

Fundamentals of Software Development Lifecycle/ Application Lifecycle Management

Overview

Software Development Life Cycle (SDLC) is a process used by the software industry to

design, develop and test high quality softwares. The SDLC aims to produce high-quality software that meets or exceeds customer expectations, reaches completion within time and cost estimates.

Goals

1. The goal of the SDLC is to produce superior software that meets and exceeds all customer expectations and demands.
2. The SDLC defines and outlines a detailed plan with stages, or phases, that each encompass their own process and deliverables.

Modules

1. Fundamentals of Web Applications
2. Introduction to Software Development Life Cycle (Application Lifecycle Management)
3. Essential Technologies of SDLC
4. Execution of Software Development Life Cycle (Application Lifecycle Management)

Module -1

Fundamentals of Web Applications

1. Web Application Architecture
2. Web Technologies
3. Web Technologies used in Projects

Web Application Architecture:

A web app architecture presents a layout with all the software components (such as databases, applications and middleware) and how they interact with each other. The modern day applications contain three-tiered architecture. Three-tier architecture, which separates applications into three logical computing tiers, is the predominant software architecture for traditional client-server applications. For decades three-tier architecture was the prevailing architecture for client-server applications.

The three tiers of applications in this architecture are:

- the presentation tier, or user interface;
- the application tier, where data is processed;
- the data tier, where the data associated with the application is stored and managed.

Web Technologies

Presentation tier:

The presentation tier is the user interface and communication layer of the application, where the end user interacts with the application.

- Its main purpose is to display information to and collect information from the user. This top-level tier can run on a web browser, as a desktop application, or a graphical user interface (GUI), for example.

- Web presentation tiers are usually developed using Hypertext Markup Language (HTML) and Cascading Style Sheets (CSS) and JavaScript.
- HTML tells a browser how to display the content of web pages, while CSS styles that content.

Application tier:

The application tier, also known as the logic tier or middle tier, is the heart of the application. In this tier, information collected in the presentation tier is processed - sometimes against other information in the data tier - using business logic, a specific set of business rules. The application tier can also add, delete or modify data in the data tier. The application tier is typically developed using React JS, Angular JS, Python, Java, (Salesforce), Express JS, Node JS and communicates with the data tier using API calls.

Data tier:

The data tier, sometimes called database tier, data access tier or back-end, is where the information processed by the application is stored and managed. This can be a relational database management system such as PostgreSQL, MySQL, MariaDB, Oracle, DB2, Informix or Microsoft SQL Server, or in a NoSQL Database server such as MongoDB, Cassandra, Couch DB.

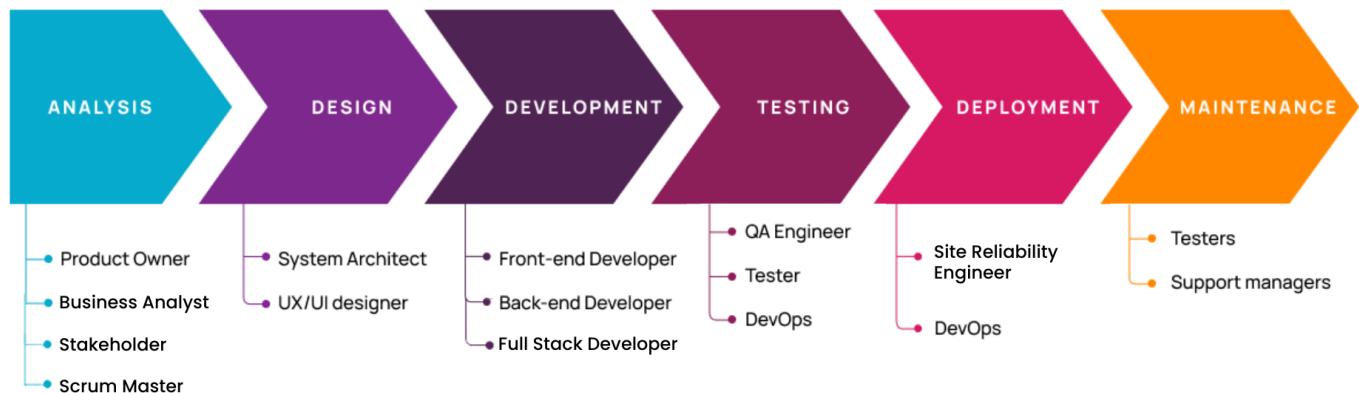
In a three-tier application, all communication goes through the application tier. The presentation tier and the data tier cannot communicate directly with one another.

Benefits:

- Faster development
- Improved scalability
- Improved reliability
- Improved security

Module - 2

Introduction to Software Development Life Cycle (Application Lifecycle Management)



- What is SDLC?
- SDLC Methodologies
- Waterfall Methodology
- Agile Methodology
- Roles involved in the process of Software Development Life Cycle/ Application Lifecycle Management
- Scrum Framework

What is the Software Development Life Cycle?

A software development life cycle (SDLC) is a methodology followed to create high-quality software. By adhering to a standard set of tools, processes, and duties, a software development team can build, design, and develop products that meet or exceed their clients' expectations.

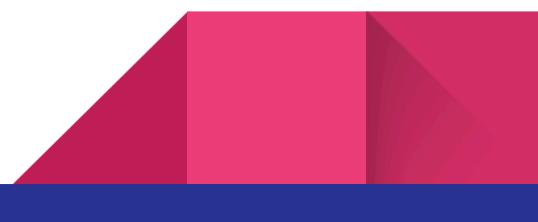
SDLC Methodologies:

- Waterfall
- Lean
- Iterative
- Spiral
- Agile

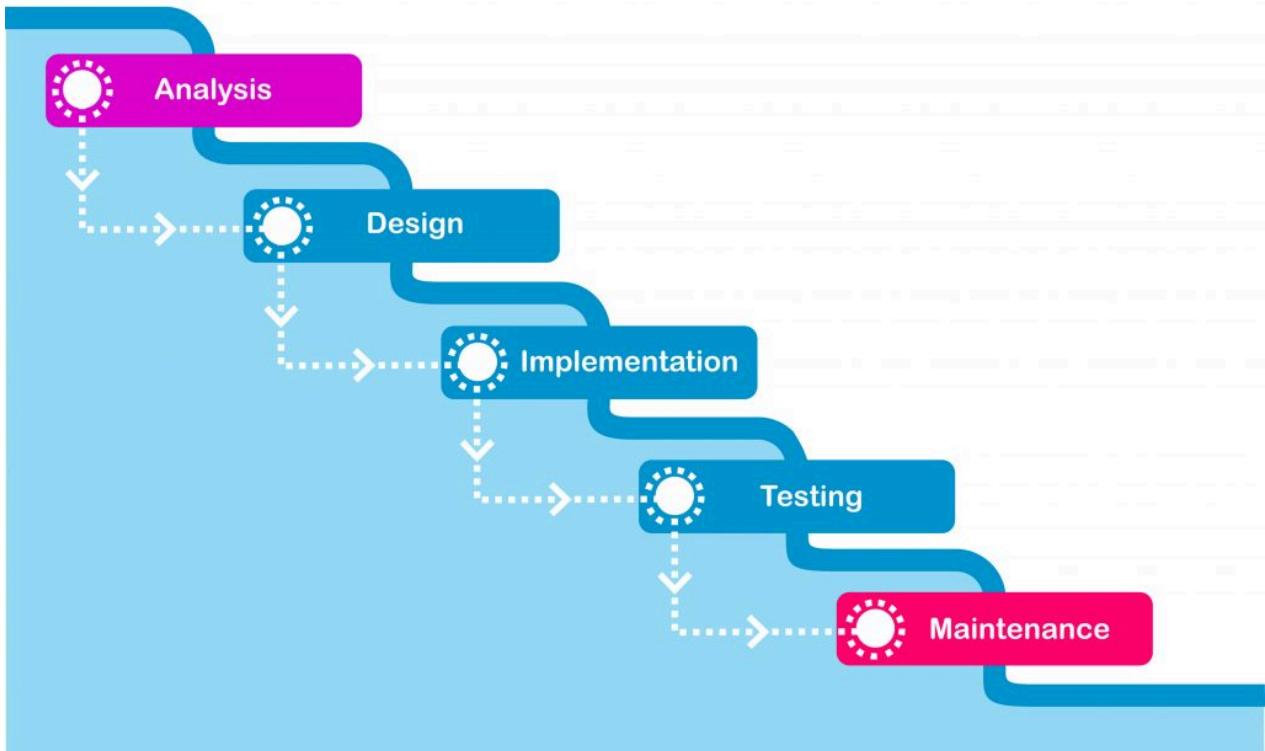
Each of these approaches varies in some ways from the others, but all have a common purpose: to help teams deliver high-quality software as quickly and cost-effectively as possible. Out of all these, the two most common methodologies are Waterfall and Agile.

Waterfall Methodology:

The Waterfall methodology—is a sequential development process that flows like a waterfall through all phases of a project (analysis, design, development, and testing, for example), with each phase completely wrapping up before the next phase begins.



WATERFALL



Disadvantages of Waterfall Methodology:

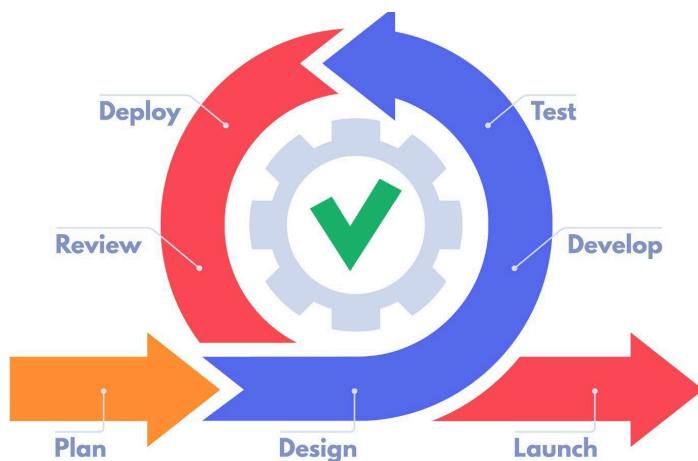
The Waterfall methodology's insistence on upfront project planning and commitment to a certain defined progress means that it is less flexible, other reasons the Waterfall methodology may not work include:

- Projects can take longer to deliver.
- Clients are not involved in the design and implementation stages.
- Deadline creep—when one phase in the process is delayed, all the other phases are delayed.

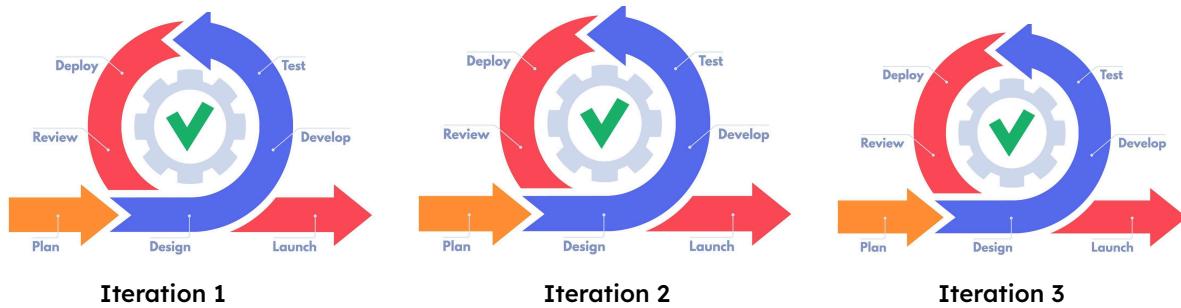
Agile Methodology:

Agile is an iterative approach to project management and software development that helps teams deliver value to their customers faster and with fewer headaches. Instead of betting everything on a "big bang" launch, work in small agile team delivers all, but consumable, increments. Requirements, plans, and results are evaluated continuously so teams have a natural mechanism for responding to change quickly.

Agile Single Iteration:



Agile Development Cycle:



Advantages of Agile Methodology:

1. Satisfy the client via early, continuous delivery of valuable software.
2. Adapt to changing requirements, even late in development.
3. Use a shorter timescale (between a couple of weeks to a couple of months) for delivering working software.
4. Developers and business people should work together on a daily basis.
5. Motivate individuals by giving them an ideal environment, support, and trust.
6. Convey information via face-to-face communication.
7. Use working software as the main measurement of progress.
8. Maintain a constant pace and promote sustainable development.
9. Pay continuous attention to good design and technical excellence.
10. Keep things as simple as possible and allow the teams to self-organize.
11. Regularly reflect on how your team can be more effective and adjust accordingly.

The Process of Software Development Life Cycle/ Application Lifecycle Management (Roles):

Business Analyst: A business analyst is a person who processes, interprets and documents business processes, products, services and software through analysis of data. The role of a business analyst is to ensure business efficiency increases through their knowledge of both IT and business function.

System Architect: The systems architect is an information and communications technology professional. Systems architects define the architecture of a computerized system in order to fulfill certain requirements.

Developer (Front End/Back End): Front and back end developers work on different sides of a website. Front end development is programming which focuses on the visual elements of a website or app that a user will interact with (the client side). Meanwhile, back end development focuses on the side of a website users can't see (the server side).

Tester/QA Engineer: A tester is the person responsible for trying out all the different usage scenarios for an application in order to ensure that it works as expected. The tester reports back to the development team listing the bugs that were discovered and what series of actions led to that particular error.

DevOps Engineer: A DevOps Engineer releases engineering including the work required to build and deploy application code. The exact tools and processes vary widely depending on many variables, such as what language the code is written in, how much of the pipeline has been automated, and whether the production infrastructure is on-premise or in the cloud.

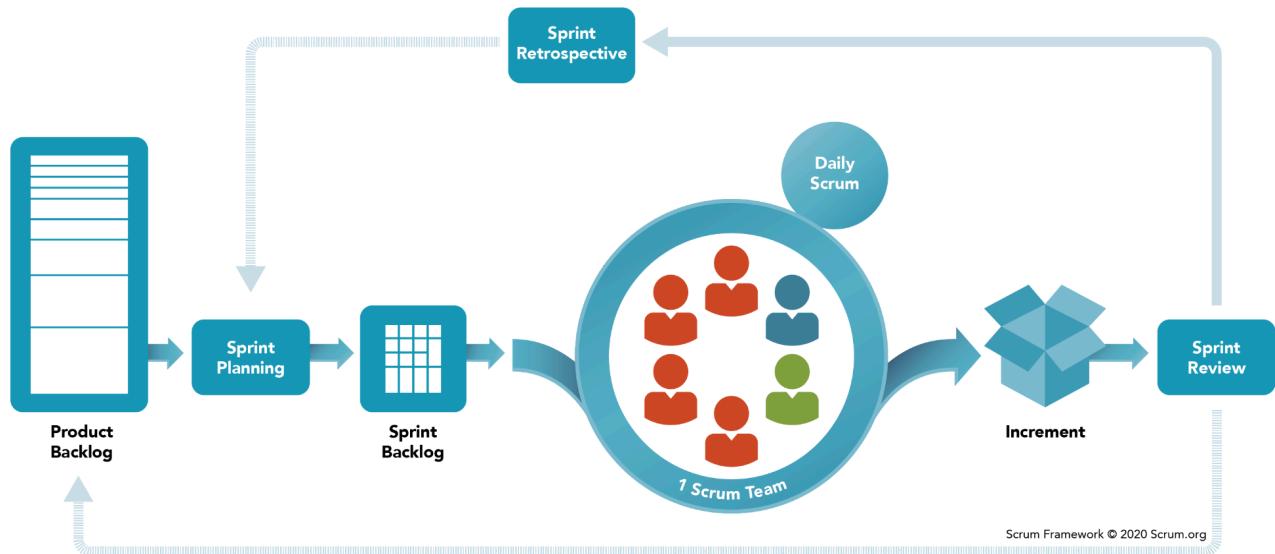
Support Managers: A support manager lays out the query escalation process for troubleshooting and to ensure that the helpdesk system remains fit for purpose as the business' technology evolves.

Scrum Framework:

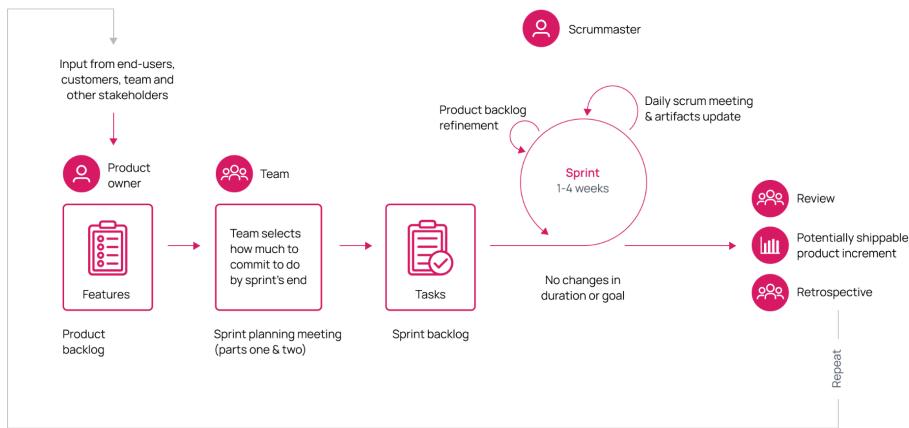
Scrum is a framework within which people can address complex adaptive problems, while productively and creatively delivering products of the highest possible. The Scrum Team consists of one Scrum Master, one Product Owner, and Developers.

In a nutshell, Scrum requires a Scrum Master to foster an environment where:

1. Product Owner orders the work for a complex problem into a Product Backlog.
2. The Scrum Team turns a selection of the work into an Increment of value during a Sprint.
3. The Scrum Team and its stakeholders inspect the results and adjust for the next Sprint.
4. Repeat



A Scrum Framework usually consists of 5 different phases:



Scrum Terminology:

Scrum Team:

A Scrum Team is expected to adapt the moment it learns anything new through Inspection. The Scrum Team commits to achieving its goals and to supporting each other. Scrum Team members respect each other to be capable, independent people, and are respected as such by the people with whom they work.

Scrum Master

Scrum Masters are true leaders who serve the Scrum Team and the larger organization. The Scrum Master serves the Scrum Team in several ways, including coaching the team members in self-management and cross-functionality. The Scrum Master serves the Scrum Team in several ways, including helping the Scrum Team focus on creating high-value Increments that meet the Definition of the planning.

Sprint Planning:

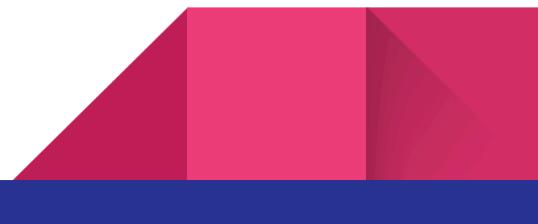
The Scrum Master serves the Scrum Team in several ways, including ensuring that all Scrum events take place and are positive, productive, and kept within the timebox. This resulting plan is created by the collaborative work of the entire Scrum Team. The whole Scrum Team then collaborates to define a Sprint Goal that communicates why the Sprint is valuable to stakeholders.

Sprints:

The Sprint is a container for all other events. Sprints are the heartbeat of Scrum, where ideas are turned into value. [Sprints] are fixed length events of one month or less to create consistency. A new Sprint starts immediately after the conclusion of the previous Sprint. All the work necessary to achieve the Product Goal, including Sprint Planning, Daily Scrums, Sprint Review, and Sprint Retrospective, happen within Sprints.

Sprint Review:

The Sprint Review is a working session and the Scrum Team should avoid limiting it to a presentation. The Scrum Team discusses what went well during the Sprint, what problems it encountered, and how those problems were (or were not) solved.



Product Backlog:

The Product Backlog is the single source of work undertaken by the Scrum Team. 10 Product Backlog items that can be Done by the Scrum Team within one Sprint are deemed ready for selection in a Sprint Planning event.

Product Goal and Sprint Goal:

The Product Goal is the long-term objective for the Scrum Team. They must fulfill (or abandon) one objective before taking on the next. The Sprint Goal also creates coherence and focus, encouraging the Scrum Team to work together rather than on separate initiatives.

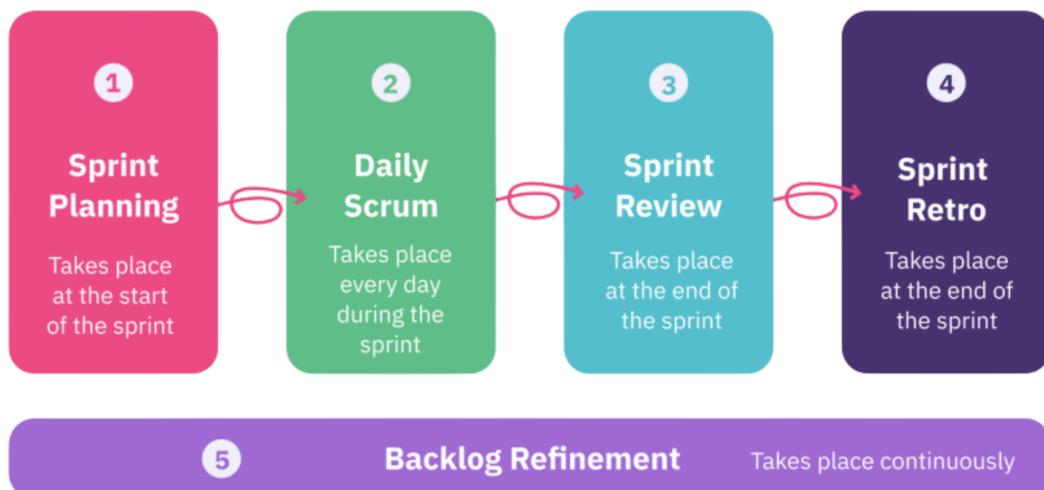
Daily Scrum:

The purpose of the Daily Scrum is to inspect progress toward the Sprint Goal and adapt the Sprint Backlog as necessary, adjusting the upcoming planned work. The Daily Scrum is a 15-minute event for the Developers of the Scrum Team. To reduce complexity, [the Daily Scrum] is held at the same time and place every working day of the Sprint.

Sprint Retrospective:

The purpose of the Sprint Retrospective is to plan ways to increase quality and effectiveness. The Scrum Team inspects how the last Sprint went with regards to individuals, interactions, processes, tools, and their Definition of Done.

The 5 Scrum Ceremonies



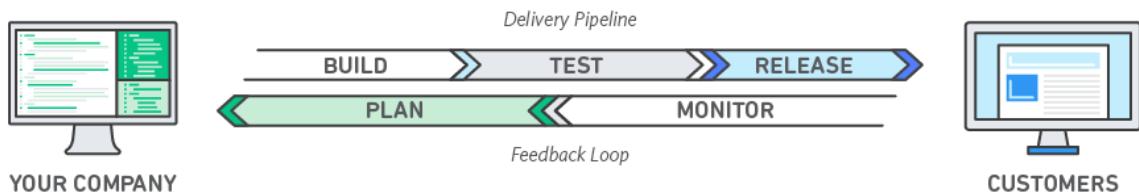
Module - 3

Essential Technologies of SDLC

- What is DevOps?
- What is Cloud, SAAS, IAAS, PAAS?
- What is Testing?

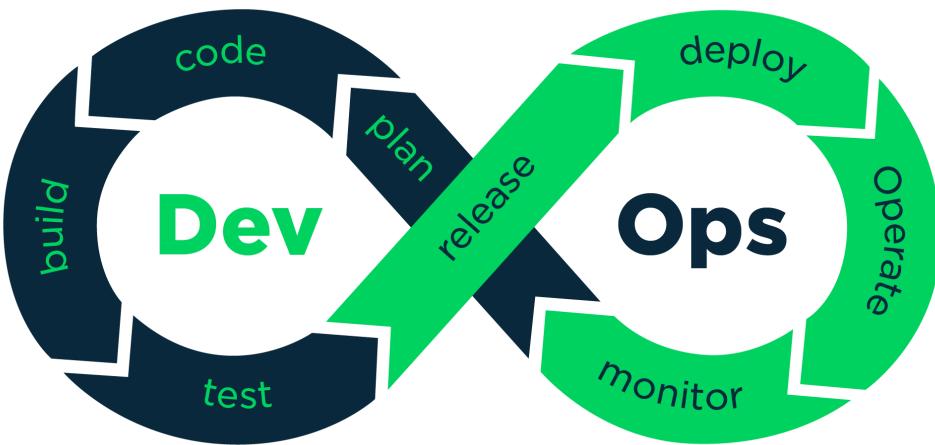
What is DevOps?

DevOps is the combination of cultural philosophies, practices, and tools that increases an organization's ability to deliver applications and services at high velocity: evolving and improving products at a faster pace than organizations using traditional software development and infrastructure management processes. This speed enables organizations to better serve their customers and compete more effectively in the market.



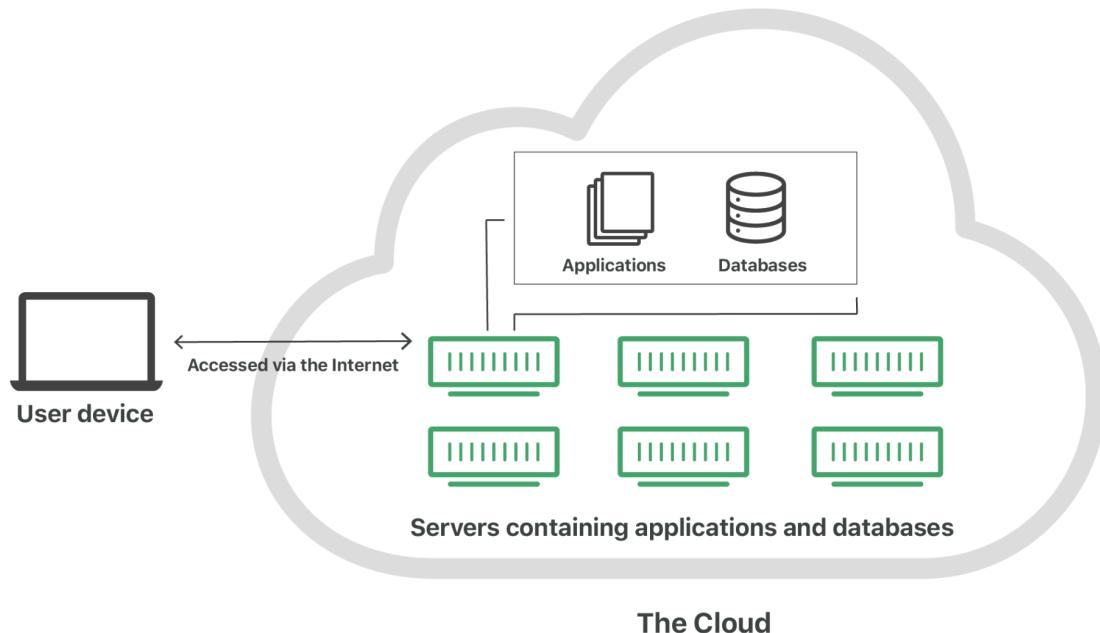
What does a DevOps Engineer do?

A DevOps engineer introduces processes, tools, and methodologies to balance needs throughout the software development life cycle, from coding and deployment, to maintenance and updates.



What is Cloud - SAAS, IAAS, PAAS?

"The cloud" refers to servers that are accessed over the Internet, and the software and databases that run on those servers. Cloud servers are located in data centers all over the world. By using cloud computing, users and companies do not have to manage physical servers themselves or run software applications on their own machines.



Main models of Cloud Computing:

Software-as-a-Service (SaaS): SaaS applications are hosted on cloud servers, and users access them over the Internet. SaaS is like renting a house: the landlord maintains the house, but the tenant mostly gets to use it as if they owned it.

Examples: Salesforce, MailChimp, and Slack.

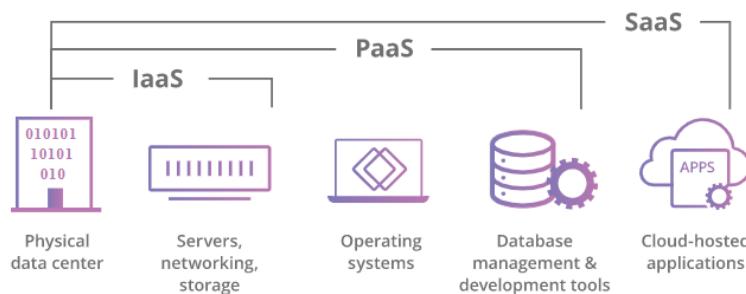
Platform-as-a-Service (PaaS): In this model, companies don't pay for hosted applications; instead they pay for the things they need to build their own applications. PaaS vendors

offer everything necessary for building an application, including development tools, infrastructure, and operating systems, over the Internet. PaaS can be compared to renting all the tools and equipment necessary for building a house, instead of renting the house itself.

Examples: Heroku and Microsoft Azure.

Infrastructure-as-a-Service (IaaS): In this model, a company rents the servers and storage they need from a cloud provider. They then use that cloud infrastructure to build their applications. IaaS is like a company leasing a plot of land on which they can build whatever they want — but they need to provide their own building equipment and materials.

Examples: DigitalOcean, Google Compute Engine, and OpenStack.

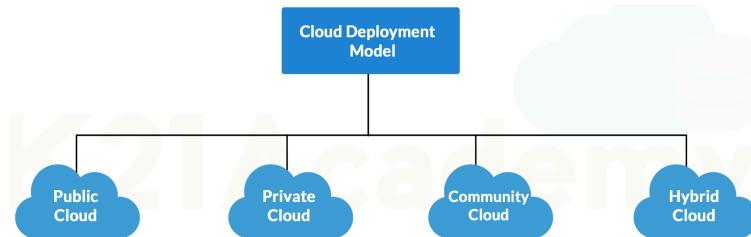


The most common cloud deployments are:

- **Private cloud:** A private cloud is a server, data center, or distributed network wholly dedicated to one organization.
- **Public cloud:** A public cloud is a service run by an external vendor that may include servers in one or multiple data centers. Unlike a private cloud, public clouds are shared by multiple organizations. Using virtual machines, individual servers may be shared by different companies, a situation that is called "multi tenancy" because multiple tenants are renting server space within the same server.
- **Hybrid cloud:** Hybrid cloud deployments combine public and private clouds, and may even include on-premises legacy servers. An organization may use their

private cloud for some services and their public cloud for others, or they may use the public cloud as backup for their private cloud.

- **Multi-cloud:** Multi-cloud is a type of cloud deployment that involves using multiple public clouds. In other words, an organization with a multi-cloud deployment rents virtual servers and services from several external vendors — to continue the analogy used above, this is like leasing several adjacent plots of land from different landlords.



What is Testing?

Software testing is the process of evaluating and verifying that a software product or application does what it is supposed to do. The benefits of testing include preventing bugs, reducing development costs and improving performance.

Three types of Testing include:

- Manual testing
- Automation testing
- Automation with Selenium testing

Manual testing:

Manual testing is a software testing process in which test cases are executed manually without using any automated tool. All test cases executed by the tester manually according to the end user's perspective. It ensures whether the application is working, as mentioned in the requirement document or not. Test cases are planned and implemented to complete

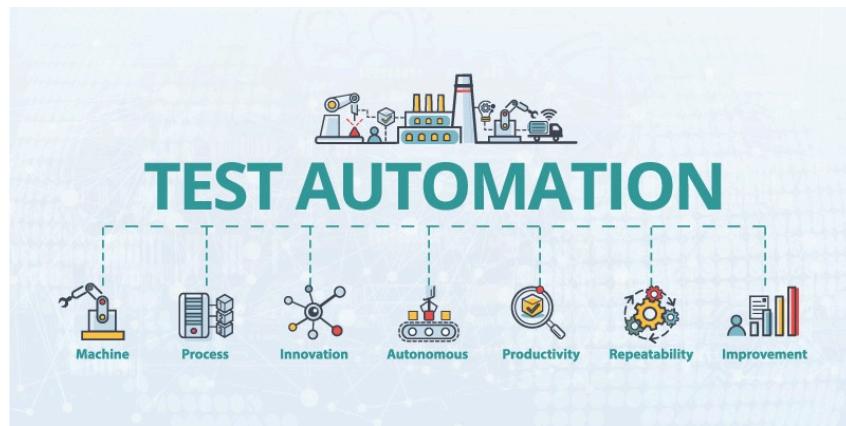
almost 100 percent of the software application. Test case reports are also generated manually.

Automation testing:

Automation Testing is a software testing technique that performs using special automated testing software tools to execute a test case suite. On the contrary, Manual Testing is performed by a human sitting in front of a computer carefully executing the test steps.

Automation with Selenium testing:

Selenium is a free (open-source) automated testing framework used to validate web applications across different browsers and platforms. You can use multiple programming languages like Java, C#, Python etc to create Selenium Test Scripts. Testing done using the Selenium testing tool is usually referred to as Selenium Testing.



Execution of Software Development Life Cycle (Application Lifecycle Management)

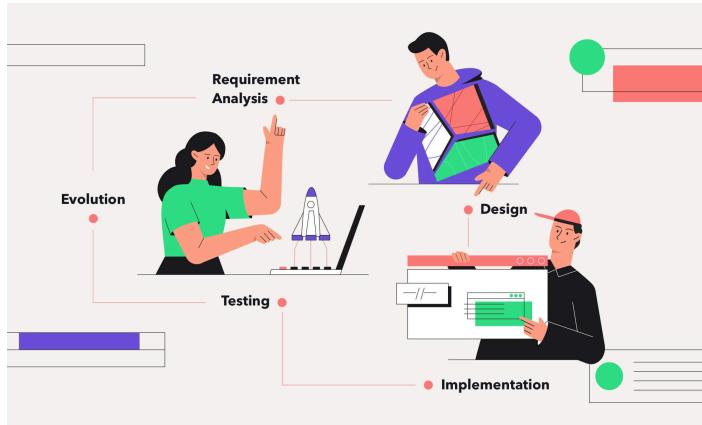
- Analysis: Azure Boards, Jira.
- Design: Photoshop, Illustrator, Figma
- Implementation: Front End Technologies, Back End Technologies, Databases and Frameworks
- Introduction to Azure DevOps
- Code Management: Azure Repos, Git and GitHub.
- Testing Release: Test Plans
- Maintenance: CI/CD Pipelines

#1 Planning and Analysis:

Requirements gathering is the first most crucial phase of the software development life cycle. It decides how your software will look and perform at the end. To build quality software, it is imperative to spend ample time in this phase. During this step, the project managers or business analysts meet with the customer and ask for the software specifications. Once you are done with the requirements gathering, you need to draw the project timeline and scope.

Steps of Planning and Analysis:

1. Defining business problem and scope
2. Producing detailed project schedule
3. Confirming project feasibility
4. Staffing the project with accurate resources
5. Planning the scrum based on the requirements
6. Launching the project



Tools required for Planning and Analysis:

The two major tools that are used for Planning and Analysis process in SDLC are

1. Azure Boards
2. Jira

#2 Design:

This type of website design is usually suited for small businesses that want to create a corporate identity without spending a lot of money on advertising. It does not have the visual appeal of other types of website design, but it does provide a way for the business to communicate its message or sales pitch without having to spend a lot of money.

Web designing is of three kinds, to be specific static, dynamic or CMS and eCommerce. Picking the sort of website design relies upon the kind of business and necessity of the entrepreneurs. Every one of these sites can be designed and developed on various platforms like *Figma*, *Adobe Illustrator* and *Adobe Premiere Pro*.

