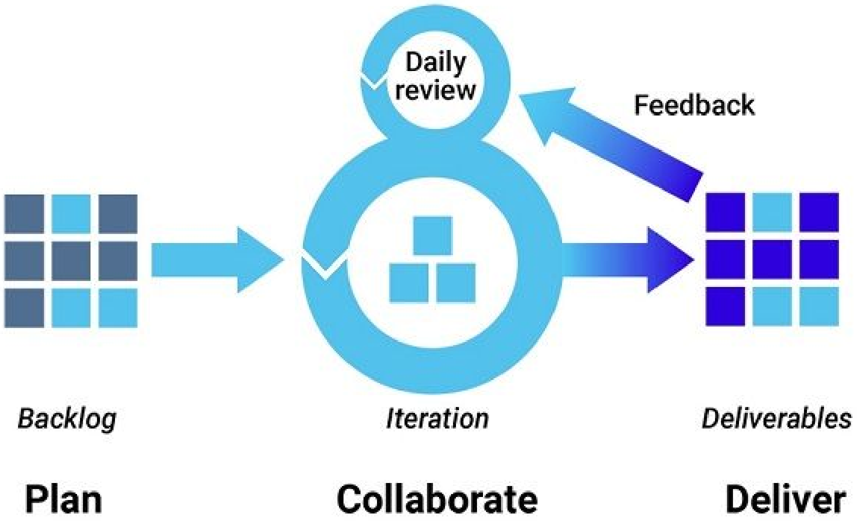
Software Development Methodologies

Agile Development Methodology

Teams use the agile development methodology to minimize risk (such as bugs, cost overruns, and changing requirements) when adding new functionality. In all agile methods, teams develop the software in iterations that contain mini-increments of the new functionality. There are many different forms of the agile development method, including scrum, crystal, extreme programming (XP), and feature-driven development (FDD).



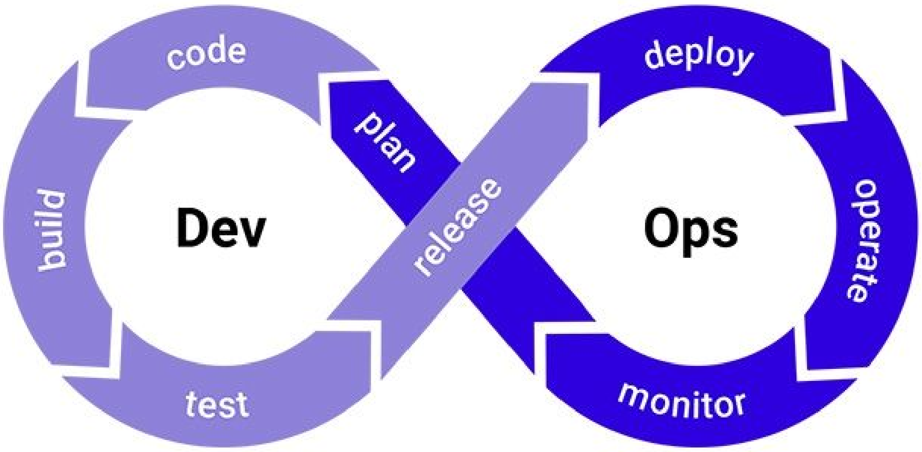
Pros: The primary benefit of agile software development is that it allows software to be released in iterations. Iterative releases improve efficiency by allowing teams to find and fix defects and align expectations early on. They also allow users to realize software benefits earlier, with frequent incremental improvements.

Cons: Agile development methods rely on real-time communication, so new users often lack the documentation they need to get up to speed. They require a huge time commitment from users and are labor intensive because developers must fully complete each feature within each iteration for user approval.

Agile development methods are similar to rapid application development (see below) and can be inefficient in large organizations. Programmers, managers, and organizations accustomed to the waterfall method (see below) may have difficulty adjusting to an agile SDLC. So a hybrid approach often works well for them.

DevOps deployment methodology

DevOps is not just a development methodology but also a set of practices that supports an organizational culture. DevOps deployment centers on organizational change that enhances collaboration between the departments responsible for different segments of the development life cycle, such as development, quality assurance, and operations.



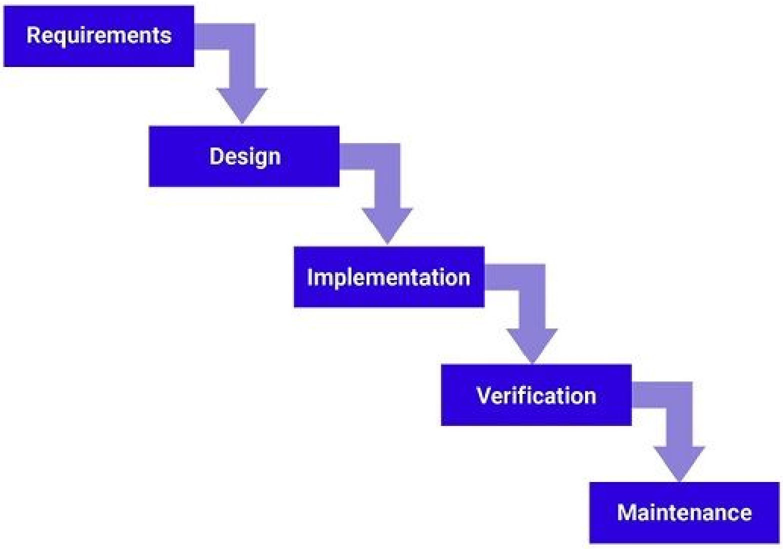
Pros: DevOps is focused on improving time to market, lowering the failure rate of new releases, shortening the lead time between fixes, and minimizing disruption while maximizing reliability. To achieve this, DevOps organizations aim to automate continuous deployment to ensure everything happens smoothly and reliably. Companies that use DevOps methods benefit by significantly reducing time to market and improving customer satisfaction, product quality, and employee productivity and efficiency.

Cons: Even in light of its benefits, there are a few drawbacks to DevOps:

Some customers don’t want continuous updates to their systems.
Some industries have regulations that require extensive testing before a project can move to the operations phase.
If different departments use different environments, undetected issues can slip into production.
Some quality attributes require human interaction, which slows down the delivery pipeline.

Waterfall development method

Many consider the waterfall method to be the most traditional software development method. The waterfall method is a rigid linear model that consists of sequential phases (requirements, design, implementation, verification, maintenance) focusing on distinct goals. Each phase must be 100% complete before the next phase can start. There’s usually no process for going back to modify the project or direction.



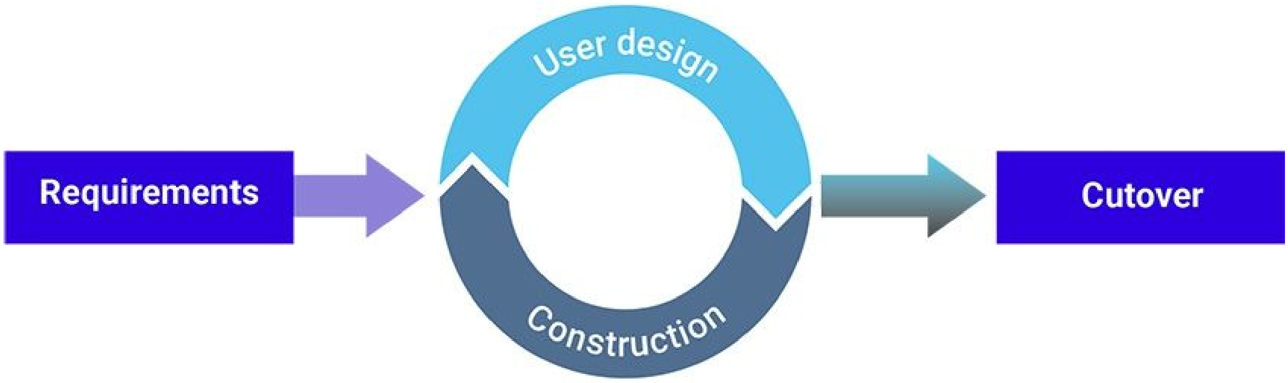
Pros: The linear nature of the waterfall development method makes it easy to understand and manage. Projects with clear objectives and stable requirements can best use the waterfall method. Less experienced project managers and project teams, as well as teams whose composition changes frequently, may benefit the most from using the waterfall development methodology.

Cons: The waterfall development method is often slow and costly due to its rigid structure and tight controls. These drawbacks can lead waterfall method users to explore other software development methodologies.

Rapid application development

Rapid application development (RAD) is a condensed development process that produces a high-quality system with low investment costs. Scott Stiner, CEO and president of UM Technologies, said in Forbes, “This RAD process allows our developers to quickly adjust to shifting requirements in a fast-paced and constantly changing market.” The ability to quickly adjust is what allows such a low investment cost.

The rapid application development method contains four phases: requirements planning, user design, construction, and cutover. The user design and construction phases repeat until the user confirms that the product meets all requirements.



Pros: Rapid application development is most effective for projects with a well-defined business objective and a clearly defined user group, but which are not computationally complex. RAD is especially useful for small to medium projects that are time sensitive.

Cons: Rapid application development requires a stable team composition with highly skilled developers and users who are deeply knowledgeable about the application area. Deep knowledge is essential in a condensed development timeline that requires approval after each construction phase. Organizations that don’t meet these requirements are unlikely to benefit from RAD.