



# AI + ML role in Testing & Revision

SE3010

Software Engineering Process and Quality Management

# Agenda

- What we covered Last Week...
- AI + ML role in Testing
  - What is AI?
    - ✓ Types of AI
    - ✓ Key Components of AI
    - ✓ AI Programming focuses on Cognitive Skills
    - ✓ Real-World Applications for AI systems Today
  - What is ML?
    - ✓ Difference between Machine Learning and Traditional Programming
    - ✓ Various Applications of Machine Learning
  - AI + ML Testing
    - ✓ Why is AI worth exploring for companies in software testing?
    - ✓ How AI and ML can be used to improve Software Testing
    - ✓ Use cases where AI and ML could assist during our Software Testing Process
    - ✓ Benefits of AI and ML for Software Testers
- Revision [Recap of Lessons...]
  - Automation
  - Software Engineering Process
  - Cloud Computing & Testing
  - Security Testing
- Recap
- Next Week...

# What we covered Last Week...

- Security Testing
  - What is Security Testing?
  - Principles of Security Testing
  - Key Areas in Security Testing
  - Types of Security Testing
  - Methodologies/ Approach / Techniques for Security Testing
  - Security Testing along with SDLC
  - Example Test Scenarios for Security Testing
  - Security Testing Tool
  - Myths and Facts of Security Testing
  - Advantages of Security Testing
  - Disadvantages of Security Testing
  - Key Roles Associated with Security Testing
- Cyber Security
  - What is Cyber Security?
  - Types of Cyber Threats
  - What is the impact of a Cyberattack?
  - Common Cyber Threats
  - Malware
  - What are Cyber Security best practices?

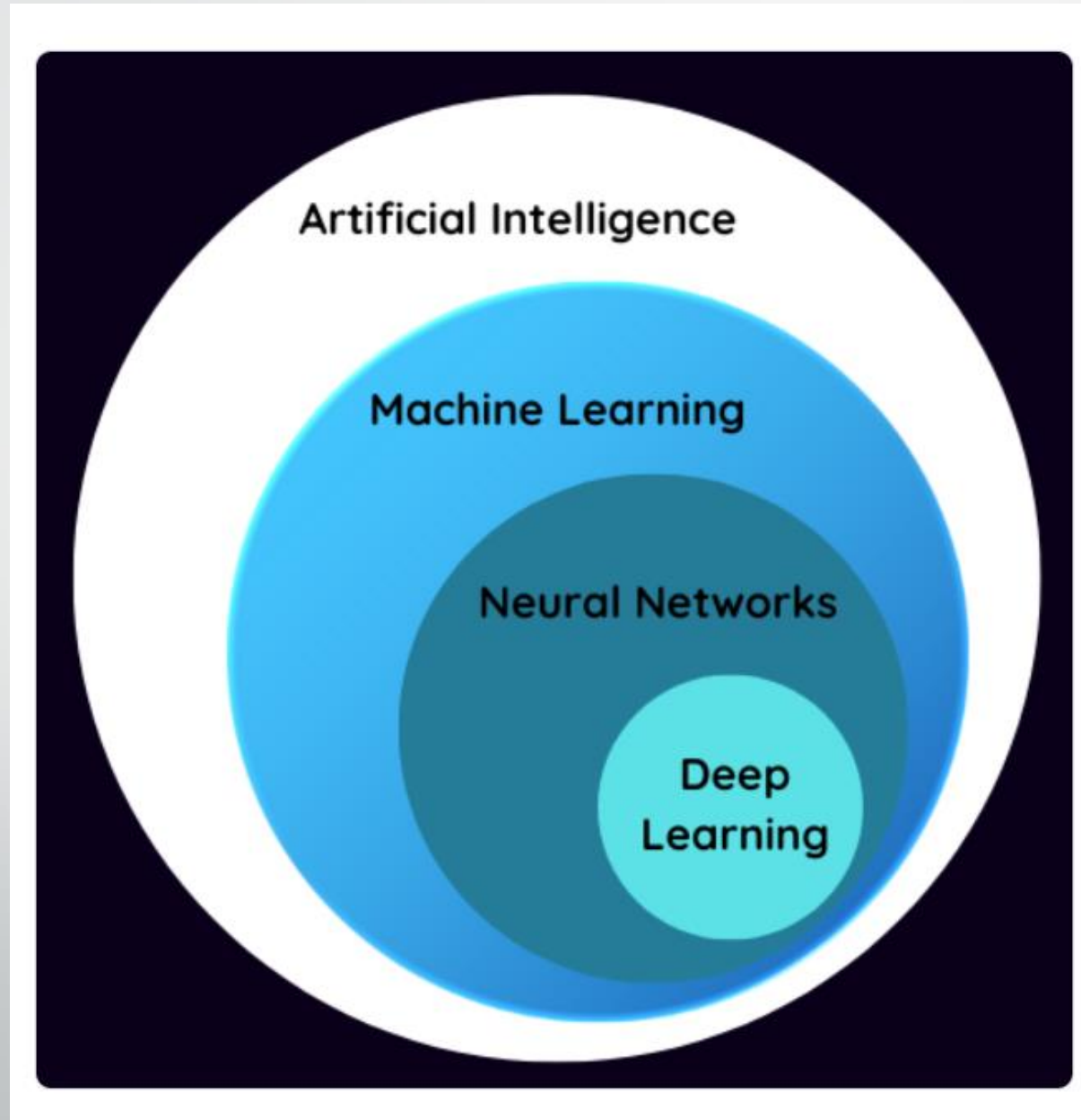


***What is AI?***

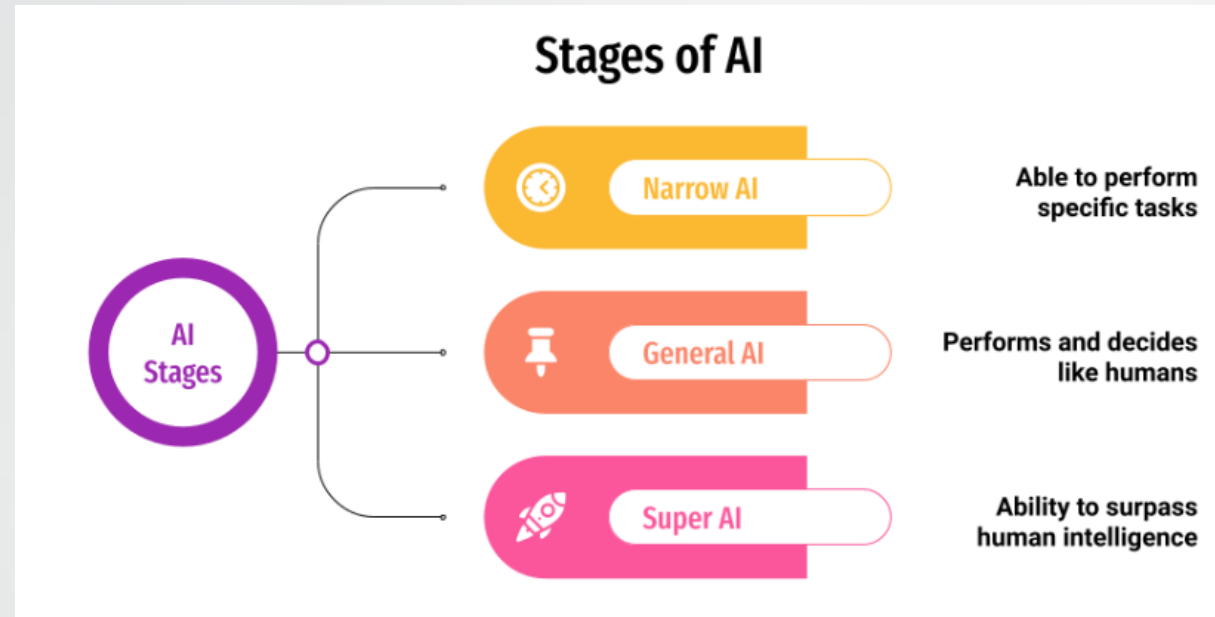
# Artificial Intelligence (AI)

- **Artificial intelligence** is the simulation of human intelligence processes by machines, especially computer systems. Specific applications of AI include expert systems, natural language processing, speech recognition and machine vision.
- Artificial intelligence, or AI, is technology that enables computers and machines to simulate human intelligence and problem-solving capabilities.
- On its own or combined with other technologies (e.g., sensors, geolocation, robotics) AI can perform tasks that would otherwise require human intelligence or intervention. Digital assistants, GPS guidance, autonomous vehicles, and generative AI tools (like Open AI's Chat GPT) are just a few examples of AI in the daily news and our daily lives.

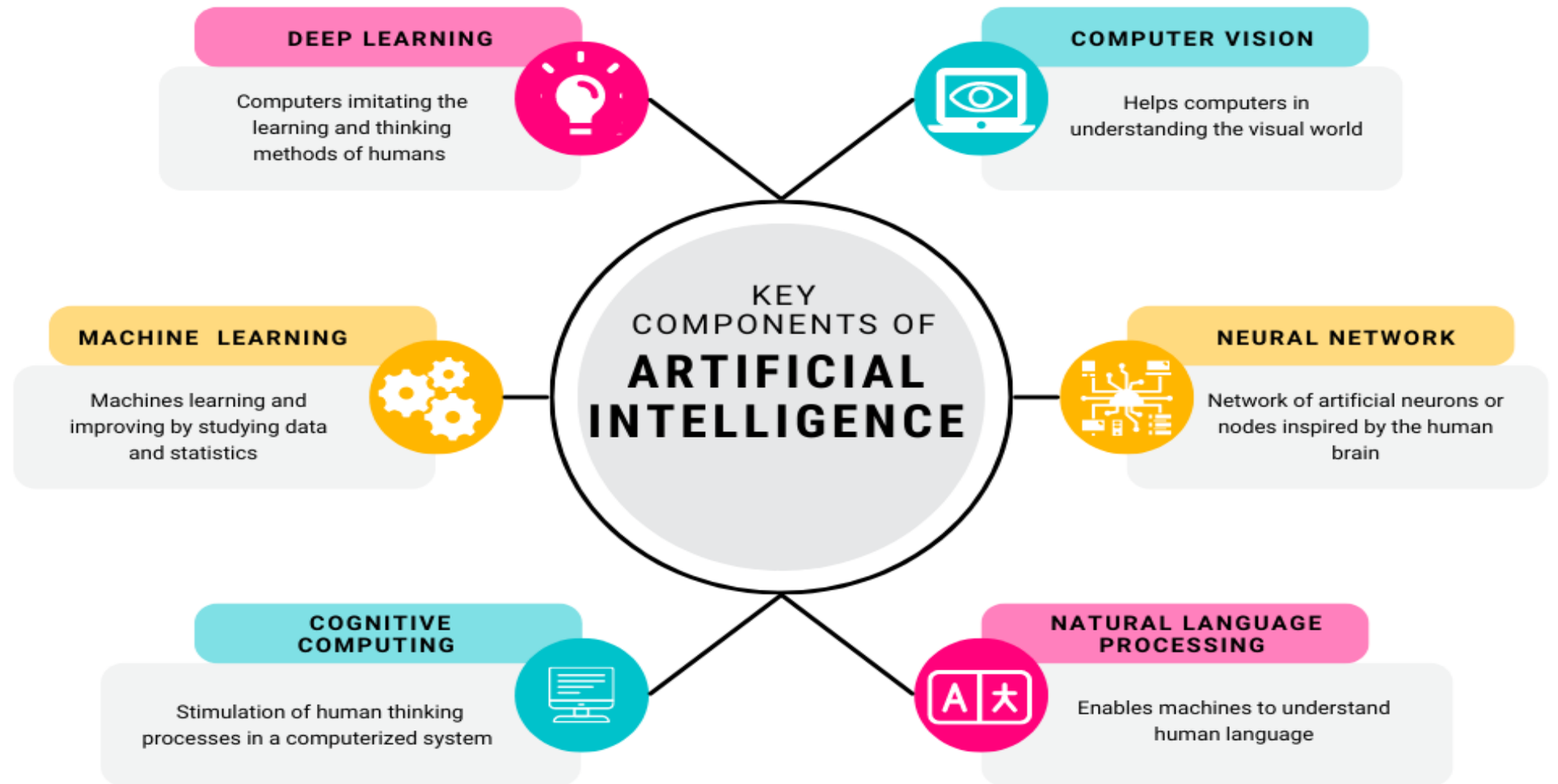
# Artificial Intelligence (AI)



# Types of Artificial Intelligence (AI)



- **Narrow AI:** Narrow AI or Weak AI refers to Artificial Intelligence systems that are built to perform only a specific set of tasks. Examples of Narrow AI include facial recognition, language translation, recommendation algorithms, and self-driving cars. To be specific, Apple's Siri is considered a Narrow or Weak AI.
- **General AI:** General AI (or Strong AI) is a sort of AI technology that can do a variety of jobs autonomously, exactly like a person. The objective of General AI is to create AI technology capable of matching human intellect. Lots of investments are done in this area by tech giants like Microsoft, and research is in full swing.
- **Super AI:** Super AI is a concept used to characterize artificial intelligence (AI) that is adept at outperforming human capabilities. Super AI is distinguished by exceptional cognitive abilities such as problem-solving, creativity, and learning, as well as the capacity to handle enormous volumes of data rapidly and correctly. Still, this is in the books and we are yet to reach that level of implementing those AIs.





# AI Programming focuses on Cognitive Skills

- **Learning:** This aspect of AI programming focuses on acquiring data and creating rules for how to turn it into actionable information. The rules, which are called algorithms, provide computing devices with step-by-step instructions for how to complete a specific task.
- **Reasoning:** This aspect of AI programming focuses on choosing the right algorithm to reach a desired outcome.
- **Self-correction:** This aspect of AI programming is designed to continually fine-tune algorithms and ensure they provide the most accurate results possible.
- **Creativity:** This aspect of AI uses neural networks, rules-based systems, statistical methods and other AI techniques to generate new images, new text, new music and new ideas.

# Real-World Applications for AI systems Today

- **Speech Recognition:** Also known as automatic speech recognition (ASR), computer speech recognition, or speech-to-text, speech recognition uses NLP to process human speech into a written format. Many mobile devices incorporate speech recognition into their systems to conduct voice search—Siri, for example—or provide more accessibility around texting in English or many widely-used languages.
- **Customer Service:** Online virtual agents and chatbots are replacing human agents along the customer journey. They answer frequently asked questions (FAQ) around topics, like shipping, or provide personalized advice, cross-selling products or suggesting sizes for users, changing the way we think about customer engagement across websites and social media platforms. Examples include messaging bots on e-commerce sites with virtual agents, messaging apps, such as Slack and Facebook Messenger, and tasks usually done by virtual assistants and voice assistants.
- **Computer Vision:** This AI technology enables computers and systems to derive meaningful information from digital images, videos and other visual inputs, and based on those inputs, it can take action. This ability to provide recommendations distinguishes it from image recognition tasks. Powered by convolutional neural networks, computer vision has applications within photo tagging in social media, radiology imaging in healthcare, and self-driving cars within the automotive industry.
- **Supply Chain:** Adaptive robotics act on Internet of Things (IoT) device information, and structured and unstructured data to make autonomous decisions. NLP tools can understand human speech and react to what they are being told. Predictive analytics are applied to demand responsiveness, inventory and network optimization, preventative maintenance and digital manufacturing.
- **Weather Forecasting:** The weather models broadcasters rely on to make accurate forecasts consist of complex algorithms run on supercomputers. Machine-learning techniques enhance these models by making them more applicable and precise.
- **Anomaly Detection:** AI models can comb through large amounts of data and discover atypical data points within a dataset. These anomalies can raise awareness around faulty equipment, human error, or breaches in security.



***What is ML?***

# Machine Learning

- **Machine Learning** is a branch of artificial intelligence that develops algorithms by learning the hidden patterns of the datasets used it to make predictions on new similar type data, without being explicitly programmed for each task.
- Traditional Machine Learning combines data with statistical tools to predict an output that can be used to make actionable insights.
- Machine learning is used in many different applications, from image and speech recognition to natural language processing, recommendation systems, fraud detection, portfolio optimization, automated task, and so on. Machine learning models are also used to power autonomous vehicles, drones, and robots, making them more intelligent and adaptable to changing environments.

# Types of Machine Learning

- **Supervised Machine Learning:** Supervised learning is a type of machine learning in which the algorithm is trained on the labeled dataset. It learns to map input features to targets based on labeled training data. In supervised learning, the algorithm is provided with input features and corresponding output labels, and it learns to generalize from this data to make predictions on new, unseen data.
- **Unsupervised Machine Learning:** Unsupervised learning is a type of machine learning where the algorithm learns to recognize patterns in data without being explicitly trained using labeled examples. The goal of unsupervised learning is to discover the underlying structure or distribution in the data.
- **Reinforcement Machine Learning:** Reinforcement learning is a type of machine learning where an agent learns to interact with an environment by performing actions and receiving rewards or penalties based on its actions. The goal of reinforcement learning is to learn a policy, which is a mapping from states to actions, that maximizes the expected cumulative reward over time.

# Difference between Machine Learning and Traditional Programming

Machine Learning	Traditional Programming	Artificial Intelligence
Machine Learning is a subset of artificial intelligence(AI) that focus on learning from data to develop an algorithm that can be used to make a prediction.	In traditional programming, rule-based code is written by the developers depending on the problem statements.	Artificial Intelligence involves making the machine as much capable, So that it can perform the tasks that typically require human intelligence.
Machine Learning uses a data-driven approach, It is typically trained on historical data and then used to make predictions on new data.	Traditional programming is typically rule-based and deterministic. It hasn't self-learning features like Machine Learning and AI.	AI can involve many different techniques, including Machine Learning and Deep Learning, as well as traditional rule-based programming.
ML can find patterns and insights in large datasets that might be difficult for humans to discover.	Traditional programming is totally dependent on the intelligence of developers. So, it has very limited capability.	Sometimes AI uses a combination of both Data and Pre-defined rules, which gives it a great edge in solving complex tasks with good accuracy which seem impossible to humans.
Machine Learning is the subset of AI. And Now it is used in various AI-based tasks like Chatbot Question answering, self-driven car., etc.	Traditional programming is often used to build applications and software systems that have specific functionality.	AI is a broad field that includes many different applications, including natural language processing, computer vision, and robotics.



# Various Applications of Machine Learning

- **Automation:** Machine learning, which works entirely autonomously in any field without the need for any human intervention. For example, robots perform the essential process steps in manufacturing plants.
- **Finance Industry:** Machine learning is growing in popularity in the finance industry. Banks are mainly using ML to find patterns inside the data but also to prevent fraud.
- **Government organization:** The government makes use of ML to manage public safety and utilities. Take the example of China with its massive face recognition. The government uses Artificial intelligence to prevent jaywalking.
- **Healthcare industry:** Healthcare was one of the first industries to use machine learning with image detection.
- **Marketing:** Broad use of AI is done in marketing thanks to abundant access to data. Before the age of mass data, researchers develop advanced mathematical tools like Bayesian analysis to estimate the value of a customer. With the boom of data, the marketing department relies on AI to optimize customer relationships and marketing campaigns.
- **Retail industry:** Machine learning is used in the retail industry to analyze customer behavior, predict demand, and manage inventory. It also helps retailers to personalize the shopping experience for each customer by recommending products based on their past purchases and preferences.
- **Transportation:** Machine learning is used in the transportation industry to optimize routes, reduce fuel consumption, and improve the overall efficiency of transportation systems. It also plays a role in autonomous vehicles, where ML algorithms are used to make decisions about navigation and safety.



# ***AI + ML role in Testing...***



# Why is AI worth exploring for companies in software testing?

- AI in software testing refers to using AI techniques and technologies to increase the testing process's efficiency, effectiveness, and accuracy.
- While traditional manual testing procedures involve more work and time, and are prone to human error, AI and ML techniques may automate testing processes such as test case generation, execution, and analysis.
- Still, it's important to remember that human-AI collaboration is essential.
- Artificial intelligence may streamline and improve testing processes, but human knowledge is crucial to designing test objectives, evaluating complex results, and making critical judgments. Software testing may achieve unprecedented efficiency and effectiveness by combining the strengths of AI and human testers, ultimately providing high-quality software products.
- Through the integration of AI and Machine Learning, testers, developers, and companies can gain valuable insights into the enhancement of testing operations.

# How AI and ML can be used to improve Software Testing

- **Automating mundane testing-related tasks:** AI and machine learning may be used to automate repetitive and time-consuming manual testing activities, such as checking the front end for defects and validating API capabilities. This allows testers to focus on more important tasks which require human intelligence.
- **Identifying potential coding defects:** AI and machine learning can be used to detect possible bugs in code, ones that are difficult to detect by human eyes. This can assist in enhancing software quality by detecting errors early in the development process.
- **Generating test cases:** AI and machine learning can be used to generate test cases based on the requirements of a software system. This can help to ensure that all aspects of the system are tested thoroughly.
- **Providing real-time feedback and recommendations:** AI and machine learning can be used to provide real-time feedback and recommendations to testers. This can help testers to improve their efficiency and effectiveness.

# Use cases where AI and ML could assist during our Software Testing Process


- **Automation of Test Case creation:** Testers can train ML models to recognize patterns in test cases and automatically generate scripts or identify potential areas for automation. This helps save time and effort, allowing testers to focus on more complex and critical testing activities.
- **Test Case Prioritization:** AI and ML algorithms may prioritize test cases based on their impact and likelihood of failure by analyzing historical data, user input, and application usage trends. This enables testers to properly manage their time and focus on the most crucial areas, guaranteeing full test coverage.
- **Defect Prediction:** AI and ML models can evaluate historical defect data, code complexity and vulnerabilities, and other relevant criteria to identify possible issues in current software releases. These predictions may be used by testers to proactively target certain areas for testing, minimizing the risk of defects reaching production.
- **Test Data Generation:** AI techniques, such as generative adversarial networks (GANs), can be used to generate synthetic data. Based on the trained data these techniques generate realistic and diverse test data. This aids testers in ensuring extensive test coverage and identifying potential edge scenarios that might otherwise go unnoticed.

# Use cases where AI and ML could assist during our Software Testing Process (Contd...)

- **Intelligent Test Execution:** AI-powered testing solutions may analyze test results in real-time and adapt test cases dynamically based on application behavior. This adaptive testing technique aids in the identification and resolution of issues, therefore enhancing the entire testing process.
- **Anomaly Detection:** AI and ML algorithms may discover anomalies or deviations from expected patterns by analyzing system logs, performance indicators, and user behavior. This information may be used by testers to uncover any bugs or vulnerabilities that may affect the software's performance or security.
- **Continuous Testing:** AI and machine learning techniques may be integrated into continuous testing pipelines, allowing for faster feedback loops and error discovery. This helps to guarantee that software releases are of higher quality and adhere to DevOps and Agile concepts.
- **Exploratory Testing:** Machine learning techniques may be used to build exploratory tests, which can identify unexpected behavior and edge situations that testers may not have considered.
- **Test Results Analysis:** AI techniques can be utilized to help in the analysis of test results to uncover plausible cause-and-effect links between the tests and the software under test or to identify patterns in the data that could suggest a regression.

# Benefits of AI and ML for Software Testers

- **Improved Accuracy:** AI and Machine Learning have significance in testing because they can rapidly and reliably analyze enormous volumes of data. AI systems can see patterns in data that humans might overlook, and they are adept at processing big datasets.
- **Increased Efficiency:** With AI and machine learning technology, the software quality process may be sped by lowering the amount of testing manually necessary. Automation of tests can minimize testing time and assure a more accurate output.
- **Reduced Costs:** Automating the software testing process can help to reduce the overall cost of the software development lifecycle.
- **Improved Risk Management:** AI and machine learning may be used to discover potential risks in software by offering insights into complex patterns.
- **Increased Data Analysis:** Organizations may use AI and machine learning to predict defect rates and identify new trends using massive datasets.



# ***Revision***

## ***[Recap of Lessons...]***

# What is Test Automation?

- Software testing technique to test and validate actual test result with expected test result with minimal or no touch.
- Automation Testing is use of Tools or Software to perform software testing.
- Developing and executing tests that can run and compare actual to expected results.
- The automation software can also enter test data into the system under test, compare expected and actual results and generate detailed test reports.

# Need for Automation Testing

- **Speed** : Automation scripts are faster when compared to manual testers effort.
- **Reliable** : Tests perform precisely the same operation each time they are run, there by eliminating human error.
- **Repeatable** : Tests can be repeated n number of times for execution of the same operation.
- **Coverage** : Automated tests increase the coverage.
- **Reusable** : We can reuse tests on different versions of an application, even if the user interface changes.



# When/What Tests to Automate

- **Regression Testing** : When the software application is fairly stable and only regression tests needs to be executed.
- **Smoke Testing** : For getting a high-level assessment of the quality of the build, and making quick go / no-go on further testing.
- **Static & Repetitive Tests** : For automating testing tasks that are repetitive and relatively unchanging from one test cycle to the next.
- **Data Driven Testing** : For testing application functions where the same function needs to be validated with lot of different inputs & large data sets (Ex: Login, & Search).
- **Load & Performance Testing** : No viable manual alternative exists.

# Benefits of Test Automation

- Faster Feedback
- Accelerated results
- Reduced business expenses
- Testing efficiency Improvements
- Reusability of Automated Tests
- Earlier detection of defects
- Thoroughness in Testing
- Faster time to market



# AUTOMATION TESTING

## Automation Testing Life Cycle

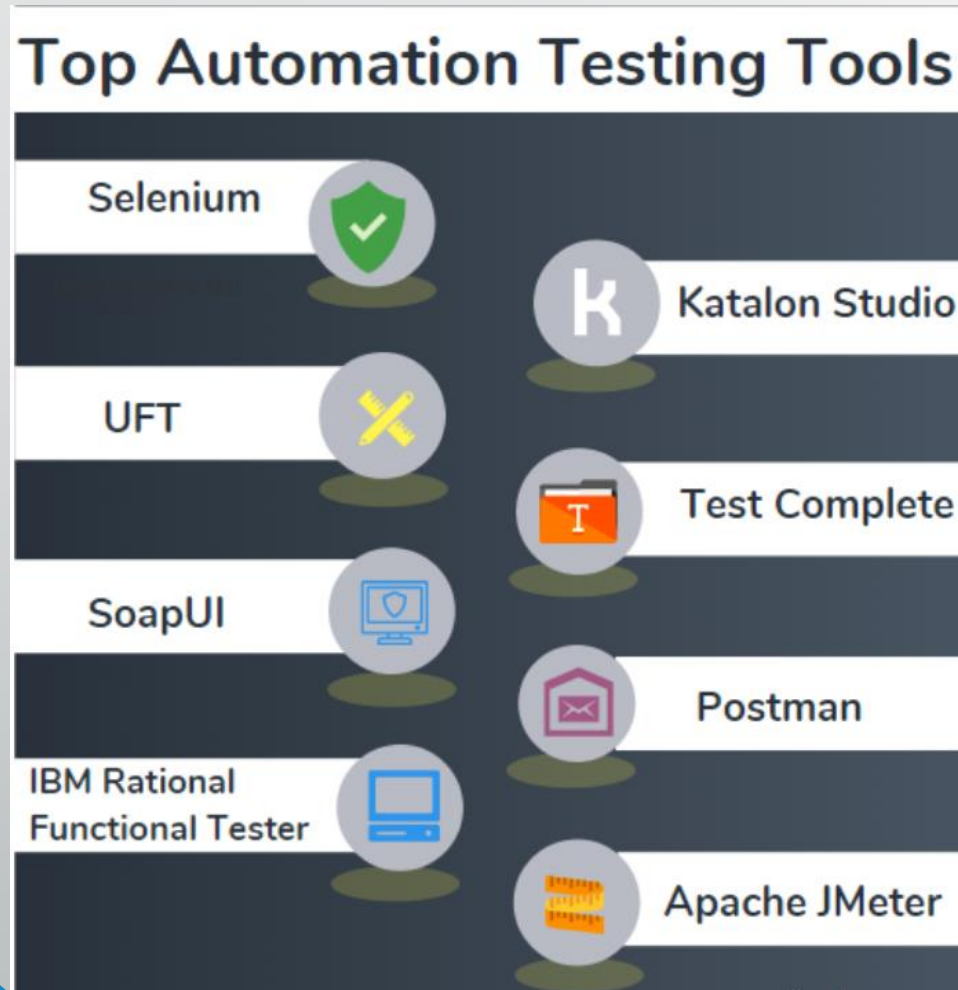


# Test Strategy

- **Test strategy** is the document that describes the testing approach of the software product. The test strategy is created to inform project managers, developers, and testers about key issues of the testing process.
- Generally, a test strategy document contains below sections:








# Automation Testing Tools - I

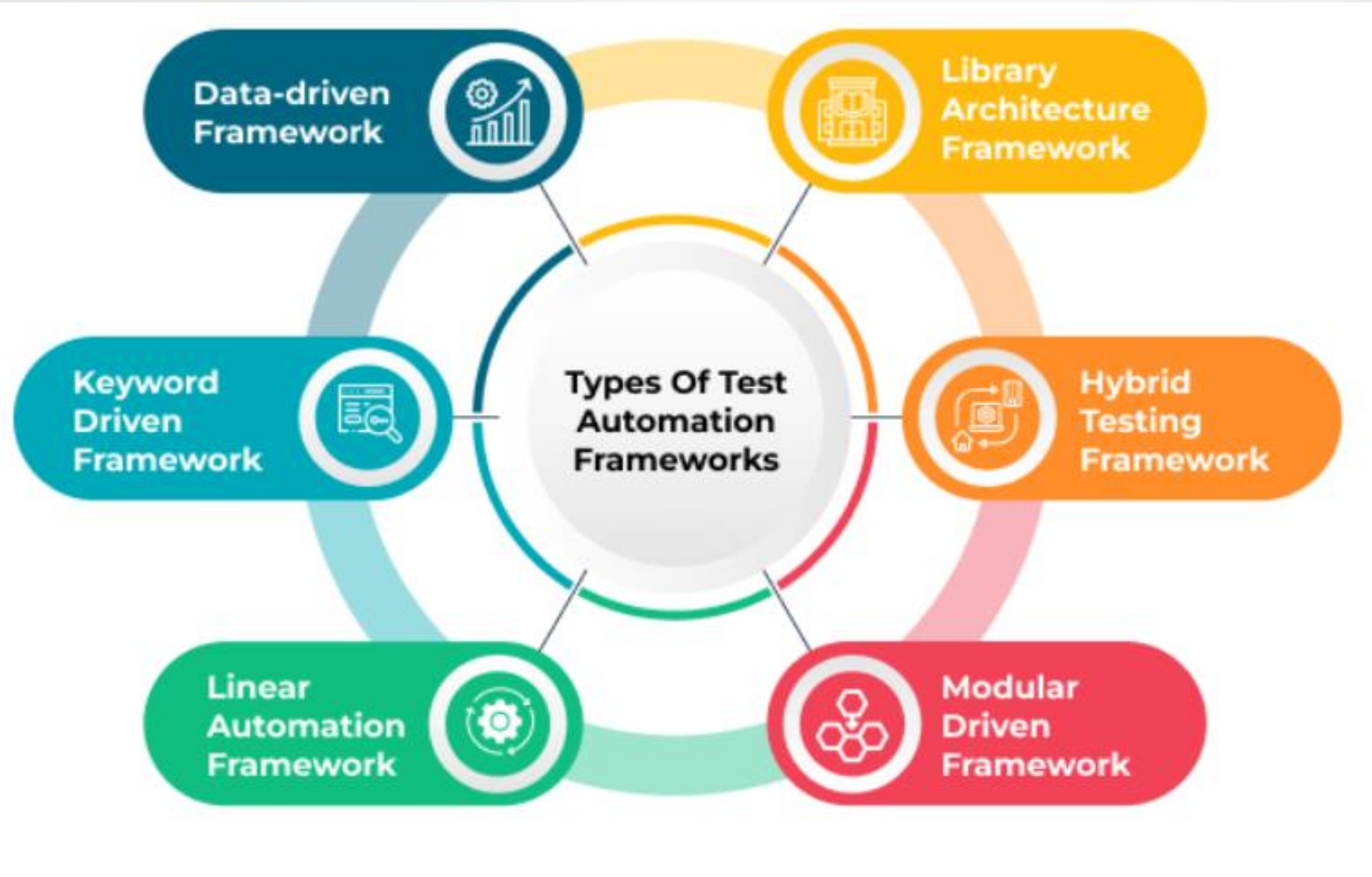




# Automation Testing Tools – Comparison

Product	 Katalon	 Selenium	 appium	 TestComplete	 cypress
Application Under Test	Web/API/ Mobile/Desktop	Web	Mobile (Android/iOS)	Web/Mobile/ Desktop	Web
Supported platform(s)	Windows/ macOS/ Linux	Windows/ macOS/ Linux/Solaris	Windows/ macOS	Windows	Windows/ macOS/ Linux
Setup & configuration	Easy	Coding Required	Coding Required	Easy	Coding Required
Low-code & Scripting mode	Both	Scripting Only	Scripting Only	Both	Scripting Only
Supported language(s)	Java & Groovy	Java, C#, Python, JavaScript, Ruby, PHP, Perl	Java, C#, Python, JavaScript, Ruby, PHP, Perl	JavaScript, Python, VBScript, JScript, Delphi, C++, C#	JavaScript
Advanced test reporting	✓	✗	✗	✗	✓
Pricing	Free and Paid	Free	Free	Paid	Free and Paid
Ratings & Reviews (Gartner)	4.4/5 740 reviews	4.5/5 443 reviews	4.4/5 90 reviews	4.4/5 45 reviews	4.6/5 27 reviews

# Test Automation Frameworks



# Barriers to Test Automation, why some fail - I?

- **Resources & Priorities**

- Initial Investment (Cost, Time, & Effort)
- Competing corporate initiatives and priorities
- Lack of clear mandate and realistic goals
- Not treated like other software dev projects

- **Tools & Environment**

- Not using proper tools and framework
- Legacy or constantly changing code
- Not having controlled or stable test environment



# Barriers to Test Automation, why some fail - II?

- **Culture & Skillset**

- Teams attitude approach and resistance to change
- Lack of experience and false sense of security
- Relies on programming language only
- Underestimate the amount of time needed
- Creating large and end-to-end tests

- **Process**

- Not reusing automation code
- Not having a test data strategy in-place
- Not making you automated tests readable

# Test Cases You Shouldn't Automate

- Tests that will be executed once
- Tests based on visual perception
- Tests without evident pass / fail results
- Anti Automation features, like CAPTCHA
- Raw and unstable functionality
- Features with changing requirements
- Newly designed test cases

# Limitation of Automation Testing

- **Need for scripting and programming skills** – Coding and technical skill level of the resource should be good enough to write robust testing code.
- **Need for maintenance of code** – Whenever application code is updated or modified, the code for automated test case must also be updated.
- **Requires more initial developer time** – Any new test automation will require time for development, creation of frameworks etc...
- **Increase tool needs** – Automation testing would increase the need for tools (either licensed or Opensource) and also software required for the same.

# Challenges of Automation Testing

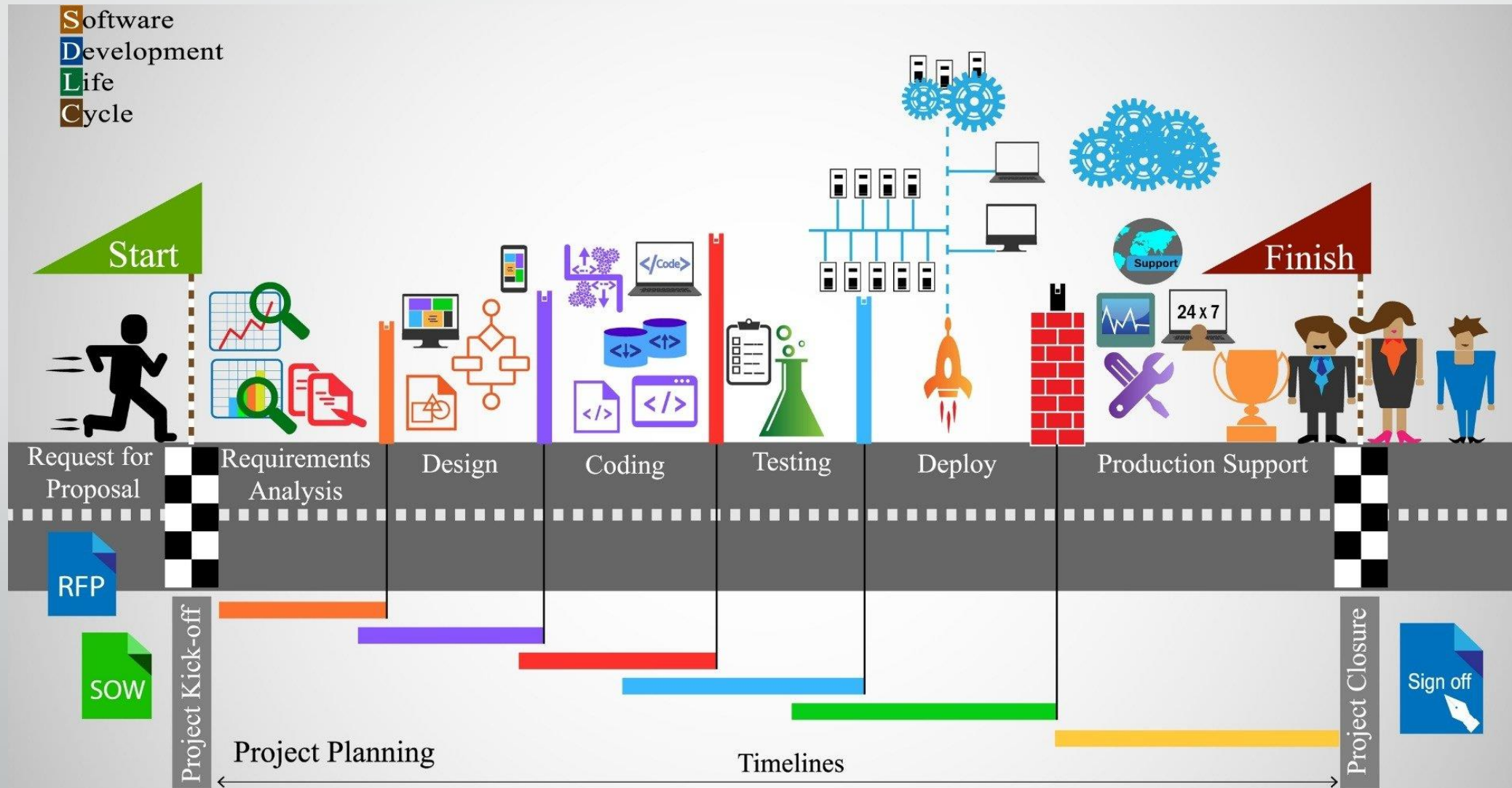
- **Unrealistic Expectations** – Generally there is a tendency to be optimistic/have high expectation about what can be achieved by a new test tool.
- **Tool Limitations** – Tools available in the market have one or other limitation, which needs to be addressed.
- **Dependency on 3<sup>rd</sup> party integration** – Integrations with other applications, plugins, patches, etc... makes automation challenging.
- **Lack of Help & Support required for the tool** – Some tool have extensive support required for the tool, others have to rely on internet and other user forums.
- **Version compatibility for tool and browser** – The browsers are updated very rapidly in the market, but the tool supportability for the version make take considerable time.

# Software Development Methodologies

Software development methodology is defined as a framework for developing information systems, focusing on planning and organization. It benefits both teams and customers by improving efficiency and adaptability to changes.

- Waterfall
- Agile
- DevOps
- Rational Unified Process
- Rapid Application Development (RAD)
- Feature-driven Development (FDD)
- Extreme Programming (XP)

# Waterfall Methodology



# Agile Methodology

- ***Agile methodology*** is a project management approach that prioritizes cross-functional collaboration and continuous improvement. It divides projects into smaller phases and guides teams through cycles of planning, execution, and evaluation.
- **Agile's Four Main Values are:**
  - Individuals and interactions over processes and tools
  - Working software over comprehensive documentation
  - Customer collaboration over contract negotiation
  - Responding to change over following a plan

# Key Terms related Agile Methodology

- Product Owner
- Scrum Master
- Backlog
- CI/CD Pipelines
- Definition of Done (DoD)
- Retrospective
- Scrum / Kanban
- Sprint
- Story Points
- Themes - Epic - User Story
- Burndown Chart
- Daily Stand-up Meetings





# Benefits of using Agile Methodology

*Agile is one of the most popular approaches to project management because it is flexible, it is adaptable to changes and it encourages customer feedback.*

*Many teams embrace the Agile approach for the following reasons:*

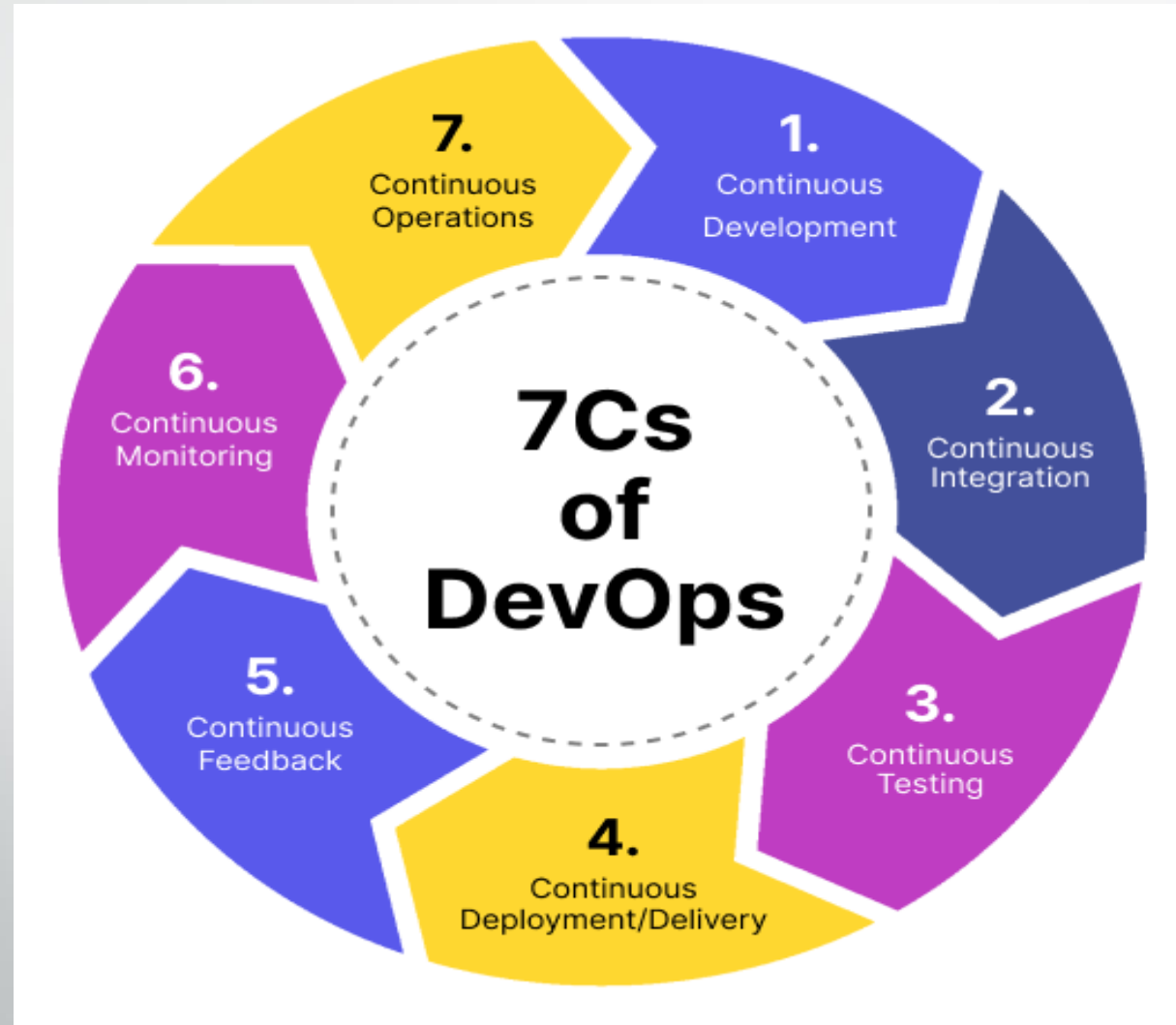
- **Rapid Progress:** By effectively reducing the time it takes to complete various stages of a project, teams can elicit feedback in real time and produce working prototypes or demos throughout the process
- **Customer and Stakeholder Alignment:** Through focusing on customer concerns and stakeholder feedback, the Agile team is well positioned to produce results that satisfy the right people
- **Continuous Improvement:** As an iterative approach, Agile project management allows teams to chip away at tasks until they reach the best end result

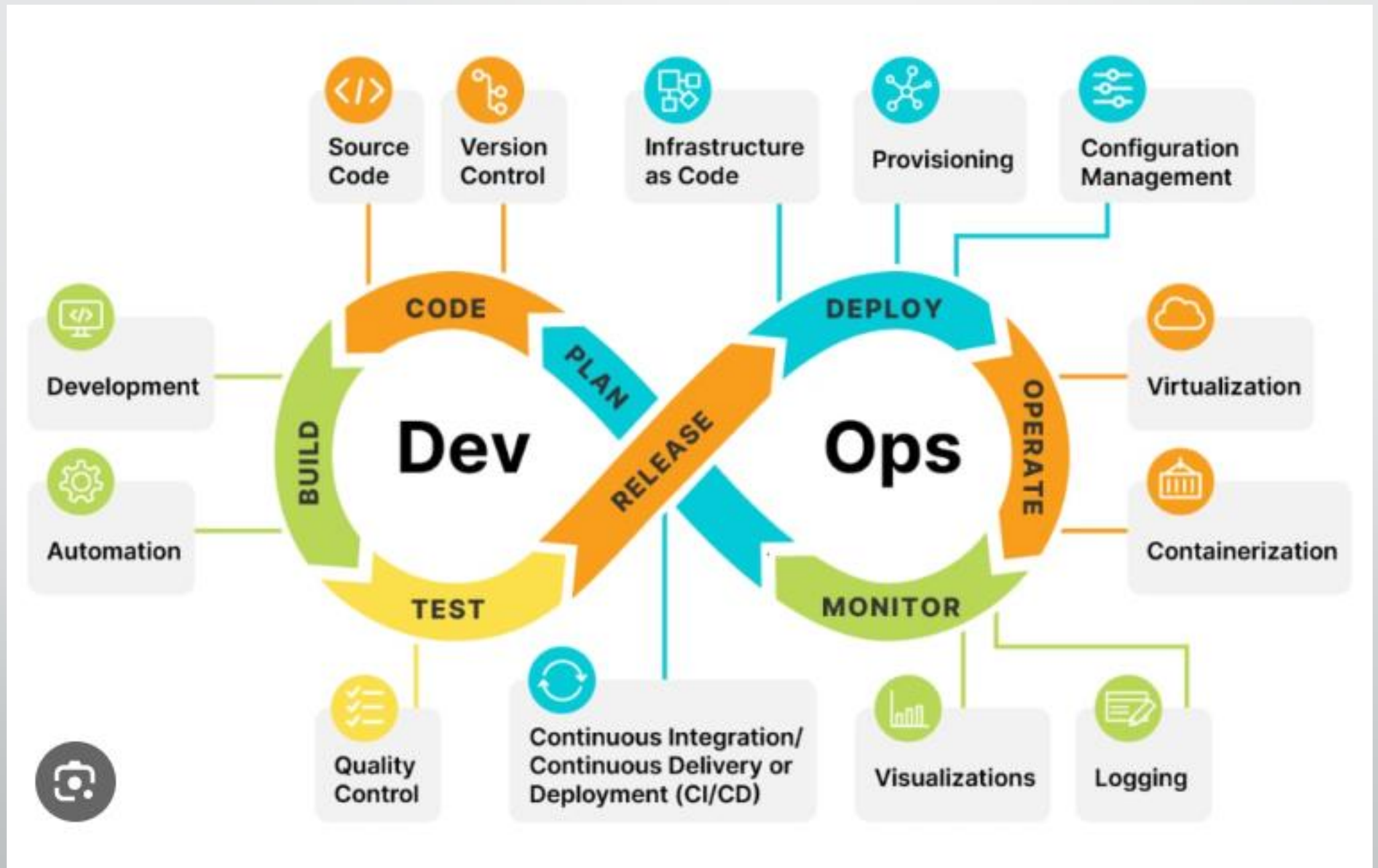
# DevOps

- DevOps is a combination of *software development* (dev) and *operations* (ops)
- A collaborative approach combining *development* and *Operations* for software delivery.

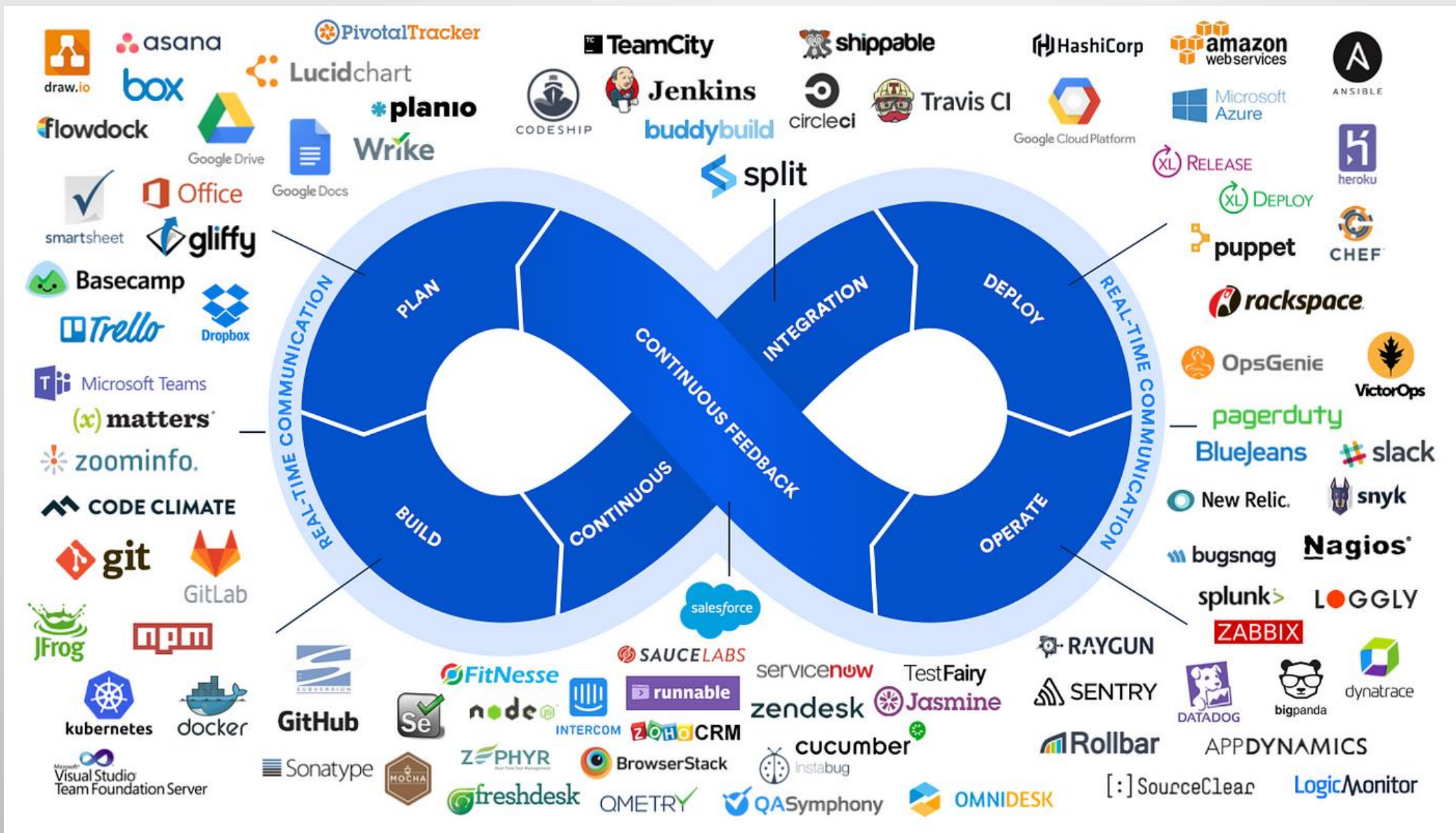


# DevOps Lifecycle









# DevSecOps

*Security has become an integral part of the software development lifecycle.*

- DevSecOps ensures that DevOps teams understand the security and compliance requirements from the very beginning of application creation and can properly protect the integrity of the software.
- By integrating security seamlessly into DevOps workflows, organizations gain the visibility and control necessary to meet complex security demands, including vulnerability reporting and auditing.
- Security teams can ensure that policies are being enforced throughout development and deployment, including critical testing phases.
- DevSecOps can be implemented across an array of environments such as on-premises, cloud-native, and hybrid, ensuring maximum control over the entire software development lifecycle.

# DevOps Vs SRE [Site Reliability Engineering ]

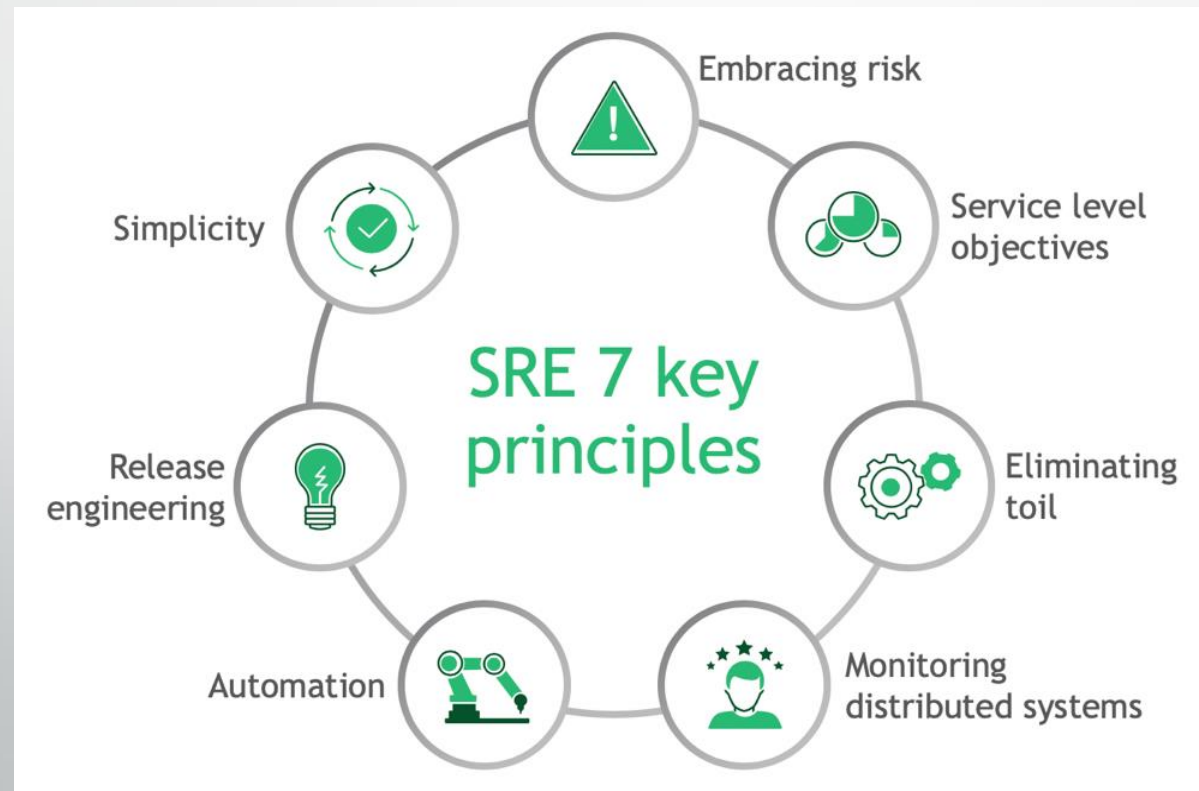
*SRE is the practical implementation of DevOps. DevOps provides the philosophical foundation of what must be done to maintain software quality amidst the increasingly shortened development timeline. Site reliability engineering offers the answers to how to achieve DevOps success. SRE ensures that the DevOps team strikes the right balance between speed and stability.*





# SRE - Site Reliability Engineering

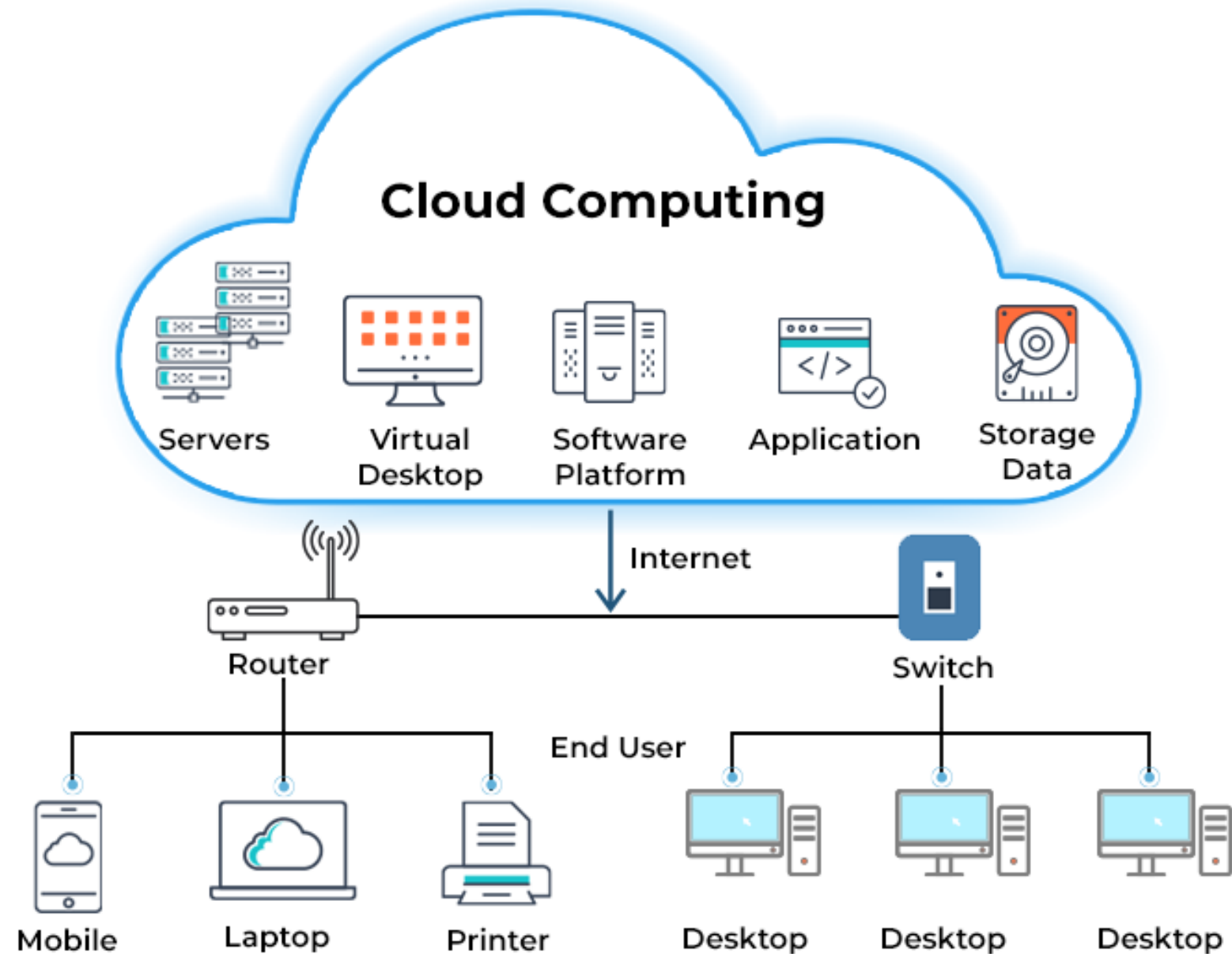
*Site Reliability Engineering (SRE) is the practice of using software tools to automate IT infrastructure tasks such as system management and application monitoring. Organizations use SRE to ensure their software applications remain reliable amidst frequent updates from development teams.*



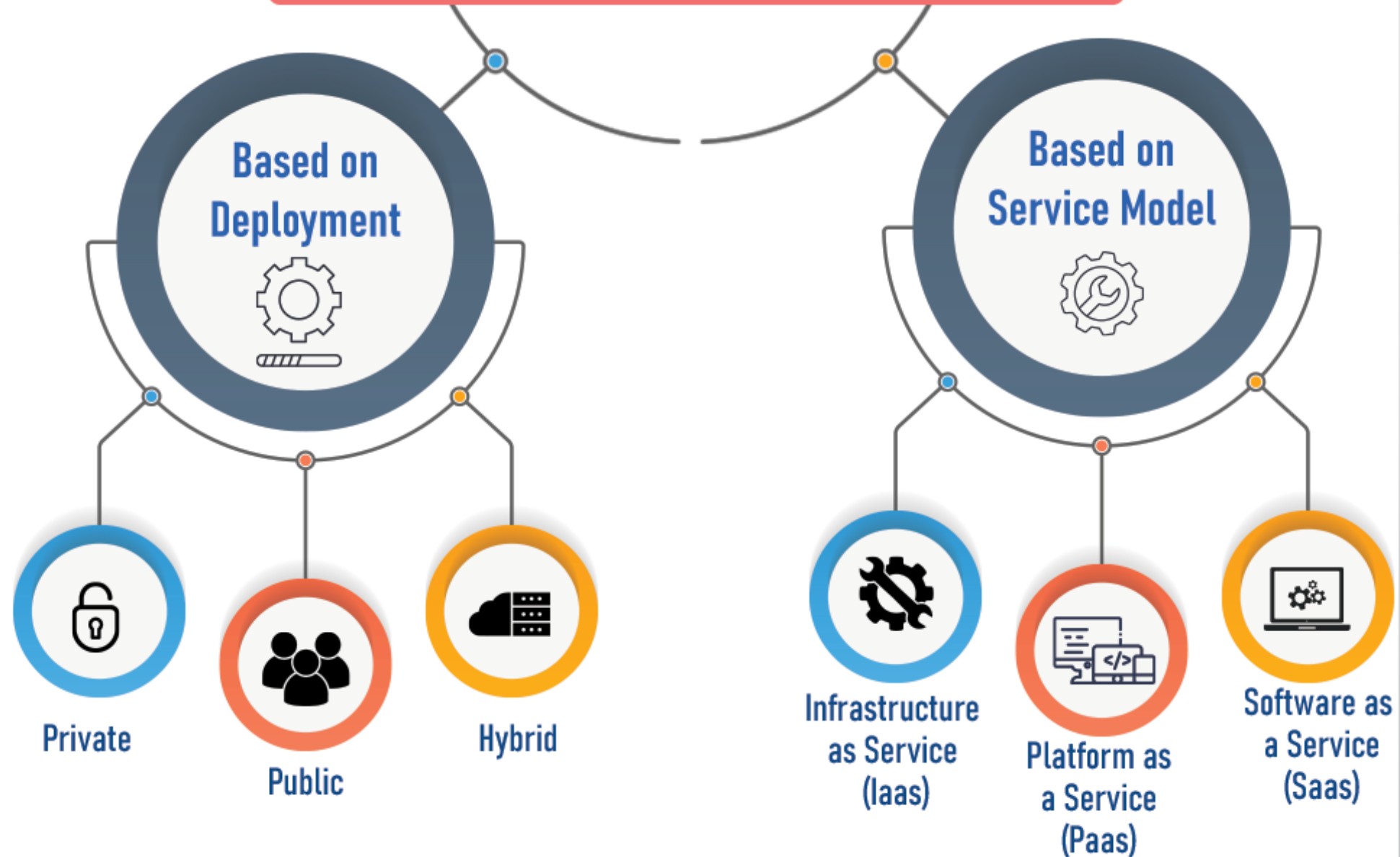
# Cloud Computing

- **Cloud Computing** is the delivery of computing services - including *servers, storage, databases, networking, software, analytics, and intelligence* - over the Internet (“the cloud”) to offer faster *innovation, flexible resources, and economies of scale*.
- **There are four main types of Cloud Computing:** Private Clouds, Public Clouds, Hybrid Clouds, and Multiclouds.
- **There are also three main types of Cloud Computing Services:** Infrastructure-as-a-Service (IaaS), Platforms-as-a-Service (PaaS), and Software-as-a-Service (SaaS).

# CLOUD COMPUTING ARCHITECTURE



# TYPES OF CLOUD COMPUTING

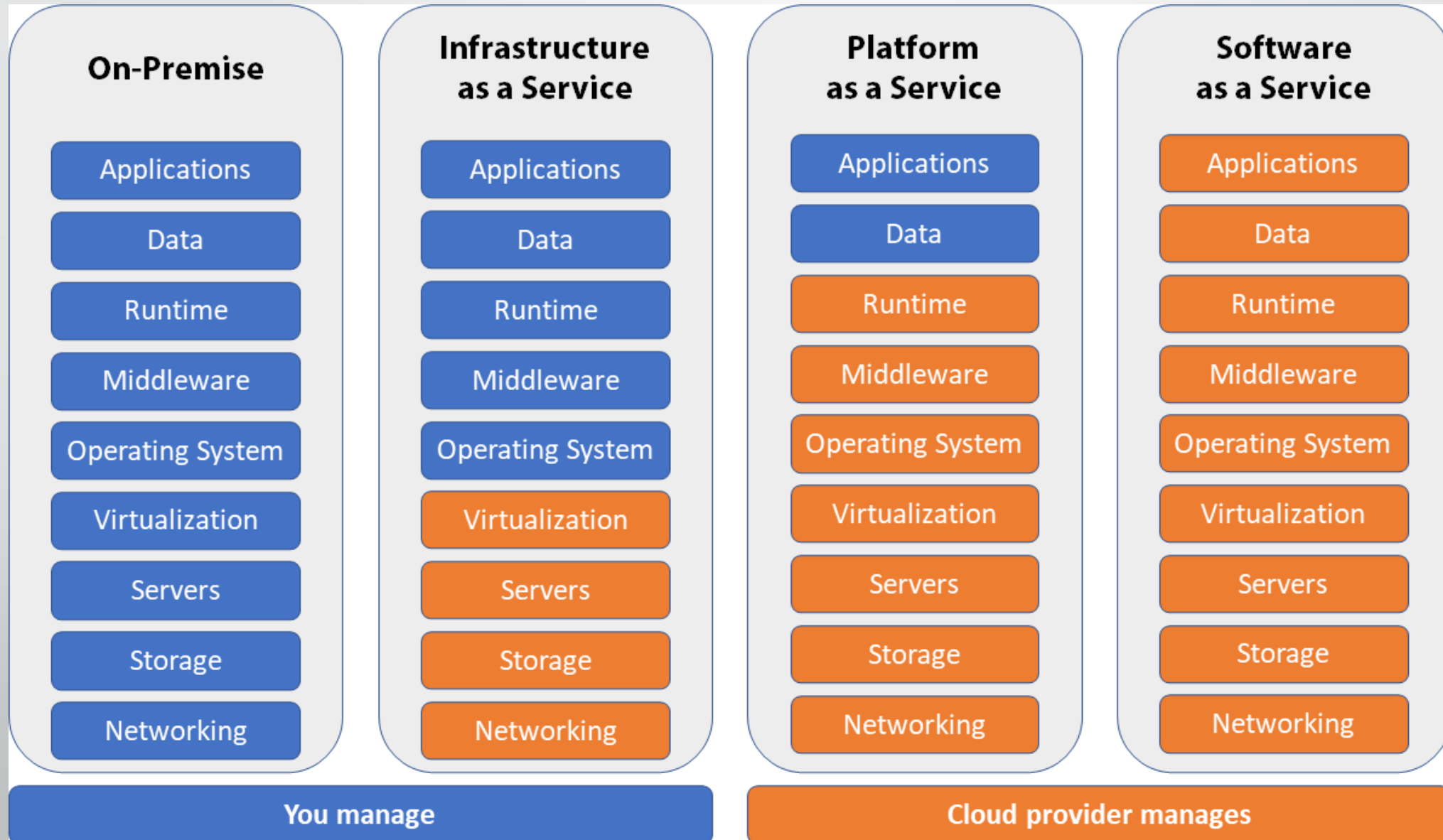


# Public Cloud Providers

- Top 7 Public Cloud Providers...



# On-Premise Vs IaaS Vs PaaS Vs SaaS Management



# SaaS Tools

- Most companies nowadays use a huge range of SaaS tools. From project management to web hosting to creative suites, there is a SaaS option for just about anything.*





# Cloud Testing

- **Cloud testing** is a process of using cloud based resources to stimulate real world user traffic and environments for testing software applications. This method leverages cloud computing environments to provide more diverse testing scenarios.
- **Cloud software testing is defined** by its ability to test programs in a scalable environment that simulates real-world situations without needing costly hardware or software infrastructures.
- **Organizations can employ cloud-based testing services to verify** that their applications are resilient, reliable, and ready to meet the demands of their consumers.

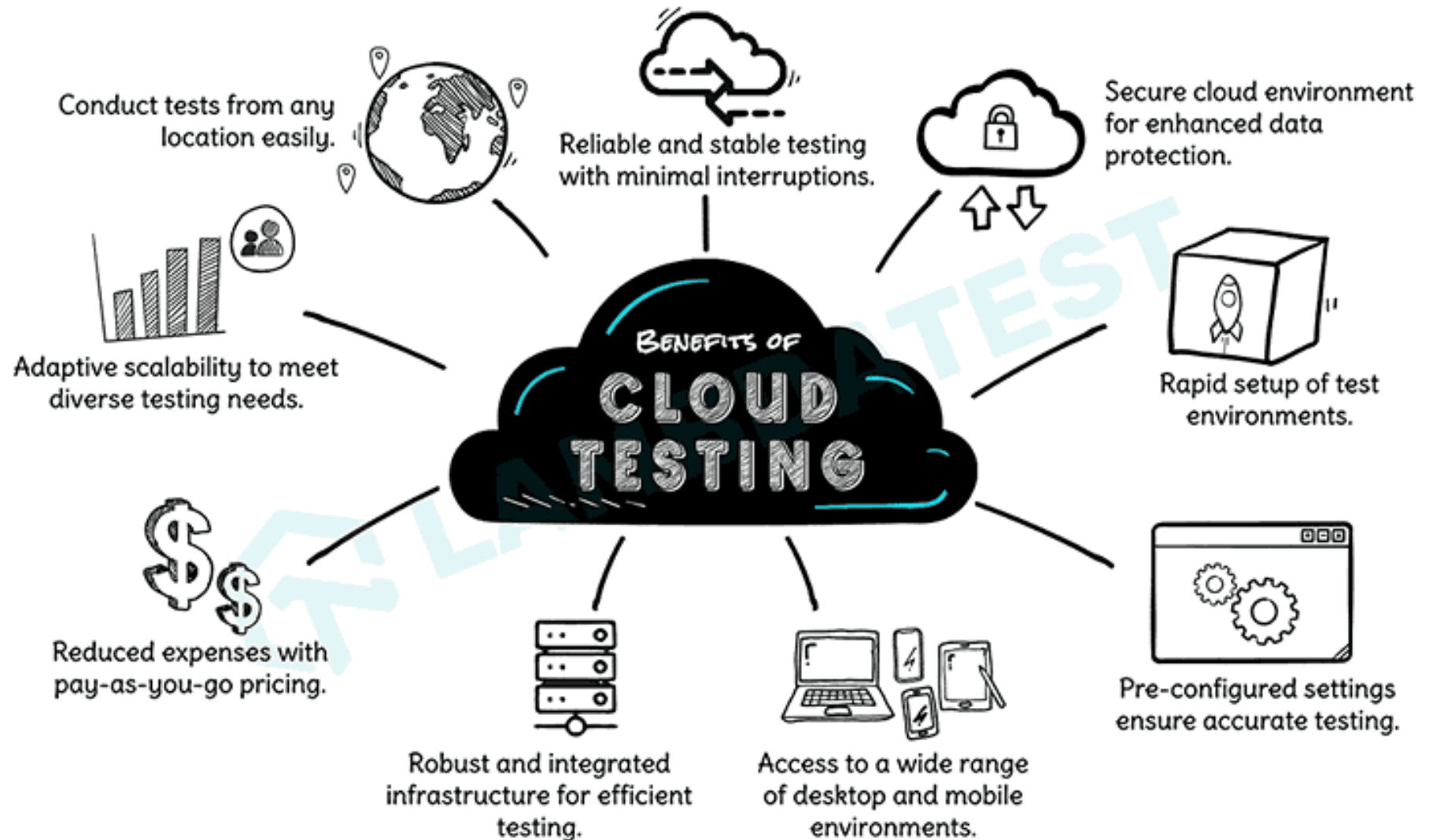


# Types of Cloud Testing

While cloud testing in broad terms refers to testing applications through cloud computing resources, there are three main types of cloud testing that vary by purpose:

- **Testing of Cloud Resources.** The cloud's architecture and other resources are assessed for performance and proper functioning. This involves testing a provider's platform as a service (PaaS) or infrastructure as a service (IaaS). Common tests may assess scalability, disaster recovery (DR), and data privacy and security.
- **Testing of Cloud-native Software.** Testing of SaaS products that reside in the cloud.
- **Testing of Software with Cloud-based Tools.** Using cloud-based tools and resources for QA testing.

# Benefits of Cloud Testing



# Security Testing

- **Security Testing** is a type of Software Testing that uncovers vulnerabilities in the system and determines that the data and resources of the system are protected from possible intruders.
- It ensures that the software system and application are free from any threats or risks that can cause a loss.
- Security testing of any system is focused on finding all possible loopholes and weaknesses of the system that might result in the loss of information or reputation of the organization.

# Principles of Security Testing

- **Confidentiality**
  - Confidentiality is one of the cornerstones of information security. Confidentiality is the obligation of an organization or individual to keep the information confidential. Confidential information is any information that is not meant to be shared with third parties. The primary purpose of confidentiality is to protect the stakeholders' interests by preventing the unauthorized disclosure of information.
- **Integrity**
  - Integrity is one of the core security concepts. It is about system and data integrity. The need for integrity stems from the fact that we often want to ensure that a file or data record has not been modified or has not been modified by an unauthorized party. Integrity is a fundamental security concept and is often confused with the related concepts of confidentiality and non-repudiation.
- **Availability**
  - The definition of availability in information security is relatively straightforward. It's the ability to access your information when you need it. A data breach might cause downtime, productivity, loss of reputation, fines, regulatory action, and many other problems. For all of these reasons, it's crucial to have a data availability plan in case a data breach happens.

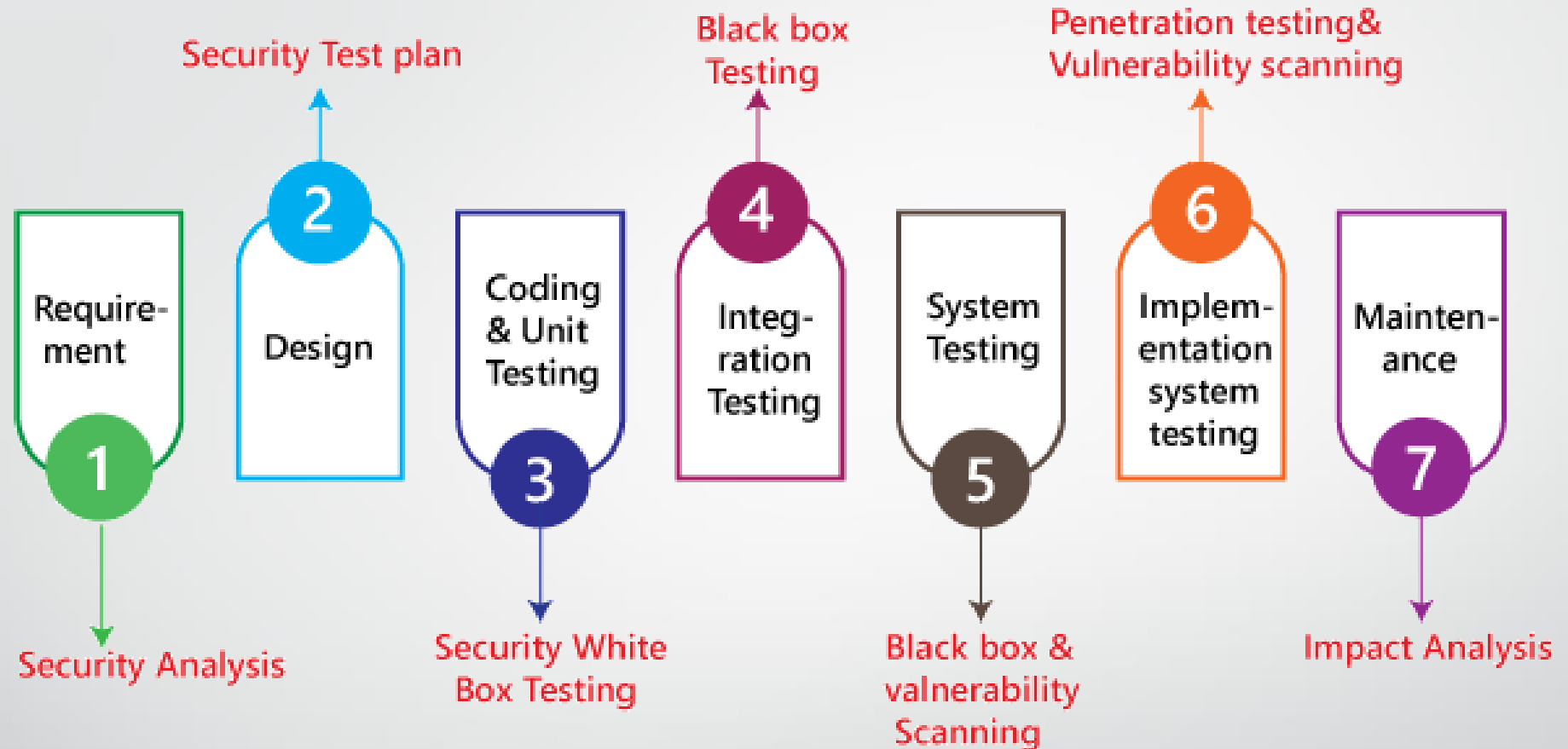
# Key Areas in Security Testing

- **Network Security:** Data transfer between UI and API is ensured to be encrypted and secured.
- **System Software Security:** The entire application, UI, server, and database must be secured.
- **Client-side Application Security:** The data sent by the user to server under HTTPS must be secured.
- **Server-side Application Security:** The interaction between the database and receiving data from the database and the client has to be secured.

# Areas in Security Testing (Contd...)

- **Authentication and Authorization:** Testing the system's ability to properly authenticate and authorize users and devices. This includes testing the strength and effectiveness of passwords, usernames, and other forms of authentication, as well as testing the system's access controls and permission mechanisms.
- **Network and Infrastructure Security:** Testing the security of the system's network and infrastructure, including firewalls, routers, and other network devices. This includes testing the system's ability to defend against common network attacks such as denial of service (DoS) and man-in-the-middle (MitM) attacks.
- **Database Security:** Testing the security of the system's databases, including testing for SQL injection, cross-site scripting, and other types of attacks.
- **Application Security:** Testing the security of the system's applications, including testing for cross-site scripting, injection attacks, and other types of vulnerabilities.
- **Data Security:** Testing the security of the system's data, including testing for data encryption, data integrity, and data leakage.
- **Compliance:** Testing the system's compliance with relevant security standards and regulations, such as HIPAA, PCI DSS, and SOC2.
- **Cloud Security:** Testing the security of cloud.

# Security Testing along with SDLC



- **Test Plan should include,**
  - Security-related test cases or scenarios
  - Test Data related to security testing
  - Test Tools required for security testing
  - Analysis of various tests outputs from different security tools

# Example Test Scenarios for Security Testing

- Password stored must be encrypted in the database.
- Application or System should not allow invalid users.
- Penetration testing like SQL injection must be tested.
- Data transfer from UI to API must also be encrypted.
- Sessions and Cookies must be tested for security.
- For financial sites, the Browser back button should not work.



## Myths and Facts of Security Testing

Myths	Facts
Security testing is needed where payment gateway integration is there.	Security testing is needed for system and network security
Security testing should be done in the end	Security testing is needed at every stage of product development
After Security testing application is 100% secure	There is never 100% security. The product is improved daily.
Security testing is needed only in large scaled products	Security Testing is needed from small to big products

# Key Roles Associated with Security Testing

- **Hackers** – Access computer system or network without authorization...
- **Crackers** – Break into the systems to steal or destroy data...
- **Ethical Hacker** – Performs most of the breaking activities but with permission from the owner...
- **Script Kiddies or Packet Monkeys** – Inexperienced Hackers with programming language skill...

# Cyber Security

- **Cyber security** is the practice of protecting networks, applications, confidential or sensitive data, and users from cyber attacks.
- Cyber attacks are malicious attempts by individuals or groups to gain unauthorized access to computer systems, networks, and devices in order to steal information, disrupt operations, or launch larger attacks.
- It's also known as **Information Technology Security** or **Electronic Information Security**.

# Types of Cyber Threats

- The threats countered by cyber-security are three-fold:
  1. **Cybercrime** includes single actors or groups targeting systems for financial gain or to cause disruption.
  2. **Cyber-attack** often involves politically motivated information gathering.
  3. **Cyberterrorism** is intended to undermine electronic systems to cause panic or fear.

# What is the impact of a Cyberattack?

- **The impact of a cyberattack can be far-reaching and devastating for businesses.** One of the most significant impacts is economic costs, as cyberattacks can result in the loss of revenue, increased expenses for remediation and recovery, and supply chain disruption.
- **Cyber attacks can also impact brand reputation.** When organizations suffer a data breach or a temporary outage, their brand image may be affected — resulting in poor media coverage and the potential loss of current and future customers to competitors.
- **Additionally,** cyberattacks can result in regulatory costs, as companies may face fines for failing to protect user data in accordance with data protection laws such as the GDPR or HIPAA.



***Thank You !!!***



***All the best for your Exams!!!***



***"Education is not the learning of facts but  
the training of the mind to think..."***

***- Albert Einstein***

***"Before anything else preparation is key to  
success..."***

***- Alexander Graham Bell***