Understanding and Optimizing SDLC processes: How to Improve and Scale the Phases of Your Software Development

Software Development Life Cycles are not new to development teams, as tech leaders across all organizations implement them. Although extremely popular and well understood by professionals, SDLCs pose challenges to engineering managers, who want to improve the efficiency and effectiveness of their teams and increase software time to market.

When well-defined, SDLC processes, enable software engineering managers to have more observability into their teams' workflows, get clarity on requirements and make accurate predictions regarding time, costs, and deliverables.

Engineering management platforms like Waydev give tech leaders and executives full transparency over the entire SDLC process. Valuable, real-time insights enable them to make good decisions based on more than just gut feelings. With the right tools, resource allocation and task prioritization are clear, and projects benefit from successfully implementing the suitable systems that ensure software quality.

To help you choose between different SDLC methodologies, identify the ones that enable you to drive maximum results for your team and customers, and streamline your processes, we will walk you through Software Development Life Cycle best practices. In the following sections, you will get a quick overview of the basics of SDLCs, the differentiators, advantages, and disadvantages of each methodology, as well as more details on how to improve your Software Development Life Cycles and deliver high-quality projects.

SDLC fundamentals: scope, phases, and methodologies

The Scope: Software Development Life Cycle, SDLC, is a process meant to help engineering teams produce the highest quality software with the minimum cost of financial and human resources. Its main objective is to optimize design, development, and software delivery efforts. The process is iterative and multi-steps, and it provides management with a systematic approach to software delivery. Why you would choose to deploy a particular methodology

or the other depends on many factors, including business requirements, budgets, time, team expertise, and many more.

The Phases: SDLC processes include 5-7 phases, depending on the needs and the methodology applied. These stages don't always follow one another. Also, cycles may repeat depending on the project's complexity.

By using Waydev across the project's stages, you may optimize work and streamline operations, achieve a higher level of engineering productivity, increase outputs and revenue.

Ideation: How to make this SDLC phase more efficient

The ideation phase starts with introducing team members to the business scope and project goals. For things to run smoothly and for the stage to be efficient, it is essential to study and review documentation provided. You need to get familiar with software already out there and get accustomed to the overall landscape.

At this time, software development and stakeholders meet on several occasions to address the project's requirements and brainstorm for ideas.

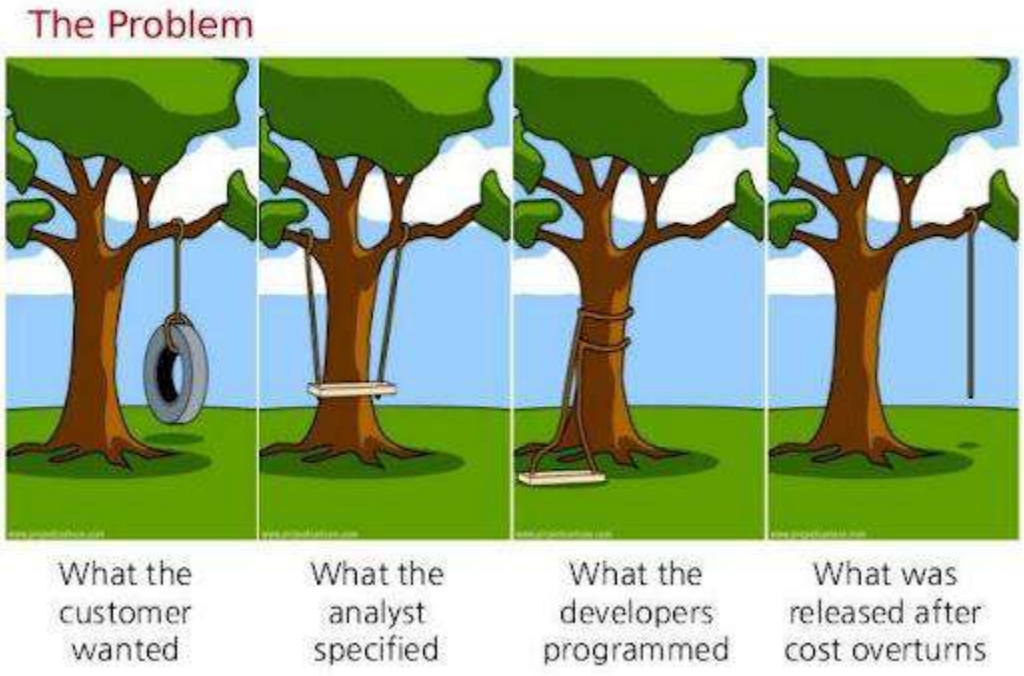
While everyone may have a different view on the final result and its functionality, it is always helpful to understand the end user's needs and address specific issues around their niche.

By the time this SDLC methodology stage is closed, the team should understand the project's general lines, purpose and benefits, stakeholders and overall patterns, competitors and broad landscape, key industry needs, and limitations.

How to make sure that Requirements are followed

When it comes to requirements, the project team discusses what the user wants from the software product, making a list of functionalities and deliverables. This enables parties to understand how the solution looks and works and ensure that the team has the time, the financial, and human resources to put it into practice. Once requirements are

set, team members should all know what is expected and what tasks and volume of work they have to deliver.



Market Leader in Development Analytics, Waydev, gives you a clear view of how requirements are being respected. Our budgeting reports help our customers understand the costs behind unplanned work and bugs and visualize the progress for each of their key initiatives.

With insights on the progress and costs of critical initiatives and deliverables, engineering executives can ensure that teams ship on schedule and according to requirements.

System design issues that you need to watch out for

During the design phase, developers and technical architects focus on designing the software and overall system, ensuring each requirement has a technical correspondent.

Since the results of this phase significantly impact the project, it is essential to ensure that all decisions validated set a good foundation for the product in the short, medium, and long term.

Some key aspects that engineering managers need to pay attention to are the technology and frameworks used and the configuration and implementation principles. It is equally essential that engineering managers provide their team with the complete requirements and the necessary tools to get the project going.

Every requirement is important at this step, and engineering managers need to look at them separately to ensure no blockages and that the process is running smoothly. For example, having more observability into your team's CI/CD processes enables you to eliminate bottlenecks and reduce costs.

When developing additional features for a product that already exists, it is mandatory to ensure that the new architecture matches what is available, integrating constraints.

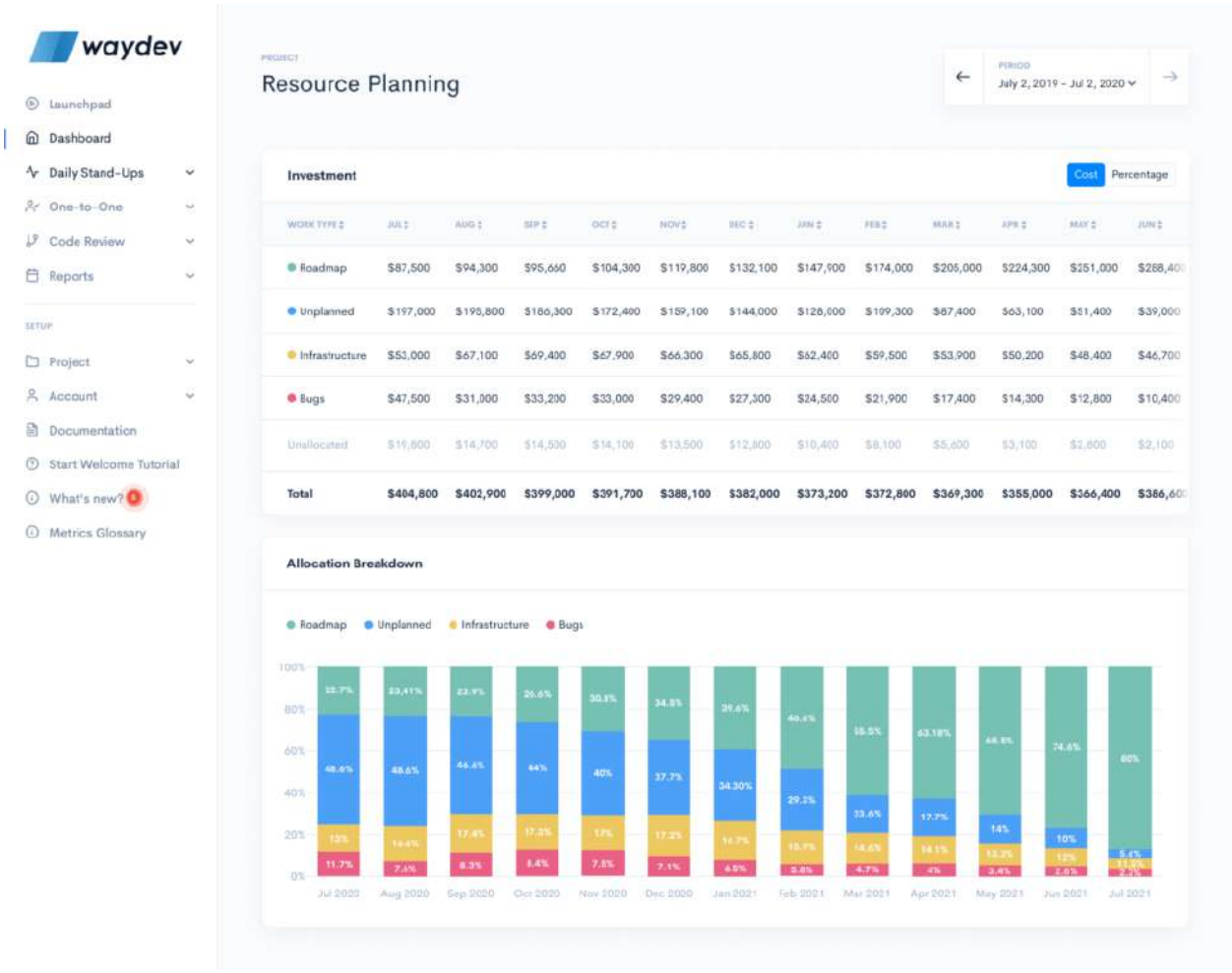
On the complementary spectrum, for new projects, scalability is critical, so the team needs to ensure that the project may be changed, grown, and expanded in different ways over time.

Of course, other things you should pay attention to include security and risks. When approving the design, the project manager should inform the stakeholders of the medium and long-term risks and needs to ensure they are aware of and agree with potential issues or further investments.

How to ensure a smooth SDLC Development Phase

In the development phase, detailed requirements and system designs are translated into actual system elements. These are tested for usability and prepared for integration into the overall system.

Waydev enables you to monitor the efficiency of your development process by offering a real-time view of what your team is doing. Our solution increases average active coding days by 15% per week while decreasing code churn rates by a reported 28%.



Waydev's Resource Planning report pulls in data from your Issue Tracking systems, as well as Salary data, to show how much your organization spends on Bugs, Unplanned work, and more.

Quality assurance in SDLC – why is it important and how to achieve it

Software quality assurance is a must in different types of SDLC methodologies. It helps teams ensure that the delivered products

and functionalities meet expectations, that they are bug-free, and that they work correctly, on different terminals, and in various conditions.

The best approach is to include testers in the conversation from the first stages to assess what requires testing and what to prioritize.

For QA to deliver the best results, we recommend encouraging an organizational culture focused on quality. Teams and individuals understand the importance of continuous testing and the impact on a project and, even more, on the business.

How to reduce cycle-time and accelerate the Deployment Phase

Once the user has tested and validated the system, it may go to deployment. Software may now be installed and provide operational and business value for the end users. Implementation includes everything, from informing users of new products or features, training them, installing hardware (where applicable), installing software, and integrating the new products into daily flows. Only when the business

confirms that the requirements are met and that the system is operating at high standards is the phase considered closed.

Successful engineering managers associate optimizing cycle time with accelerating the deployment phase and with scaling their SDLC processes. By continuously improving the teams' workflows, they don't compromise on the project's requirements in terms of time to market, deadline, or quality.

At Waydev, we understand that when managers focus on cycle times, they foster result-oriented environments while also ensuring team members feel empowered, more satisfied, and have higher performance ratings.

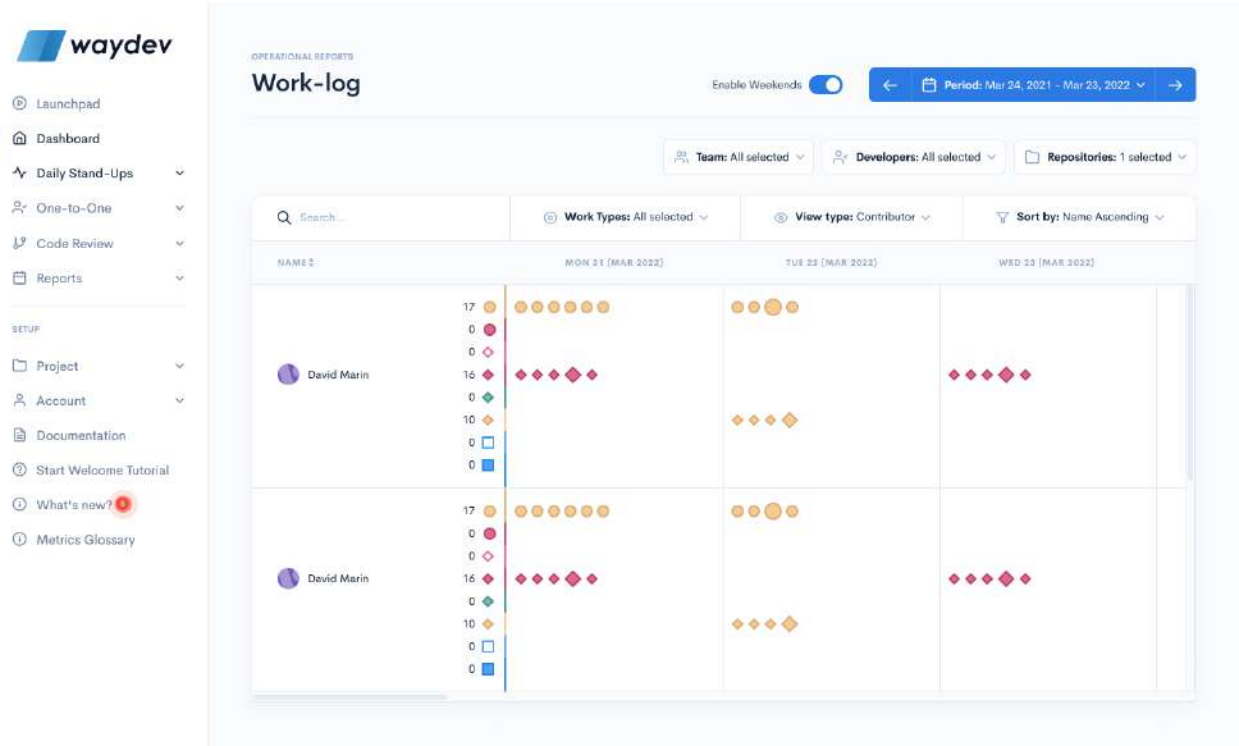
Our solution automates cycle time monitoring to cover all aspects and tasks performed by team members and provides real-time views and detailed periodical reports, long-term visualization, and comparisons. Waydev, thus, identifies impediments and bottlenecks, turning data into insights, which enhance performance and output.

If you made it this far, the Maintenance Phase would feel like a breeze

Work doesn't stop when the product is fully delivered. Any form of software is an ongoing project that needs to continuously respond to both organization's and end user's needs, ensuring performance.

Maintenance usually includes software upgrades, repairs, and fixes, if or when the software fails. These require additional testing.

In this stage, technical debt is visible, showcasing to what extent it affects the project. If engineers choose shortcuts to respond to user requests, delivery deadlines, or financial constraints, this is the time to assess their impact and repay technical debt.



With Waydev, you can visualize both new work and old work activity for your team and individual levels. The Work Log dashboards enable color-coded tickets to ensure it offers you a clear understanding of how technical debt affects your work.

Choosing The right type of software development life cycle methodologies for your project

Choosing between different SDLC methodologies for your projects is not an easy task, primarily since it depends on several variables such as the stakeholders' needs, the business priorities of the project, the capabilities of the technical team, financial and technical constraints.

A decision implies you have a good understanding of all the attributes and differentiators of all software development life cycle methodologies. In the following section, we will focus on developing each, paying attention to the advantages and disadvantages that they bring to the table.

The good old Waterfall methodology

This is where it all started: back in 1956 when Herbert D. Benington described the use of this methodology for the first time in software engineering.

Waterfall is the first SDLC model to be applied by engineering teams. As its name suggests, it is a linear-sequential life cycle model. This

means that each phase must be completed before the next one starts, as no overlapping is accepted.

Its 6 phases that cascade one another include: requirement gathering and analysis, system design, implementation, integration and testing, deployment of the system, and maintenance.

The Waterfall approach delivers the best results when the project requirements are well documented, and there aren't any grey areas. In this context, teams need to know what they should achieve and how the result should look and work. Waterfall requires a high level of stability, which means that projects should be based on a good understanding of the technical environment and ample resources that support team members in delivering each stage of the process. Linear-sequential life cycle models like Waterfall usually work best for short-time projects.

Advantages for choosing Waterfall

In terms of advantages, Waterfall has a clear structure set early in projects and is effective for small projects because teams need to complete a phase before the next one starts.

When it comes to setting the premise for success, the methodology is reliable, as it provides structure, ensuring that team members benefit from clear deliverables and tasks. Moreover, they may also efficiently transfer information from one side to the other.

Disadvantages of using Waterfall, the drawbacks

Waterfall is not a flexible way of approaching projects as it doesn't accommodate changes well. Moreover, the fact that it delays testing until the second half of the project makes it difficult to identify bottlenecks and even more complicated to manage them. These are some of the reasons why Waterfall is not ideal for long, complex projects.

In the relationship with the stakeholders, Waterfall has a low end-user involvement since the stakeholders only have access to it

in the later stages. This might translate into change requests that are difficult to accommodate and turn out to be costly.

Prototyping SDLC methodology

As its name suggests, prototyping involves developing a prototype of the project so that the team may test it and refine it together with the stakeholders until it reaches its best form. The model is excellent for when the business can't pinpoint the exact requirements, so a bit of exploring is necessary.

Advantages for choosing Prototyping

Prototyping offers a high level of design flexibility and is very good for accommodating changes. Due to its constant interaction with users, this model detects eros and missing functionalities at the right time. This form of transparency makes businesses feel comfortable across the process and trust what is happening.

Disadvantages of using Prototyping, the drawbacks

Since prototyping requires developing actual prototypes, it may prove to be an extremely costly model until the stakeholders accept the result.

Out of all SDLC methodologies, Prototyping has the highest number of valuables. Its requirements are constantly changing and diversifying; thus, it may be poorly documented.

Moreover, as iterations are known from the start, it is difficult to objectively assess team efforts, hours, and financial needs.

Another drawback of the prototyping model is that it has a higher project abandonment rate than the others. This happens because requirements aren't clear from the start. After seeing a prototype, stakeholders often decide not to go through with it.

Spiral model

The Spiral model combines the Waterfall and Prototyping methodologies and is mainly used for large, complex projects because it mixes structure with flexibility. This SDLC has four stages: requirements identification, design, construct or build, evaluation, and risk assessment.

The model is prevalent in the software industry, and it is excellent for projects that present medium to high risks, as businesses are unsure of the requirements and desired results.

Advantages for choosing the Spiral model

The spiral model makes optimal prototypes, enabling users and customers to see and offer feedback on the deliverables at an early stage. This makes it highly flexible to change, which is an excellent plus.

When utilizing it, teams also reduce risks, as they may develop the complicated parts first, thus ensuring they have the time to solve everything without complicating things in the development process.

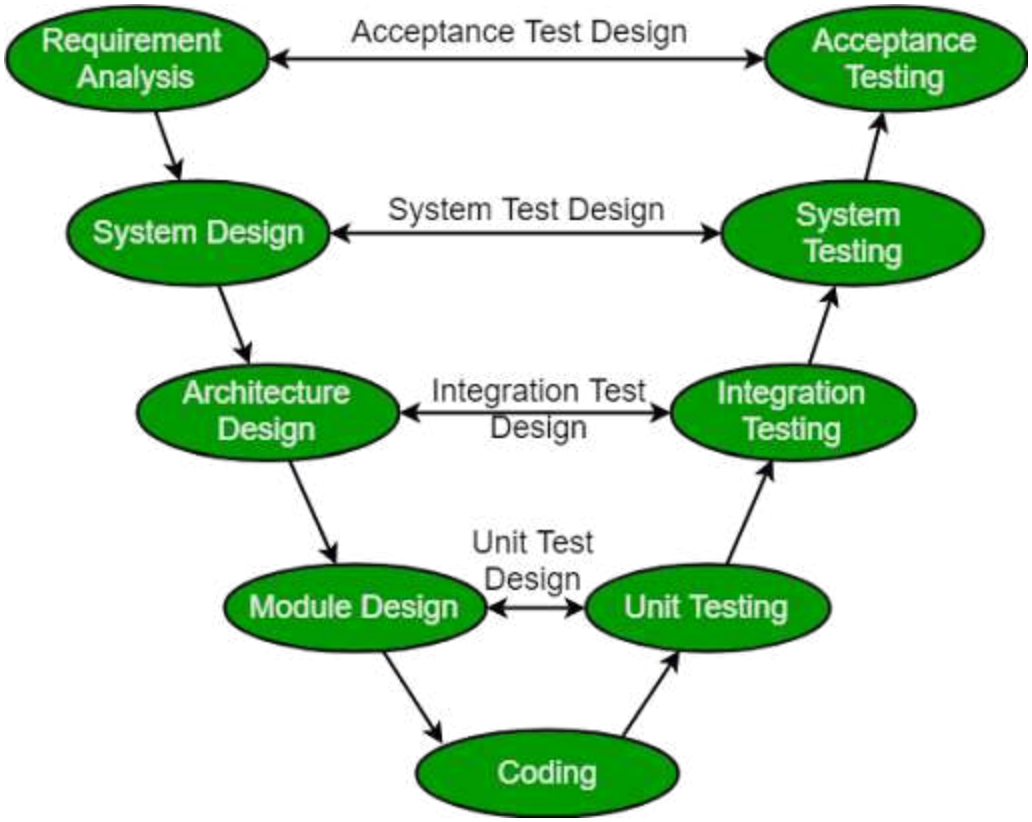
Disadvantages of using the Spiral model, the drawbacks

The Spiral model is complex and, thus, challenging to manage because spirals may follow one another indefinitely. Of course, when adopting the methodology, the project may be split into small stages, which would make it easier to follow, but that would require a high level of documentation.

Overall, this is not an approach suitable for small-sized products that don't present risks because it is pretty expensive, and the costs will certainly not be justified.

V-model

Also known as the Verification and Validation model, the V-model is an SDLC model that happens in a sequential V-shape manner. Each development stage has an associated testing stage. Taking elements from the Waterfall model, the V-model has a high level of structure since phases may start only after the completion of the previous one.



The model best suits short projects with clearly defined requirements and fixed technologies. (Source)

Advantages for choosing the V-shaped model

With its sequential phases, the V-shaped model provides an excellent level of structure and discipline, making it easy to deploy and manageable, especially in the case of smaller projects, with precise requirements.

Disadvantages of using the V-shaped model, the drawbacks

This type of SDLC methodology lacks flexibility and doesn't accommodate changes. When applied, the software is produced in the late stages of the project, making it difficult to get stakeholder and end-user feedback.

This is why it is unsuitable for longer, complex projects or ongoing requirements.

The Agile SDLC model

This is a highly flexible model because it combines iterative and incremental elements to offer the highest level of adaptability and stakeholder involvement. When starting a development process, team members only know what features they need to develop without having a clear list of tasks and plans. Failure risk is minimized by splitting the project into small incremental builds that are released continuously and assessed by the business and end-users.

This dynamic approach to software development brings together cross-functional teams, where different professionals work simultaneously, focusing on planning, requirement analysis, design, coding, unit testing, and acceptance testing.

Advantages for choosing the Agile SDLC methodology

As stakeholder interaction is the backbone of this SDLC methodology, Agile benefits from high customer orientation and team members' collaboration. It is also highly realistic and pragmatic. That is why it is flexible in accommodating the change.

Out of all SDLC models, Agile is most suitable for ongoing projects that need to adjust to market requirements and demands.

Disadvantages of using the Agile SDLC methodology, the drawbacks

Agile is the ultimate proof that, with flexibility, comes responsibility. The SDLC methodology depends on the professional abilities and

expertise of the engineering manager and the team members' capabilities of collaborating because the model relies on very little documentation.

This makes it less suitable for complex projects. Moreover, since, in Agile, user feedback and interaction are essential, any lack of response or delay will constantly affect the progress and the overall deadline.

Waydev enables developing teams to switch to a data-driven Agile methodology and allows executives to identify potential bottlenecks and roadblocks. Managers generate reports on activity at a group and individual level in real-time and remove distractions.

DevOps SDLC methodology

DevOps is considered a disruptor in the SDLC ecosystem since its philosophy is to bring cultural change that focuses on delivering functionalities faster while also increasing the quality rate. With DevOps, teams bridge the gap between developers and operations, fostering near real-time deployments several times per day, with

minimum downtime. The model involves a high level of automation that reduces human intervention and emphasizes continuous feedback and improvement.

Advantages for choosing the DevOps SDLC methodology

With its stakeholders in mind, this SDLC methodology brings teams together for a common goal: delivering better software faster.

What DevOps changed in the overall software development life cycle is that it simplified development by enabling teams to focus on one feature at a time. This way, if things go wrong, the issues may be easily identified.

Moreover, the model's highly automated DNA improved development, making it more efficient.

Disadvantages of using the DevOps SDLC methodology, the drawbacks

What does your team prefer, the road they know or the shorter road? If the answer is the first, you might be in some difficulty. Given that this is a disrupting model, the adoption of DevOps requires significant changes in organizational flows and culture. This might make the learning curve steep and complicated.

Another aspect you should pay attention to is that the approach relies very much on speed and requires rapid feedback, thus reducing the opportunity for careful consideration.

Rapid development often poses security challenges, so teams need also to have security procedures implemented.

The data-driven approach to measuring your team's SDLC processes performance in DevOps models

In a data-driven environment, each team member uses precise data as a basis for decision-making. To democratize data, professionals need to have constant access to it. By providing insights on the team and individual metrics and a real-time view of what professionals are doing, Waydev enables engineering managers to leverage metrics and minimize risks in assessing DevOps models.

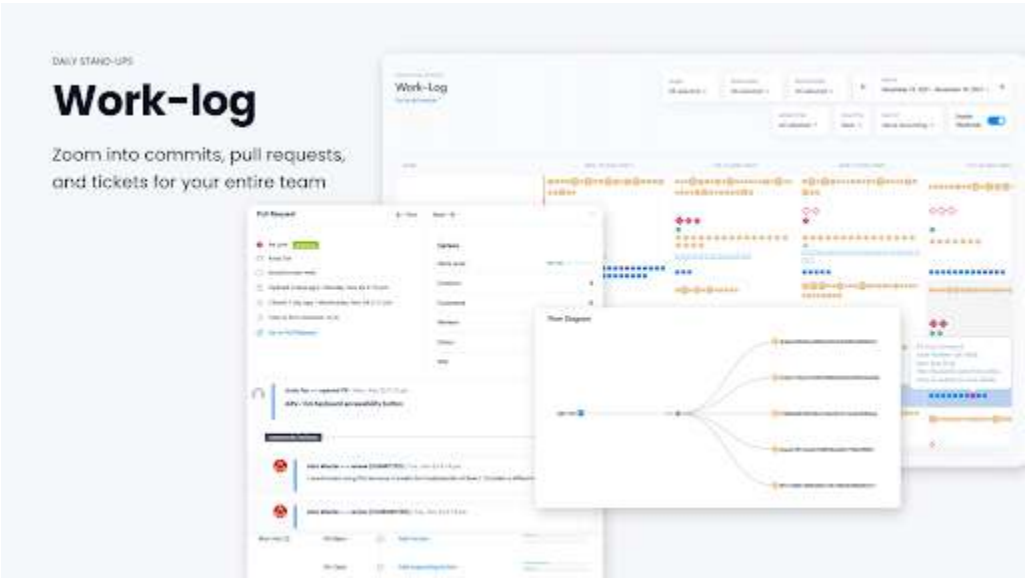
When it comes to measuring your team's SDLC process performance, you can track metrics at each step.

Continuous Development (CD)

With continuous development, software development constantly happens in an iterative form.

CD requires monitoring real-time performance metrics visible to team members and focusing on team and individual contributions to iterations. By keeping an eye on these, managers will identify bottlenecks and impediments, tracking what affects burn-down rates (calculated as tasks remaining versus time remaining) and velocity.

No matter what SDLC you use, either a more traditional Waterfall method or a modern, more flexible Agile one, you still need a data-driven approach to understand the team's real-time progress and performance. You can focus on metrics that enable you to offer constant, continuous feedback and generate improvement opportunities.



Waydev's, Work-log feature gives you a clear view of the output of each contributor in real-time. Our features track focus metrics like new work, churn, legacy refactor, helping others, impact.

Continuous Integration (CI)

With continuous integration, developers constantly merge their changed code to the branch, thus reducing the chances for last-minute challenges that affect releases.

Successful projects often require continuous integration servers that automate builds by managing the shared repos and the code coming in. Choosing the correct CI server is a critical task that depends on several variables, such as your project requirements, the tech stack that your team uses, and the way you approach your daily operational flows.

Waydev integrates with multiple CI/CD tools, like Jenkins, Azure Pipelines, CircleCi, and more.

By pairing Waydev with any CI/CD tools, users benefit from code-level analytics, real-time code review metrics, and project management insights, which come with a wide array of functionalities and flexibility.

Continuous Testing (CT)

Continuous testing starts from the premise that executing automated tests across the software delivery pipeline is beneficial to a project.

CT offers a series of benefits. The most popular is that it keeps up with the developers' work, provides immediate, real-time feedback, and identifies risks early in the SDLC, thus optimizing efforts and costs.

Continuous Deployment (CD)

When applying the Continous Deployment strategy for software releases, engineers release into production each code commit that passes automated testing into production. This way, changes are implemented in record times and are visible to the software's end users.

CD helps teams automate software release processes, making delivery fast and efficient, increasing development teams'

productivity by freeing them from manual tasks, identifying and solving bugs before they escalate, deliver updates more rapidly.

Continuous Monitoring (CM)

Through continuous monitoring, IT managers may quickly detect compliance issues, security risks, and system failures of the existing operating software.

CM impacts business performance, reduces system downtown in case of a failure, enables quick, efficient incident response, and increases visibility on how the network operates, focusing on key metrics used to detect threats.

Contact us for a demo if you want to find out what Waydev can do to help you optimize your SDLC methodology of choice.

About Us

Our mission is to provide engineering leaders with a way of measuring the performance of their engineering teams. We strive to help the technology industry move towards a data-driven agile development methodology and make decisions supported by data.

We are trusted by **Fortune 500 companies**, such as Blue Cross Blue Shield, TATA, and Carrier, and we are also loved by startups (#1 Product on Product Hunt).

Waydev is the **G2 Market Leader** in Winter and Spring 2022

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