

Software Engineering Process

SE3010

Software Engineering Process and Quality Management

Agenda

- What we covered Last Week...
- Why Software Engineering Process is required?
- Software Development Methodologies,
- Waterfall Methodology
- Agile Methodology
- DevOps Methodology
- DevOps Vs SRE
- Site Reliability Engineering
- Discussion
- What was covered so far
- Recap
- Next Week...

What we covered Last Week...

- **Test Automation**
 - What is Test Automation?
 - Need for Automation Testing
 - Classifications of Test Automation
 - When/What Tests to Automate
 - Benefits of Test Automation
 - Automation Testing Life Cycle (ATLM)
 - Automation Testing Tools
 - Test Automation Frameworks
 - Barriers to Test Automation, why some fail?
 - Test Cases You Shouldn't Automate
 - Limitation & Challenges of Automation Testing
 - Test Automation will not...



Why Software Engineering Process is required?

Software Development Methodologies

infromation systems develop krnna thyna framework ekak thamai software development kiyanna

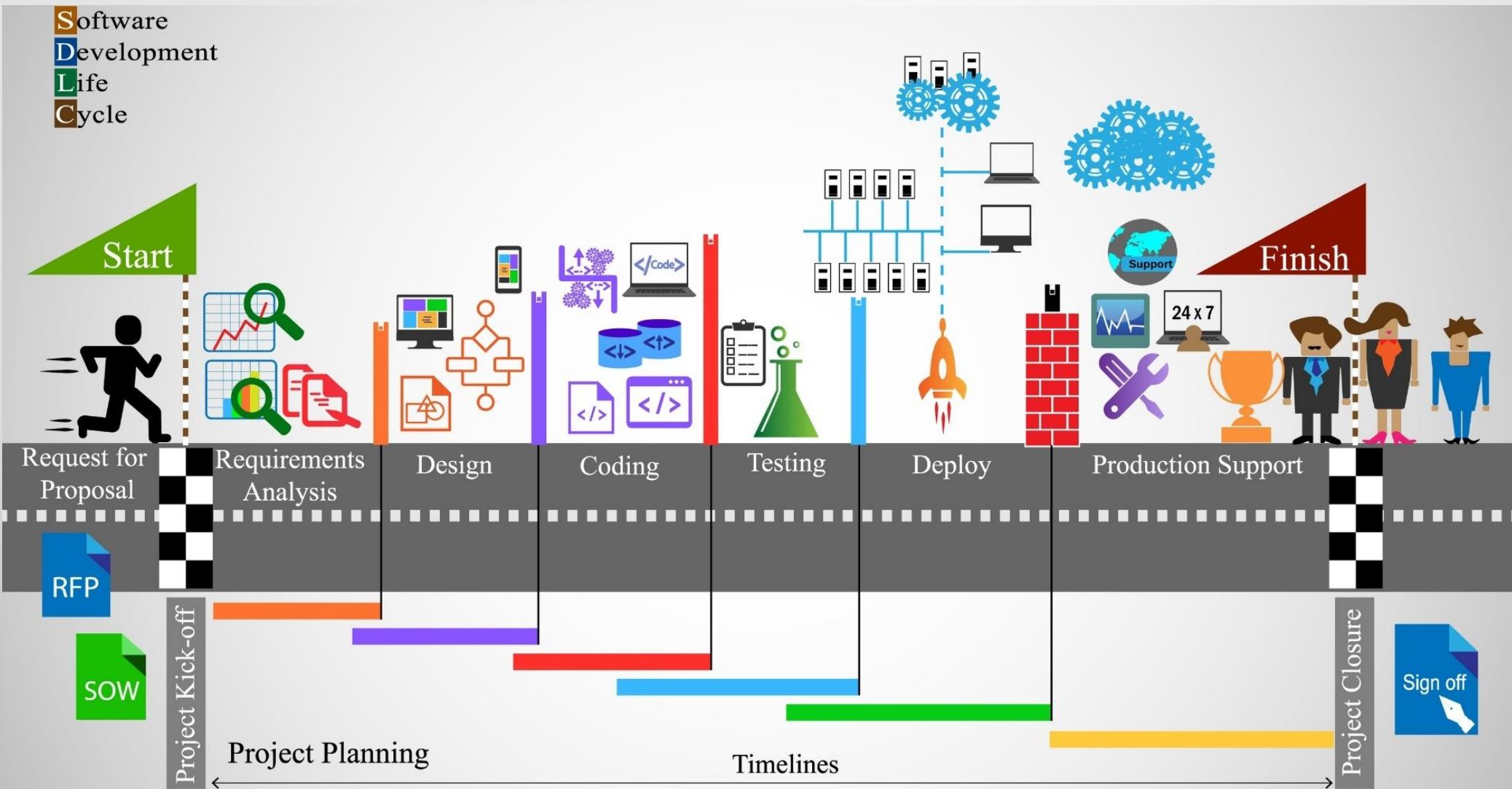
Software development methodology is defined as a framework for developing information systems, focusing on planning and organization. It benefits both teams and customers by improving efficiency and adaptability to changes.

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



What is Waterfall Methodology?

Waterfall Methodology





What is Agile 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.

project eka part part walata kadala part ekn part eka plan krala execute krla evalute krnwa

- **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

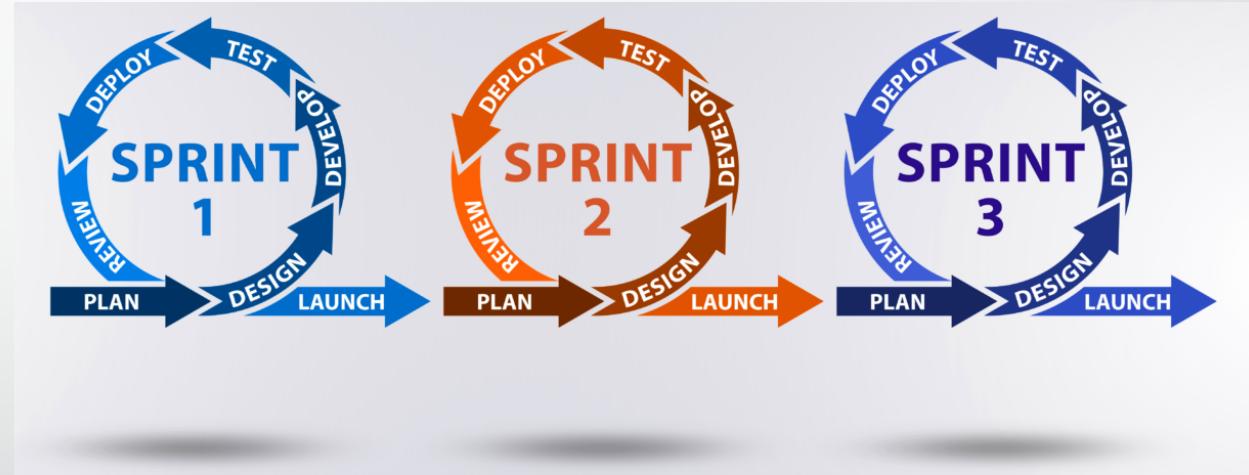
Types of Agile methodologies

*Agile project management is not a singular framework but an **umbrella term** that includes a wide range of methodologies, including Scrum, Kanban, Extreme Programming (XP), and the Adaptive Project Framework (APF).*

- **Scrum:** It is ideal for projects with rapidly changing requirements, using short sprints.
- **Kanban:** It visualizes project progress and is great for tasks requiring steady output.
- **Lean:** It streamlines processes, eliminating waste for customer value.
- **Extreme Programming (XP):** It enhances software quality and responsiveness to customer satisfaction.
- **Adaptive Project Framework (APF):** Works well for projects with unclear details, as it adapts to constantly evolving client needs.

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:

fast progress
customer focus
always getting better

- **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



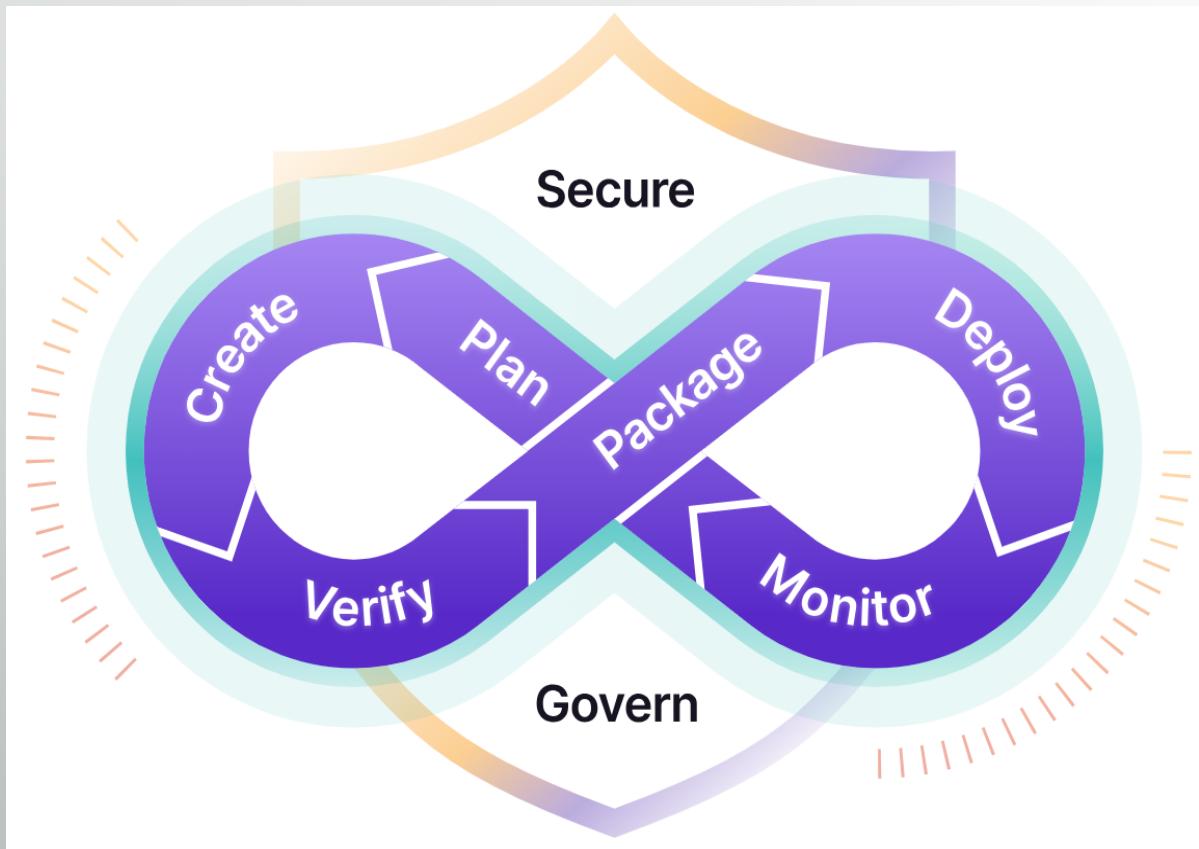
What is DevOps Methodology?

DevOps

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



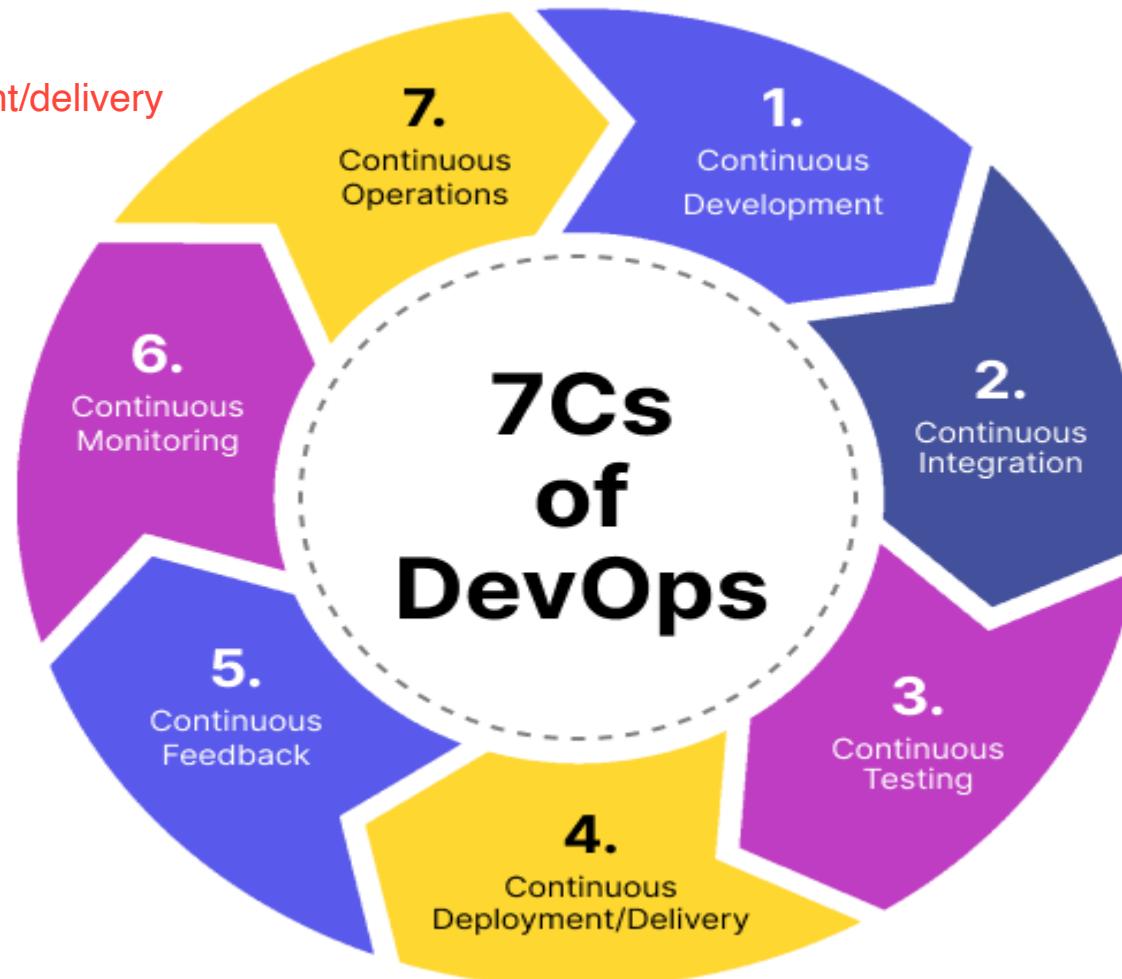
DevOps



- DevOps represents a change in mindset for IT culture.
- In building on top of Agile, lean practices, and systems theory, DevOps focuses on incremental development and rapid delivery of software.
- Success relies on the ability to create a culture of accountability, improved collaboration, empathy, and joint responsibility for business outcomes.

DevOps Lifecycle

- 1. continuous development
- 2. Continuous integration
- 3. Continuous testing
- 4. Continuous Deployment/delivery
- 5. Continuous Feedback
- 6. Continuous Monitoring
- 7. Continuous operations



Core DevOps Principles

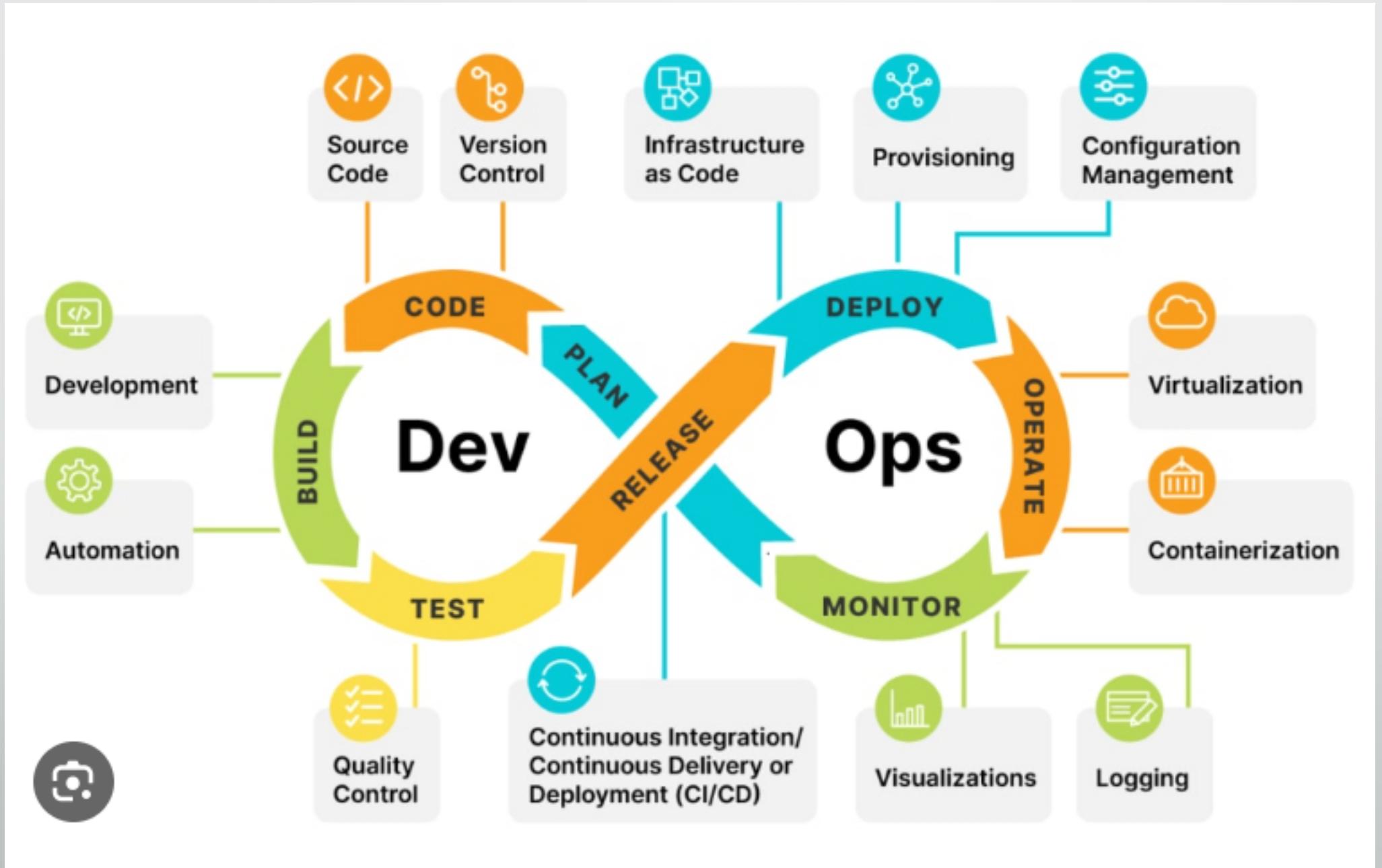
- **Automation of the software development lifecycle:** This includes automating testing, builds, releases, the provisioning of development environments, and other manual tasks that can slow down or introduce human error into the software delivery process.
- **Collaboration and communication:** A good DevOps team has automation, but a great DevOps team also has effective collaboration and communication.
- **Continuous improvement and minimization of waste:** From automating repetitive tasks to watching performance metrics for ways to reduce release times or mean-time-to-recovery, high performing DevOps teams are regularly looking for areas that could be improved.
- **Hyper focus on user needs with short feedback loops:** Through automation, improved communication and collaboration, and continuous improvement, DevOps teams can take a moment and focus on what real users really want, and how to give it to them.

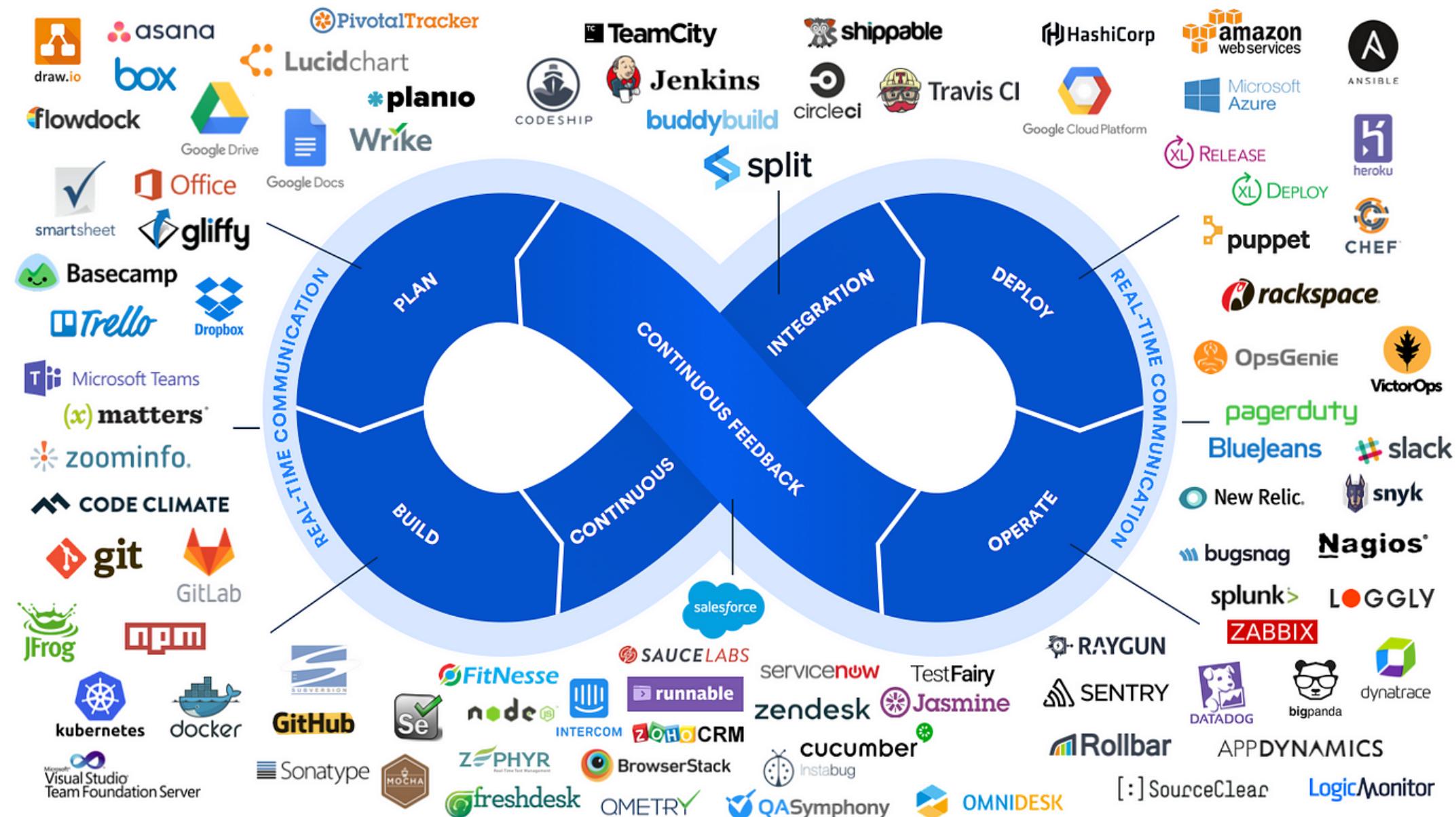
By adopting these principles, organizations can improve code quality, achieve a faster time to market, and engage in better application planning.

DevOps Practices

DevOps covers a wide range of practices across the application lifecycle.

- **Version Control:** The fundamental practice of tracking and managing every change made to source code and other files. Version control is closely related to source code management.
- **Agile:** Agile development means taking iterative, incremental, and lean approaches to streamline and accelerate the delivery of projects.
- **Continuous Integration (CI):** The practice of regularly integrating all code changes into the main branch, automatically testing each change, and automatically kicking off a build.
- **Continuous Delivery (CD):** Continuous delivery works in conjunction with continuous integration to automate the infrastructure provisioning and application release process. They are commonly referred to together as CI/CD.
- **Shift Left:** A term for shifting security and testing much earlier in the development process. Doing this can help speed up development while simultaneously improving code quality.





DevSecOps

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

meyala karanne app eke muladima security part add krla software eka hondata protect krna eka

- 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.**

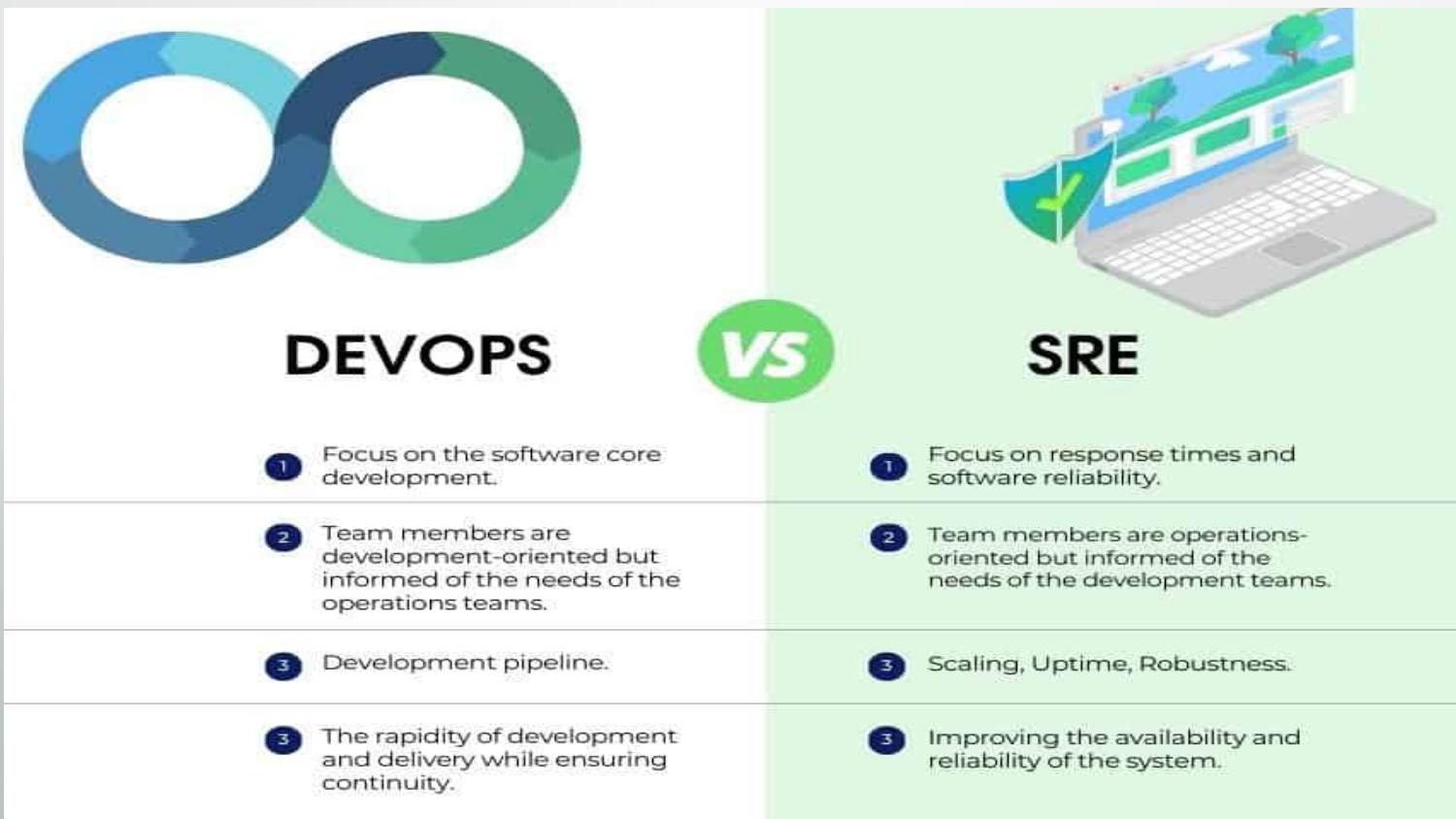
How DevOps can benefit from AI and ML?

- Artificial intelligence (AI) and machine learning (ML) can assist in analyzing test data, identifying coding anomalies that could lead to bugs, as well as automating security and performance monitoring to detect and proactively mitigate potential issues.
- AI and ML can find patterns, figure out the coding problems that cause bugs, and alert DevOps teams so they can dig deeper.
- DevOps teams can use AI and ML to sift through security data from logs and other tools to detect breaches, attacks, and more. Once these issues are found, AI and ML can respond with automated mitigation techniques and alerting.

Devopsla krnne build process eka automate krana eka. mekedi meyla integration wala indla release krna ekata wenknma. SREla krnne ethanin passe availability eka ensure karana eka.meyala linux , python wage dewal danan innona. hadissiyewath automate krpu eke awlk awa gmn hadann manually knoledge ekath thynnona

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.

- 1. Embracing risk
- 2. Simplicity
- 3. Service Level Objectives
- 4. Eliminating toil
- 5. Monitoring distributed systems
- 6. Automation
- 7. Release engineering



Observability in Site Reliability Engineering

Observability is a process that prepares the software team for uncertainties when the software goes live for end users. Site reliability engineering (SRE) teams use tools to detect abnormal behaviors in the software and, more importantly, collect information that helps developers understand what causes the problem. Observability involves collecting the following information with SRE tools.

- **Metrics:** Metrics are quantifiable values that reflect an application's performance or system health. SRE teams use metrics to determine if the software consumes excessive resources or behaves abnormally.
- **Logs:** SRE software generates detailed, timestamped information called logs in response to specific events. Software engineers use logs to understand the chain of events that lead to a particular problem.
- **Traces:** Traces are observations of the code path of a specific function in a distributed system. Traces consist of an ID, name, and time. They help software developers detect latency issues and improve software performance.

Key Metrics for Site Reliability Engineering

Site reliability engineering (SRE) teams measure the quality of service delivery and reliability using the following metrics.

- Service-Level Objectives (SLOs) are specific and quantifiable goals that you are confident the software can achieve at a reasonable cost to other metrics, such as the following:
 - Uptime, or the time a system is in operation / System throughput / System output / Download rate, or the speed at which the application loads
- Service-Level Indicators (SLIs) are the actual measurements of the metric an SLO defines. In real-life situations, you might get values that match or differ from the SLO. For example, your application is up and running 99.92% of the time, which is lower than the promised SLO of 99.95%.
- Service-Level Agreements (SLAs) are legal documents that state what would happen when one or more SLOs are not met. For example, the SLA states that the technical team will resolve your customer's issue within 24 hours after a report is received. If your team could not resolve the problem within the specified duration, you might be obligated to refund the customer.



One of the key components of software engineering process is Testing [Quality Management]...

Without Test Automation achieving the success of Agile & DevOps Methodologies are nearly Impossible...

Next Week...

- Cloud Computing Platforms
 - Introduction to Cloud Platforms: AWS, GCP, Azure & Hybrid Cloud,
 - Testing on Cloud Environments,
 - SaaS, PaaS, & IaaS Offerings,
- Security
 - Introduction to Cyber Security,
 - Security Testing,



Thank You !!!

Tutorial – 21/04/2024

Q1: Explain the terms RFP, MSA, & SOW?

Q2: Name currently used software development methodologies...

Q3: Name the four core values of Agile methodology...

Q4: Briefly explain DevOps...

Q5: What are the 7C's of DevOps?

Q6:

Q7:

Q8:

Tutorial – 21/04/2024 [Answers - I]

Q1: Explain the terms RFP, MSA, & SOW?

A request for proposal (RFP) is both the process and documentation used in soliciting bids for potential business or IT solutions required by an enterprise or government agency.

A Statement of Work, often known as an SOW, is a business agreement that outlines deliverables and project goals.

A master service agreement is a legal contract that establishes fundamental agreements between two parties. MSAs allow vendors and clients to agree on basic terms at the outset of a business relationship before any business commences.

Q2: Name currently used software development methodologies...

- Waterfall
- Agile
- DevOps
- Rational Unified Process

Tutorial – 21/04/2024 [Answers - II]

Q3: Name the four core values of Agile methodology...

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan

Q4: Briefly explain DevOps...

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

Q5: What are the 7C's of DevOps?

- Continuous development and delivery.
- Continuous integration.
- Continuous testing.
- Continuous monitoring.
- Continuous feedback.
- Continuous deployment.
- Continuous operations.

Tutorial – 21/04/2024 [Answers - III]

Q6:

Q7:

Q8:

Tips to Face a Successful Interview...

- Prerequisites : To get called for an Interview,
- Preparation for the called Interview,
- Interview Day,
 - Focus on creating a positive first impression,
 - ✓ Punctuality,
 - ✓ Read the job description for the interview one more time,
 - Dress professionally to your interview,
 - ✓ It gives out your intentions towards the job,
 - ✓ Virtual interviews require more preparation,
 - Throughout the Interview – Have a good posture. Always smile. Demonstrate good manners.
 - Start by answering the First Question Well – “***Tell me about yourself***
 - ✓ Skills & Qualities,
 - ✓ Experience or Qualifications,
 - ✓ Significant achievements,
 - ✓ Type of person you are and what you're going to do if hired,



Thank You Again!!!