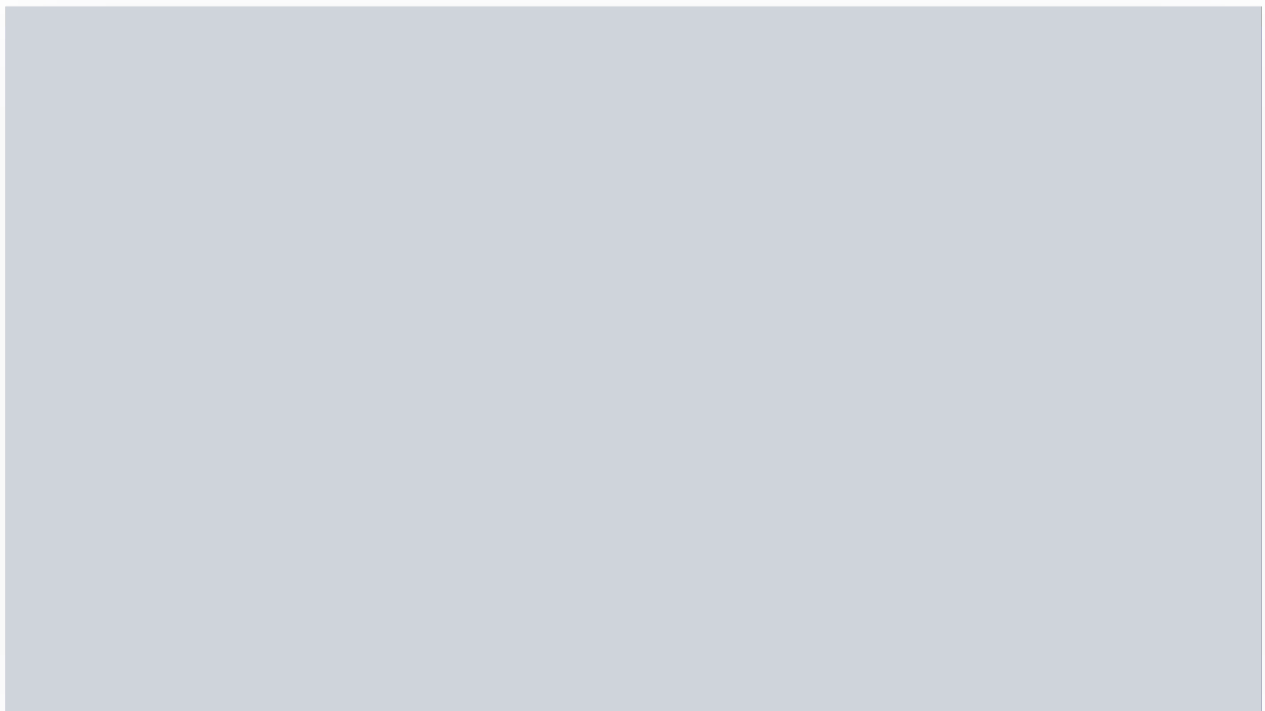
Share Article

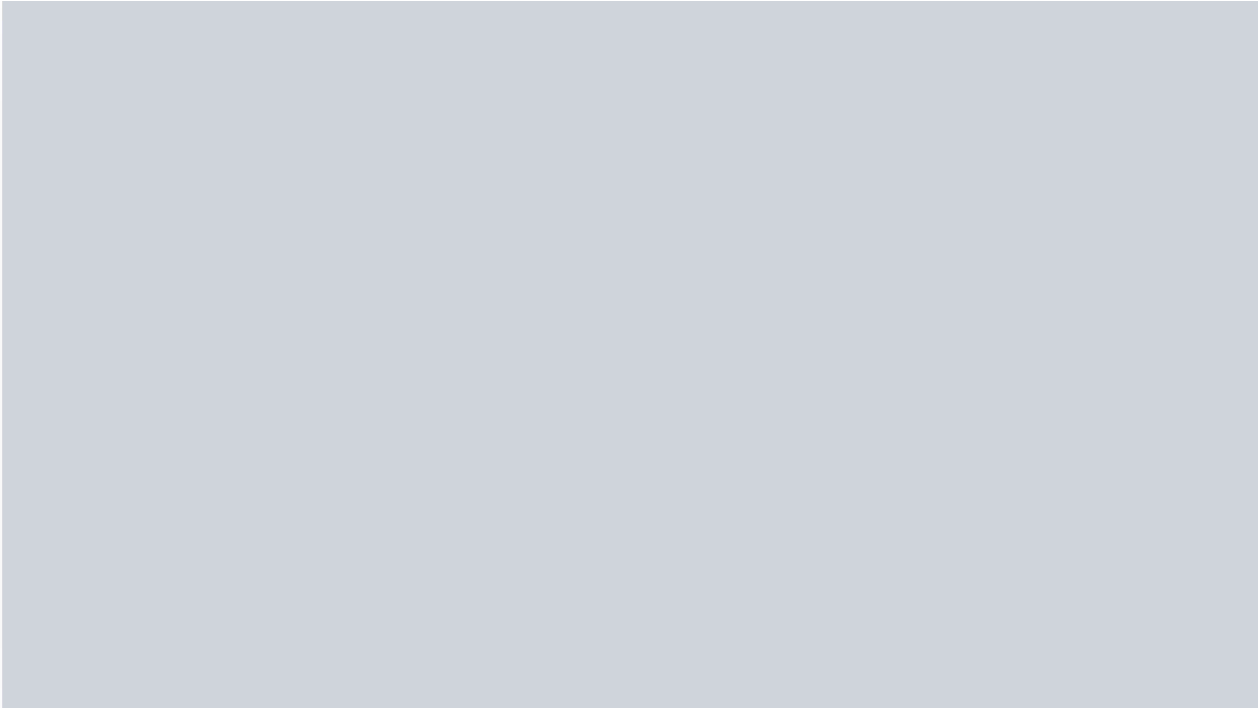**More from our blog**

Privacy - Terms

2022 Data-Driven Engineering Leadership E-book

APRIL 21ST, 2022

Waydev named Spring 2022 Market Leader in Software Development Analytics, new documentation, better UI, and ClickUp integration

APRIL 13TH, 2022

Privacy - Terms

MARCH 29TH, 2022

READY TO IMPROVE YOUR TEAMS' PERFORMANCE?

Request a product demo

Schedule a call

Request a POC

Git Analytics Intelligence platform for engineering leaders

Market Leader in Development Analytics **(G2 Crowd's Winter Report 2022)**

contact@waydev.co
(+1) (650) 449-4492

2021 Fillmore St #34
San Francisco, CA 94115

1007 North Orange Street #97
Delaware, DE 19801

Privacy - Terms

OUR CERTIFICATES

OUR PARTNERS

INDUSTRIES   ⌄

FEATURES   ⌄

USE CASES   ⌄

COMPANY   ⌄

RESOURCES   ⌄

Terms of Service
Privacy Policy
Security
SOC 3 Compliant
Status

© 2022 Waydev Inc. All rights reserved.

Privacy - Terms