**1. What is DevOps?**

DevOps’s main concept is collaboration, sharing goals and open communication means it’s complete the gap between development and operation. It’s a collaboration between the IT team, developer, and tester which helps to production faster.

DevOps implements the following practices:

* Continuous Development
* Continuous Integration
* Continuous Testing
* Continuous Monitoring
* Continuous Delivery
* Continuous Deployment

**Agile Development Models – Software Engineering**

**Last Updated : 03 Jul, 2024**

**In earlier days, the Iterative Waterfall Model was very popular for completing a project. But nowadays, developers face various problems while using it to develop software. The main difficulties included handling customer change requests during project development and the high cost and time required to incorporate these changes. To overcome these drawbacks of the Waterfall Model, in the mid-1990s the Agile Software Development model was proposed.**

**Table of Content**

* [**What is Agile Model?**](https://www.geeksforgeeks.org/software-engineering-agile-development-models/#what-is-agile-model)
* [**Agile SDLC Models/Methods**](https://www.geeksforgeeks.org/software-engineering-agile-development-models/#agile-sdlc-modelsmethods)
* [**Steps in the Agile Model**](https://www.geeksforgeeks.org/software-engineering-agile-development-models/#steps-in-the-agile-model)
* [**Principles of the Agile Model**](https://www.geeksforgeeks.org/software-engineering-agile-development-models/#principles-of-the-agile-model)
* [**Characteristics of the Agile Process**](https://www.geeksforgeeks.org/software-engineering-agile-development-models/#characteristics-of-the-agile-process)
* [**When To Use the Agile Model?**](https://www.geeksforgeeks.org/software-engineering-agile-development-models/#when-to-use-the-agile-model)
* [**Advantages of the Agile Model**](https://www.geeksforgeeks.org/software-engineering-agile-development-models/#advantages-of-the-agile-model)
* [**Disadvantages of the Agile Model**](https://www.geeksforgeeks.org/software-engineering-agile-development-models/#disadvantages-of-the-agile-model)
* [**Questions For Practice**](https://www.geeksforgeeks.org/software-engineering-agile-development-models/#questions-for-practice)
* [**Conclusion**](https://www.geeksforgeeks.org/software-engineering-agile-development-models/#conclusion)
* [**Frequently Asked Questions on Agile Model – FAQs**](https://www.geeksforgeeks.org/software-engineering-agile-development-models/#frequently-asked-questions-on-agile-model-faqs)

**What is Agile Model?**

**The Agile Model was primarily designed to help a project adapt quickly to change requests. So, the main aim of the Agile model is to facilitate quick project completion. To accomplish this task, agility is required. Agility is achieved by fitting the process to the project and removing activities that may not be essential for a specific project. Also, anything that is a waste of time and effort is avoided. The Agile Model refers to a group of development processes. These processes share some basic characteristics but do have certain subtle differences among themselves.**

**Agile SDLC Models/Methods**

**Given below are some Agile SDLC Models:**

* **Crystal Agile methodology: The**[**Crystal Agile Software Development Methodology**](https://www.geeksforgeeks.org/crystal-methods-in-agile-development-framework/)**places a strong emphasis on fostering effective communication and collaboration among team members, as well as taking into account the human elements that are crucial for a successful development process. This methodology is particularly beneficial for projects with a high degree of uncertainty, where requirements tend to change frequently.**
* **Dynamic Systems Development Method (DSDM):**[**DSDSM methodology**](https://www.geeksforgeeks.org/dynamic-systems-development-method-dsdm/)**is tailored for projects with moderate to high uncertainty where requirements are prone to change frequently. Its clear-cut roles and responsibilities focus on delivering working software in short time frames. Governance practices set it apart and make it an effective approach for teams and projects.**
* **Feature-driven development (FDD):**[**FDD**](https://www.geeksforgeeks.org/fdd-full-form/)**approach is implemented by utilizing a series of techniques, like creating feature lists, conducting model evaluations, and implementing a design-by-feature method, to meet its goal. This methodology is particularly effective in ensuring that the end product is delivered on time and that it aligns with the requirements of the customer.**
* **Scrum:**[**Scrum methodology**](https://www.geeksforgeeks.org/scrum-software-development/)**serves as a framework for tackling complex projects and ensuring their successful completion. It is led by a Scrum Master, who oversees the process, and a Product Owner, who establishes the priorities. The Development Team, accountable for delivering the software, is another key player.**
* **Extreme Programming (XP):**[**Extreme Programming**](https://www.geeksforgeeks.org/software-engineering-extreme-programming-xp/)**uses specific practices like pair programming, continuous integration, and test-driven development to achieve these goals. Extreme programming is ideal for projects that have high levels of uncertainty and require frequent changes, as it allows for quick adaptation to new requirements and feedback.**
* **Lean Development:**[**Lean Development**](https://www.geeksforgeeks.org/lean-software-development-lsd/)**is rooted in the principles of lean manufacturing and aims to streamline the process by identifying and removing unnecessary steps and activities. This is achieved through practices such as continuous improvement, visual management, and value stream mapping, which helps in identifying areas of improvement and implementing changes accordingly.**
* **Unified Process:**[**Unified Process**](https://www.geeksforgeeks.org/rup-and-its-phases/)**is a methodology that can be tailored to the specific needs of any given project. It combines elements of both waterfall and Agile methodologies, allowing for an iterative and incremental approach to development. This means that the UP is characterized by a series of iterations, each of which results in a working product increment, allowing for continuous improvement and the delivery of value to the customer.**

**All**[**Agile Software Development Methodology**](https://www.geeksforgeeks.org/what-is-agile-framework-and-methodology-in-software-development)**discussed above share the same core values and principles, but they may differ in their implementation and specific practices. Agile development requires a high degree of collaboration and communication among team members, as well as a willingness to adapt to changing requirements and feedback from customers.**

**In the Agile model, the requirements are decomposed into many small parts that can be incrementally developed. The Agile model adopts Iterative development. Each incremental part is developed over an iteration. Each iteration is intended to be small and easily manageable and can be completed within a couple of weeks only. At a time one iteration is planned, developed, and deployed to the customers. Long-term plans are not made.**

**Steps in the Agile Model**

**The agile model is a combination of iterative and incremental process models. The steps involve in agile**[**SDLC models**](https://www.geeksforgeeks.org/software-development-life-cycle-sdlc/)**are:**

* [**Requirement gathering**](https://www.geeksforgeeks.org/requirements-gathering-introduction-processes-benefits-and-tools)
* **Design the Requirements**
* **Construction / Iteration**
* [**Testing**](https://www.geeksforgeeks.org/software-testing-basics/)**/**[**Quality Assurance**](https://www.geeksforgeeks.org/software-engineering-software-quality-assurance/)
* [**Deployment**](https://www.geeksforgeeks.org/software-deployment-in-software-development/)
* **Feedback**

***Steps in Agile Model***

1. **Requirement Gathering:- In this step, the development team must gather the requirements, by interaction with the customer. development team should plan the time and effort needed to build the project. Based on this information you can evaluate technical and economical feasibility.**
2. **Design the Requirements:- In this step, the development team will use user-flow-diagram or high-level**[**UML diagrams**](https://www.geeksforgeeks.org/unified-modeling-language-uml-introduction)**to show the working of the new features and show how they will apply to the existing software. Wireframing and designing user interfaces are done in this phase.**
3. **Construction / Iteration:- In this step, development team members start working on their project, which aims to deploy a working product.**
4. **Testing / Quality Assurance:- Testing involves**[**Unit Testing,**](https://www.geeksforgeeks.org/unit-testing-software-testing)[**Integration Testing**](https://www.geeksforgeeks.org/software-engineering-integration-testing)**, and**[**System Testing.**](https://www.geeksforgeeks.org/system-testing)**A brief introduction of these three tests is as follows:**
   * **Unit Testing:- Unit testing is the process of checking small pieces of code to ensure that the individual parts of a program work properly on their own. Unit testing is used to test individual blocks (units) of code.**
   * **Integration Testing:- Integration testing is used to identify and resolve any issues that may arise when different units of the software are combined.**
   * **System Testing:- Goal is to ensure that the software meets the requirements of the users and that it works correctly in all possible scenarios.**
5. **Deployment:- In this step, the development team will deploy the working project to end users.**
6. **Feedback:- This is the last step of the Agile Model. In this, the team receives feedback about the product and works on correcting bugs based on feedback provided by the customer.**

**The time required to complete an iteration is known as a Time Box. Time-box refers to the maximum amount of time needed to deliver an iteration to customers. So, the end date for an iteration does not change. However, the development team can decide to reduce the delivered functionality during a Time-box if necessary to deliver it on time. The Agile model’s central principle is delivering an increment to the customer after each Time-box.**

**Principles of the Agile Model**

* **To establish close contact with the customer during development and to gain a clear understanding of various requirements, each Agile project usually includes a customer representative on the team. At the end of each iteration stakeholders and the customer representative review, the progress made and re-evaluate the requirements.**
* **The agile model relies on working software deployment rather than comprehensive documentation.**
* **Frequent delivery of incremental versions of the software to the customer representative in intervals of a few weeks.**
* **Requirement change requests from the customer are encouraged and efficiently incorporated.**
* **It emphasizes having efficient team members and enhancing communications among them is given more importance. It is realized that improved communication among the development team members can be achieved through face-to-face communication rather than through the exchange of formal documents.**
* **It is recommended that the development team size should be kept small (5 to 9 people) to help the team members meaningfully engage in face-to-face communication and have a collaborative work environment.**
* **The agile development process usually deploys**[**Pair Programming**](https://www.geeksforgeeks.org/pair-programming)**. In Pair programming, two programmers work together at one workstation. One does coding while the other reviews the code as it is typed in. The two programmers switch their roles every hour or so.**

**Characteristics of the Agile Process**

* **Agile processes must be adaptable to technical and environmental changes. That means if any technological changes occur, then the agile process must accommodate them.**
* **The development of agile processes must be incremental. That means, in each development, the increment should contain some functionality that can be tested and verified by the customer.**
* **The customer feedback must be used to create the next increment of the process.**
* **The software increment must be delivered in a short span of time.**
* **It must be iterative so that each increment can be evaluated regularly.**

**When To Use the Agile Model?**

* **When frequent modifications need to be made, this method is implemented.**
* **When a highly qualified and experienced team is available.**
* **When a customer is ready to have a meeting with the team all the time.**
* **when the project needs to be delivered quickly.**
* **Projects with few regulatory requirements or not certain requirements.**
* **projects utilizing a less-than-strict current methodology**
* **Those undertakings where the product proprietor is easily reachable**
* **Flexible project schedules and budgets.**

**Advantages of the Agile Model**

* **Working through Pair programming produces well-written compact programs which have fewer errors as compared to programmers working alone.**
* **It reduces the total development time of the whole project.**
* **Agile development emphasizes face-to-face communication among team members, leading to better collaboration and understanding of project goals.**
* **Customer representatives get the idea of updated software products after each iteration. So, it is easy for him to change any requirement if needed.**
* **Agile development puts the customer at the center of the development process, ensuring that the end product meets their needs.**

**Disadvantages of the Agile Model**

* **The lack of formal documents creates confusion and important decisions taken during different phases can be misinterpreted at any time by different team members.**
* **It is not suitable for handling complex dependencies.**
* **The agile model depends highly on customer interactions so if the customer is not clear, then the development team can be driven in the wrong direction.**
* **Agile development models often involve working in short sprints, which can make it difficult to plan and forecast project timelines and deliverables. This can lead to delays in the project and can make it difficult to accurately estimate the costs and resources needed for the project.**
* **Agile development models require a high degree of expertise from team members, as they need to be able to adapt to changing requirements and work in an iterative environment. This can be challenging for teams that are not experienced in agile development practices and can lead to delays and difficulties in the project.**
* **Due to the absence of proper documentation, when the project completes and the developers are assigned to another project, maintenance of the developed project can become a problem.**

**Questions For Practice**

**1. Which of the following is not a key issue stressed by an agile philosophy of software engineering? [**[**UGC NET CS 2017**](https://www.geeksforgeeks.org/questions/ugc-net-ugc-net-cs-2017-dec-2-question-43/)**]**

**(A) The importance of self-organizing teams as well as communication and collaboration between team members and customers**

**(B) Recognition that change represents an opportunity**

**(C) Emphasis on rapid delivery of software that satisfies the customer**

**(D) Having a separate testing phase after a build phase**

**Solution: Correct Answer is (D).**

**2. ​Which of the following is not one of the principles of the agile software development method? [**[**UGC NET CS 2018**](https://www.geeksforgeeks.org/questions/ugc-net-ugc-net-cs-2018-dec-ii-question-95/)**]**

**(A) Following the plan**

**(B) Embrace change**

**(C) Customer involvement**

**(D) Incremental delivery**

**Solution: Correct Answer is (A).**

**Conclusion**

**Agile development models prioritize flexibility, collaboration, and customer satisfaction. They focus on delivering working software in short iterations, allowing for quick adaptation to changing requirements. While Agile offers advantages like faster delivery and customer involvement, it may face challenges with complex dependencies and lack of formal documentation. Overall, Agile is best suited for projects requiring rapid development, continuous feedback, and a highly skilled team.**

**Frequently Asked Questions on Agile Model – FAQs**

**1. What is Product Backlog in Agile?**

***Product Backlog simply refers to the list of features, and tasks that are to be developed in Software Product. These things are continuously monitored and managed by the Product Owner.***

**2. Is it possible to use Agile Model for large and complex Projects?**

***Yes, it is possible to use for Large and Complex Projects, but we need to change some adaptations for using it like we have to add some frameworks like SAFE (***[***Scaled Agile Framework***](https://www.geeksforgeeks.org/software-testing-safe-methodology)***) and LeSS (Large-Scale Scrum) to use large and complex projects.***

**3. What is Sprint Review in Agile Model?**

***Sprint Review is simply a type of meeting that is held at the end of each sprint in the Agile Model. In this meeting, the development team details the work done to the stakeholder and Product Owner.***

**Related Posts:**

* [**Agile Software Development Methodology | Framework, Principles, and Benefits**](https://www.geeksforgeeks.org/what-is-agile-framework-and-methodology-in-software-development)

**Want to learn Software Testing and Automation to help give a kickstart to your career? Any student or professional looking to excel in Quality Assurance should enroll in our course,**[***Complete Guide to Software Testing and Automation***](https://gfgcdn.com/tu/Q6s/)**, only on GeeksforGeeks. Get hands-on learning experience with the latest testing methodologies, automation tools, and industry best practices through practical projects and real-life scenarios. Whether you are a beginner or just looking to build on existing skills, this course will give you the competence necessary to ensure the quality and reliability of software products. Ready to be a Pro in Software Testing? Enroll now and Take Your Career to a Whole New Level!**

**2. What is ITIL?**

ITIL stands for Information Technology Infrastructure Library. It’s actually used to guide it team to improve the value of service by focusing on solving the business issues and creating business value. One of the important parts of ITIL is the configuration management database which provides services, users, software, hardware, documents, and IT components to the central authority. ITIL has been adopted by thousands of organizations worldwide like Microsoft, NASA, and HSBC. ITIL’s current version which is use in the market is ITIL4, it provides a practical and flexible basis to support organizations for a digital transformation.

**Misconceptions about DevOps vs. ITIL :**

1. **DevOps can replace ITIL:** There are essential business functions like service Management, support, operation, governance, and costing these processes are important and they both given by DevOps and ITIL, only DevOps or ITIL can’t provide them.
2. **ITIL is always documentation:** It’s actually not true information ITIL is what the team makes the choice the rules are made by team, ITIL has Its own guidelines but the decision is always made by the IT team.
3. **ITIL is only for large companies:**It’s true that ITIL is used by large enterprises but Its guidelines can also benefit small businesses, small business only needs to know how to handle change management, major incident, and knowledge management.

**Use cases for DevOps and ITIL :**

IT needs both ITIL and DevOps elements.

DevOps is an automated development, collaboration, and blame-free culture.

It’s better collaboration between IT and business when we use ITIL and DevOps together and it also helps in customer satisfaction due to the improved delivery of services and better management of issues. ITIL provides a tried-and-true starting point, where DevOps can add improvement with the blameless examination, automation, and other collaborative approaches which makes every IT process easier.

**Difference between DevOps and ITIL :**

|  |  |  |
| --- | --- | --- |
| **S.NO.** | **DevOps** | **ITIL** |
| 01. | DevOps refers to effective collaboration between the development team and operations team. | ITIL refers to a set of detailed guidelines for effective and efficient management of an organization’s IT services. |
| 02. | Use a methodical approach to minimize the friction between two teams. | It uses a systematic approach to manage the IT service to ensure growth. |
| 03. | CI and CD are the backbone of modern DevOps philosophy. | ITIL aims to increase the delivery process. |
| 04. | Continuous integration and continuous delivery are critical to increasing. | Services are built, discuss, tested, and implemented. |
| 05. | DevOps focus on the concept. It has a dynamic body of knowledge. | DevOps focus on the concept. It has a dynamic body of knowledge. ITIL focus on development. It has a static body of knowledge. |