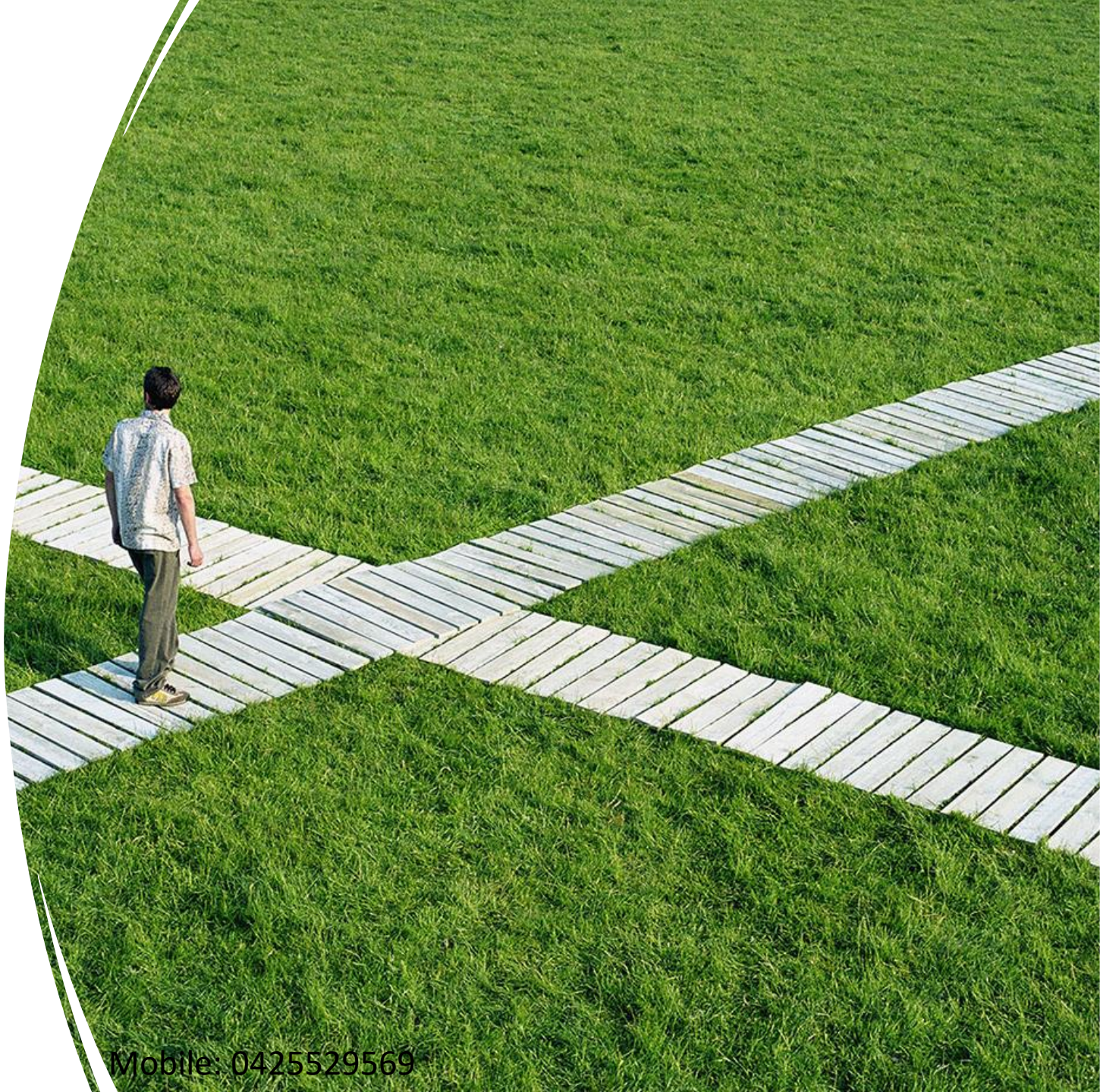
Welcome to DevOps Session

Sunil Dhurve



Mobile: 0425529569

ACKNOWLEDGEMENT OF COUNTRY

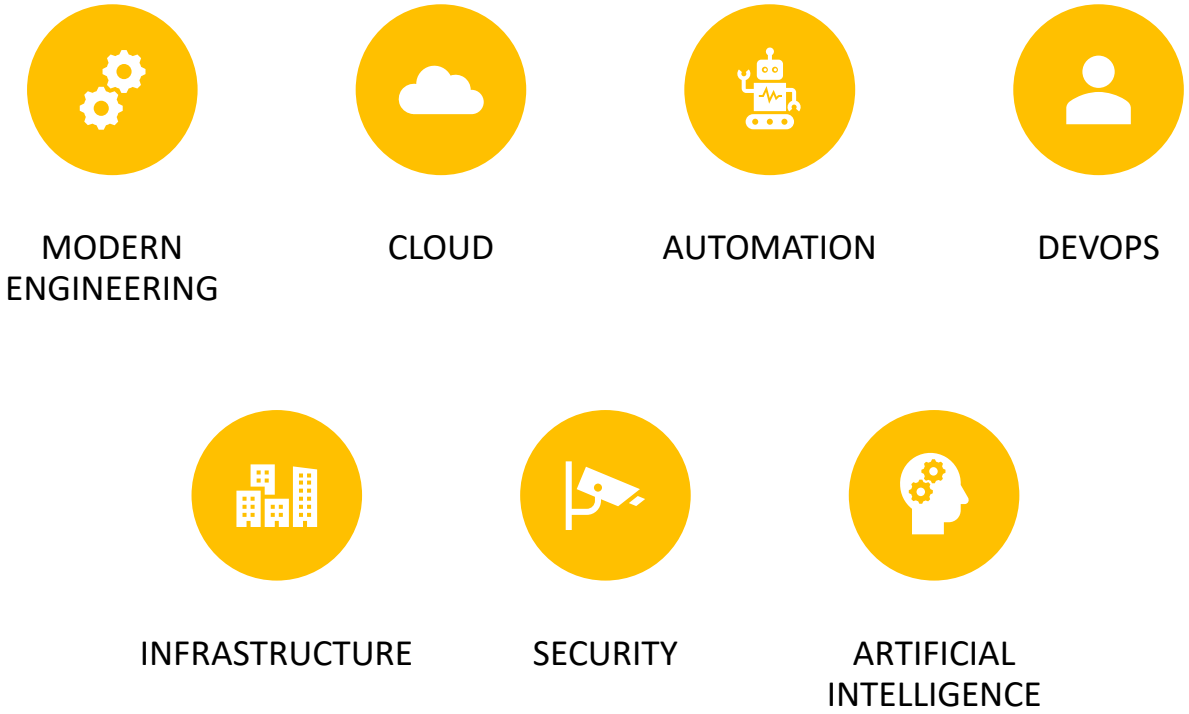
I would like to begin by acknowledging the Traditional Custodians of the land on which we meet today. I would also like to pay my respects to elders past, present and emerging.

#MORETOGETHERNOW

Agenda

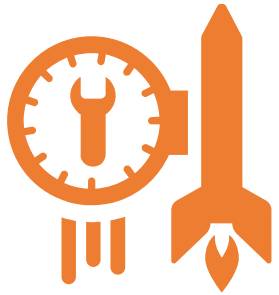
1. Importance Of Today's Technology & Modern Engineering
2. Waterfall vs Agile in SDLC
3. How does Agile and DevOps fit in?
4. Devops in Detail
5. Cloud Computing
6. Devops Practice
7. Devops Case Study

How today’s technology is important to you?

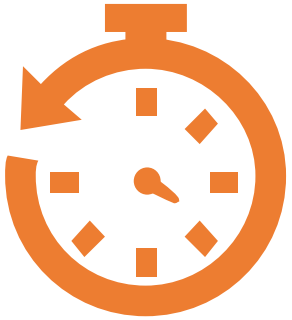


- It is important to be knowledgeable about key technologies such as Modern Engineering, Cloud, and Automation. By utilizing these technologies, teams can increase productivity and improve the quality of their work. Automation should be implemented wherever possible, and DevOps should be used for continuous integration and delivery in custom projects. Understanding the impact of Cloud on the enterprise and the future of technology is crucial for preparing for the future. Intelligent Automation can also significantly improve the customer experience, reduce processing costs, and improve decision-making across the business.

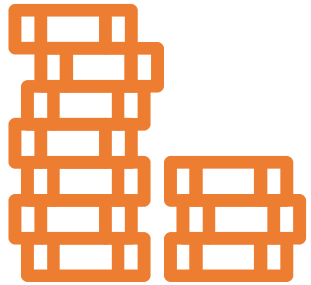
Fifty deployments per day is possible



Amazon deploys new features to production every 11 seconds



Uber deploys more than 500 times per day



Flipboard saves $19M every month by optimizing their workloads in the cloud

Netflix uses chaos engineering to resolve platform issues before they impact customers


PRODUCT AND CUSTOMER CENTRIC
AGILE
DECENTRALIZED EXECUTION
VALUE PRIORITIZED
MULTI-DISCIPLINARY TEAMS
DEVOPS
DELIVER–REFLECT–IMPROVE
BUILT-IN QUALITY
AUTOMATED PIPELINES (CONTINUOUS EVERYTHING)
CLOUD-NATIVE
DECOUPLED & DISTRIBUTED
RESILIENT BY DESIGN
ELASTIC SUPPLY OF RESOURCES
MODERN ENGINEERING

BY DOING LESS…

Methodology	App. Architecture	Deployment Model	App. Infrastructure
WATERFALL	MONOLITHS & N-TIER	SERVERS	DATACENTRE HOSTED

AND DOING MORE…

Methodology	App. Architecture	Deployment Model	App. Infrastructure
AGILE & DEVOPS	MICRO-SERVICES	CONTAINERS & FUNCTIONS	CLOUD PLATFORMS

Software Delivery style

Software systems are complex. SDLC methodologies help manage the level of complexity in the systems by defining the activities performed at each level of development.

Two of the most common which will be referenced throughout this training are:

Sequential (Waterfall)

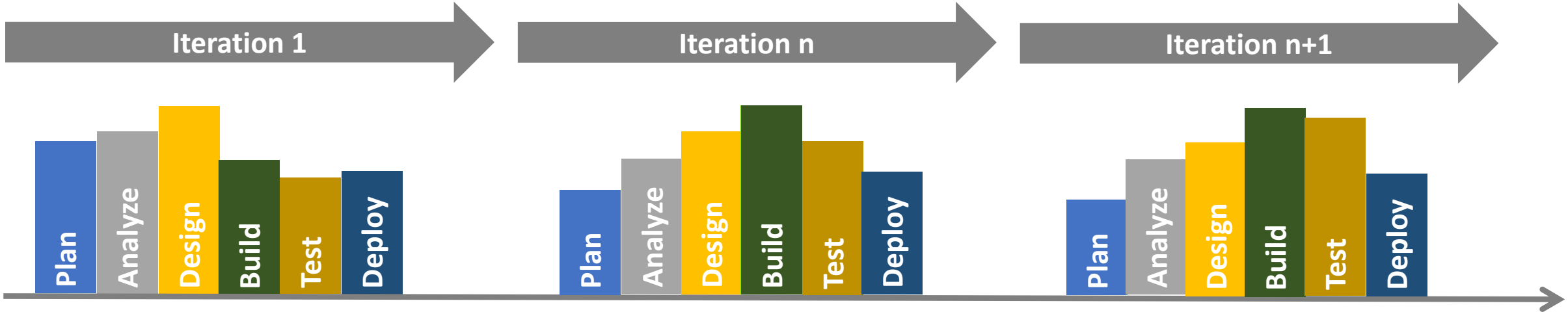
Agile

The Software Development Lifecycle (SDLC) model provides a sequence of activities and a structure for executing the activities needed for developing software applications. This model has six stages.

Agile vs. traditional approach

Delivery Methodology

Agile Development is focused on an iterative (addressing all aspects of the lifecycle in each iteration), incremental and flexible approach to software development.

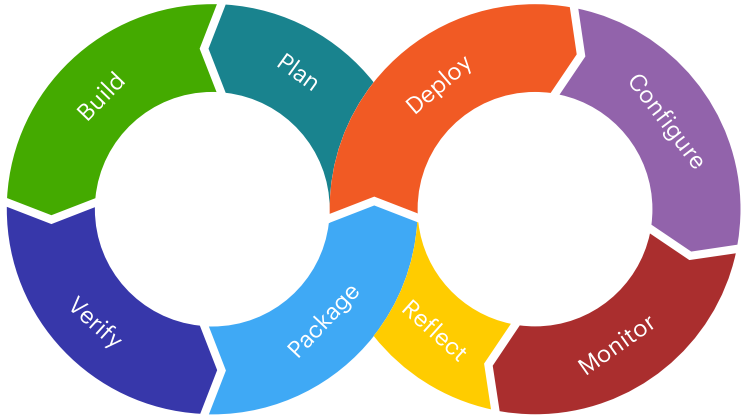


Achieving the product vision with agile and DevOps

Agile focuses on the interactions between the CUSTOMER (business) and DEVELOPMENT

DevOps focuses on the interactions between DEVELOPMENT and OPERATIONS

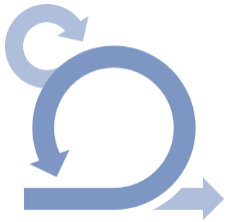
IDEATION



VALUE

Agile & DevOps both focus on nurturing **self-organizing, multi-disciplinary** teams, with all the skills and capabilities needed to achieve the **product vision.**

What are the agile principles



Our highest priority is to satisfy the customer through EARLY AND CONTINUOUS DELIVERY of valuable software.

Welcome changing requirements, even late in development. Agile processes HARNESS CHANGE for the customer's competitive advantage.

Deliver working software FREQUENTLY, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

Businesspeople and developers must WORK TOGETHER daily throughout the project.

Build projects around MOTIVATED INDIVIDUALS. Give them the environment and support they need and trust them to get the job done.

The most efficient and effective method of conveying information to and within a development team is FACE-TO-FACE CONVERSATION.

WORKING SOFTWARE is the primary measure of progress.

Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a CONSTANT PACE INDEFINITELY.

Continuous attention to TECHNICAL EXCELLENCE and good design enhances agility.

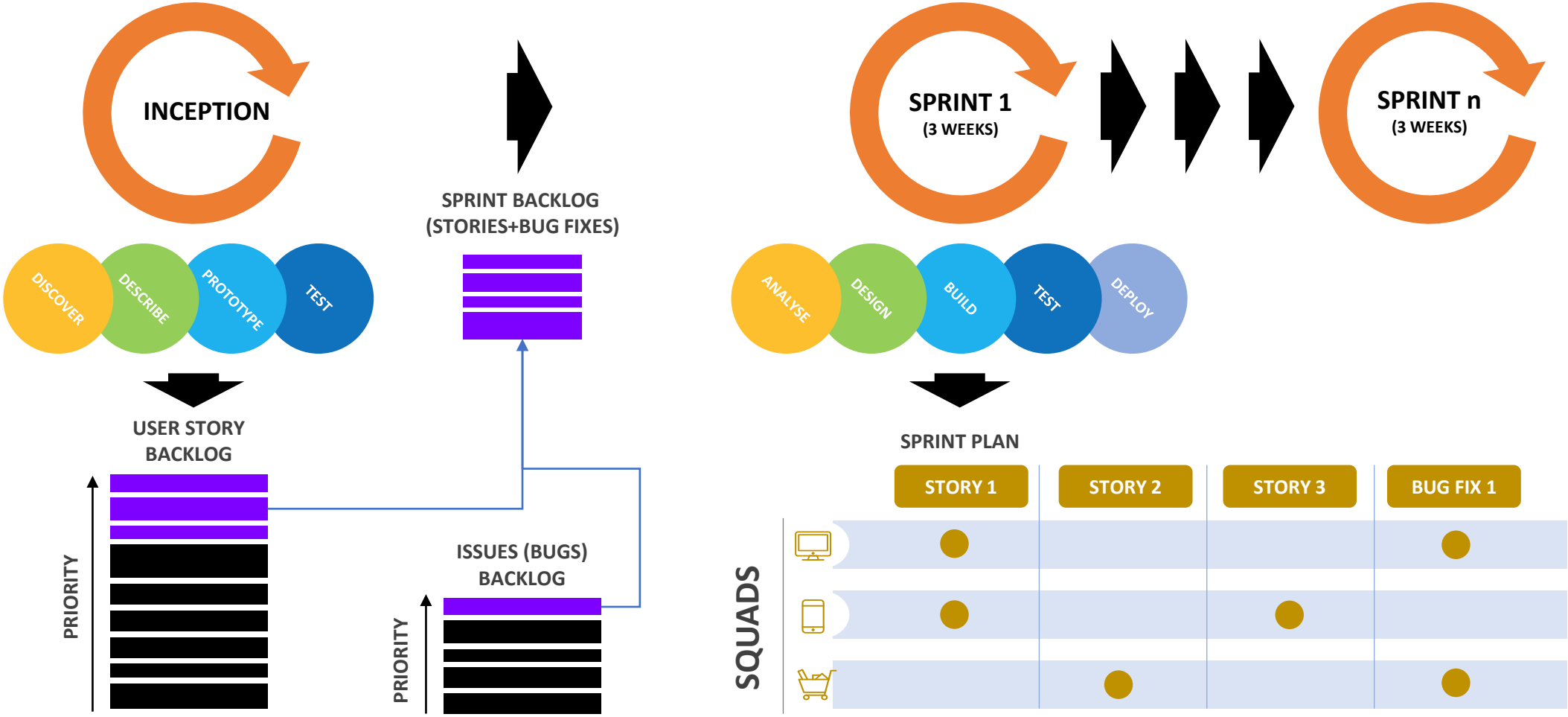
SIMPLICITY – the art of maximizing the amount of work not done – is essential.

The best architectures, requirements, and designs emerge from SELF-ORGANIZING TEAMS.

At regular intervals, the team REFLECTS on how to become more effective, then tunes and adjusts its behavior accordingly.

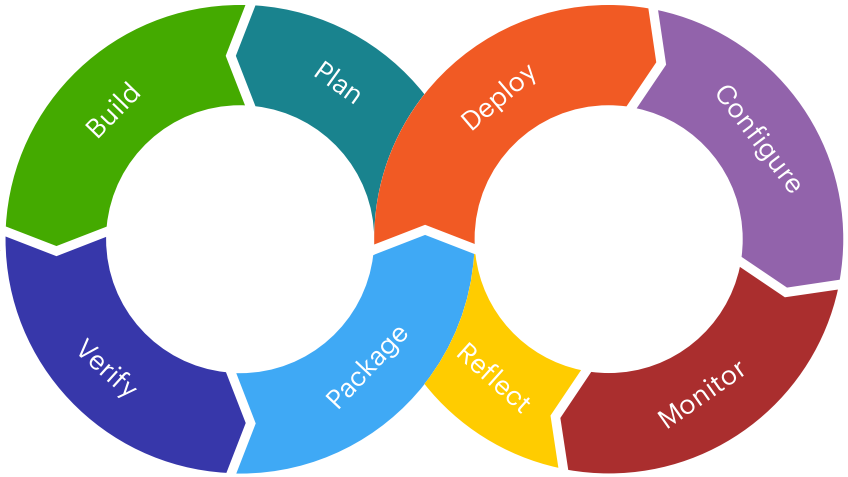
(*) The Agile Manifesto: https://agilemanifesto.org/

What is iterative development?



What is DevOps

DEV wants CHANGE

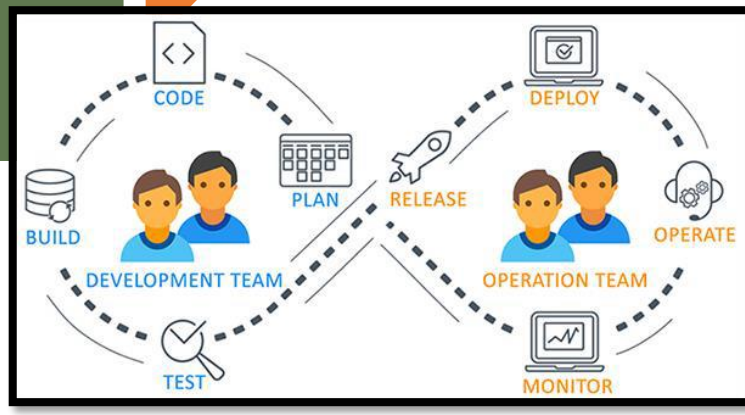


OPS wants STABILITY

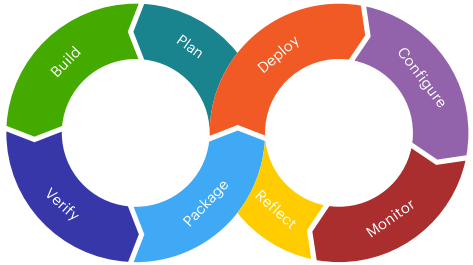
EVERYONE wants...

Stable and operable software
Consistent and optimized release process
Conformance to infrastructure and platform standards

Low risk, high throughput of change
Confident code and software integrity
Highly available and predictable environments



DevOps principles

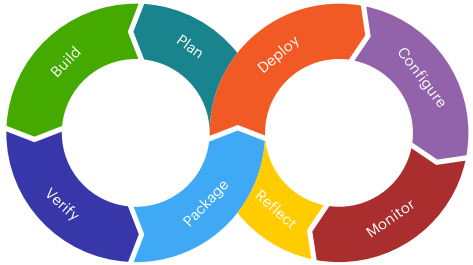


Although not formally set in stone it is commonly accepted that DevOps ideals are embodied by the following six principles:

- Reduce organizational **silos**
- Accept **failure** as normal
- Implement gradual **change**
- **Automate** everything
- **Measure** everything
- **Improve** continuously

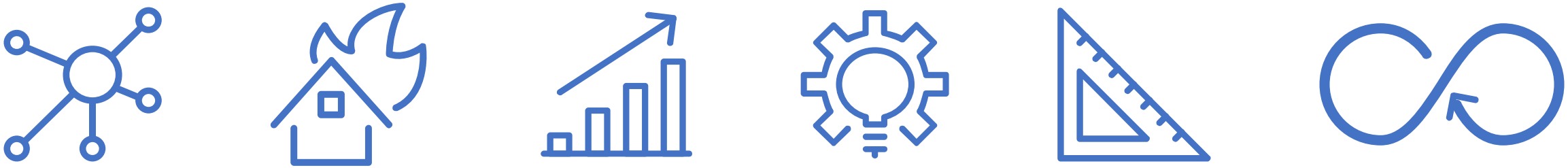
What DevOps is	What Devops is Not
1. DevOps is a mindset to work in a certain way that could solve the problem of Developers and IT Operation teams within the organization. 2. DevOps is the union of people, process, and products to enable continuous delivery of value to our end users. 3. DevOps is a culture and technical movement that emphasizes using of various	• DevOps isn't a Software • DevOps isn't a Job Title (like developer or tester or administrator) • DevOps isn't a process or a technology.

DevOps principles



With DevOps, organizations can release faster and more frequently, react rapidly to customer feedback, and ultimately, EXPERIMENT easily and without fear of change because they know they can monitor closely the impact after any change is introduced, and they can trivially roll back any change if necessary.

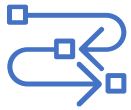
EXPERIMENT WITHOUT REGRET



DevOps automation practices

LIFE-CYCLE MANAGEMENT

Traceability of development life-cycle activities, from requirements to releases
Strong configuration management with consistent branching & versioning



CONTINUOUS INTEGRATION

Continuous integration(*) with choreographed pipelines
Exhaustive code inspection inclusive of security matters

CONTINUOUS DEPLOYMENT

Automated deployments up to production
Deployments are trivial to release and trivial to rollback

CONTINUOUS TESTING

Automated tests, early and often, aligned with acceptance criteria
Executed in production-like environments

CONTINUOUS INFRASTRUCTURE

Software-defined infrastructure, under configuration management
Resilient by design, highly available, easy to distribute and to scale

CONTINUOUS MONITORING

Pervasive data collection across all environments and functions
Automated operators and control plane, capable of self-monitor and self-tune

What is Cloud?

Every one of us at has an important role to play, even if you are not directly involved with Cloud First or SaaS cloud work. We must all understand cloud, its impact on the enterprise and future of technology, so we can help our clients, and our business, use its power today to prepare for the future.

Cloud computing is a model in which organizations access a shared pool of configurable computing resources — such as servers, networks, storage, development tools, or applications.

Five key characteristics*:

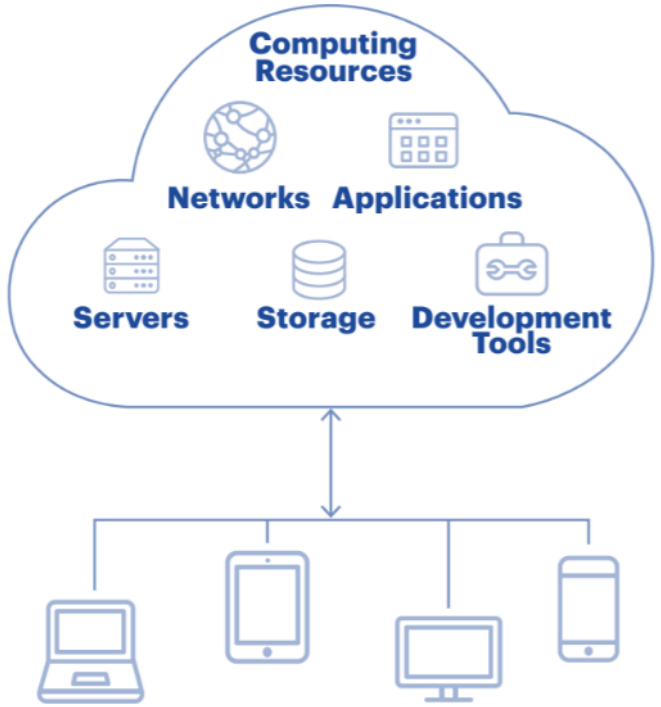
On Demand Self-Service

Shared Resource Pooling

Rapid Elasticity

Measured Service

Internet Access



Cloud deployment models

Cloud is a model of computing that can be implemented several ways.

Four Deployment Models:

Cloud service models

Cloud service providers offer varying service levels to organizations, from the entire infrastructure to a platform for application development or certain cloud-based software applications.

Three Service Models:

SaaS

USE a full-feature software application in the cloud & customize with your own data

PaaS

Access a software development environment in the cloud to **BUILD** your own applications

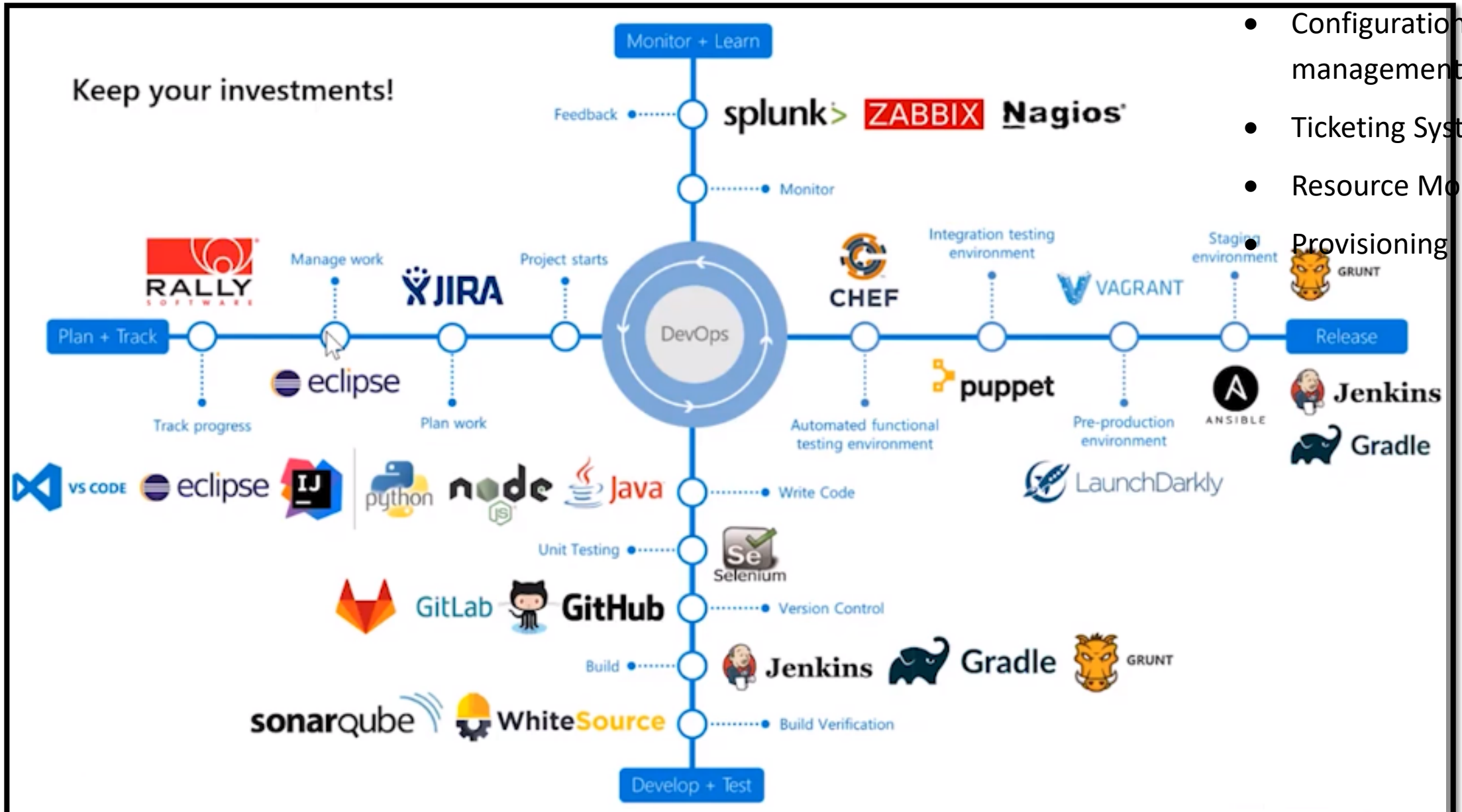
IaaS

MOVE infrastructure (e.g., hardware, servers, storage) to the cloud

DevOps Tools

Tools Required for

- Version Control System
- Configuration management
- Ticketing System
- Resource Monitoring
- Provisioning



Azure DevOps

- You can use one or more of the following features based on your business needs:

- 1. **Azure Boards** delivers a suite of Agile tools to support planning and tracking work, code defects, and issues using Kanban and Scrum methods.

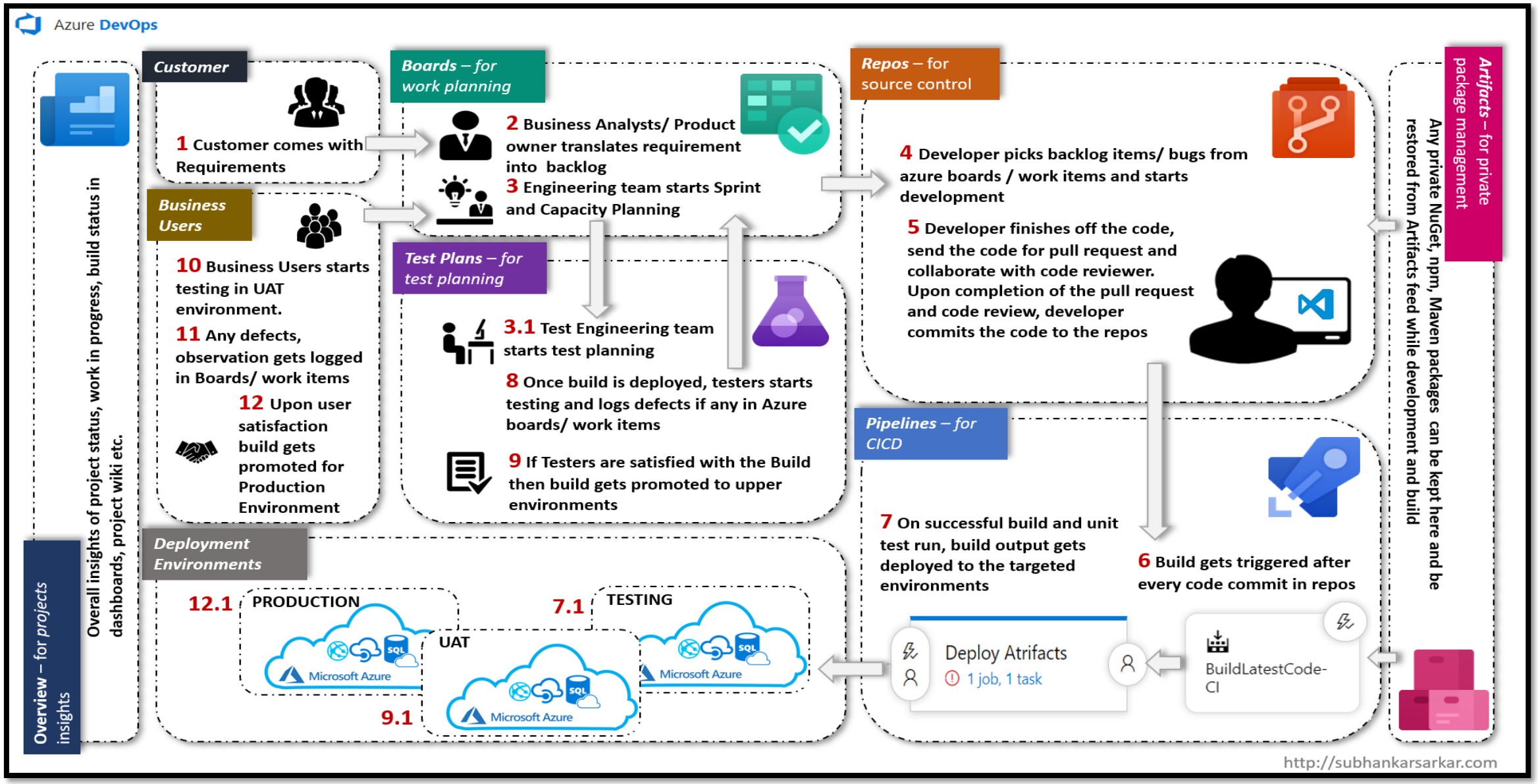
- 2. **Azure Repos** provides Git repositories or Team Foundation Version Control (TFVC) for source control of your code.

- 3. **Azure Pipelines** provides build and release services to support continuous integration and delivery of your apps.

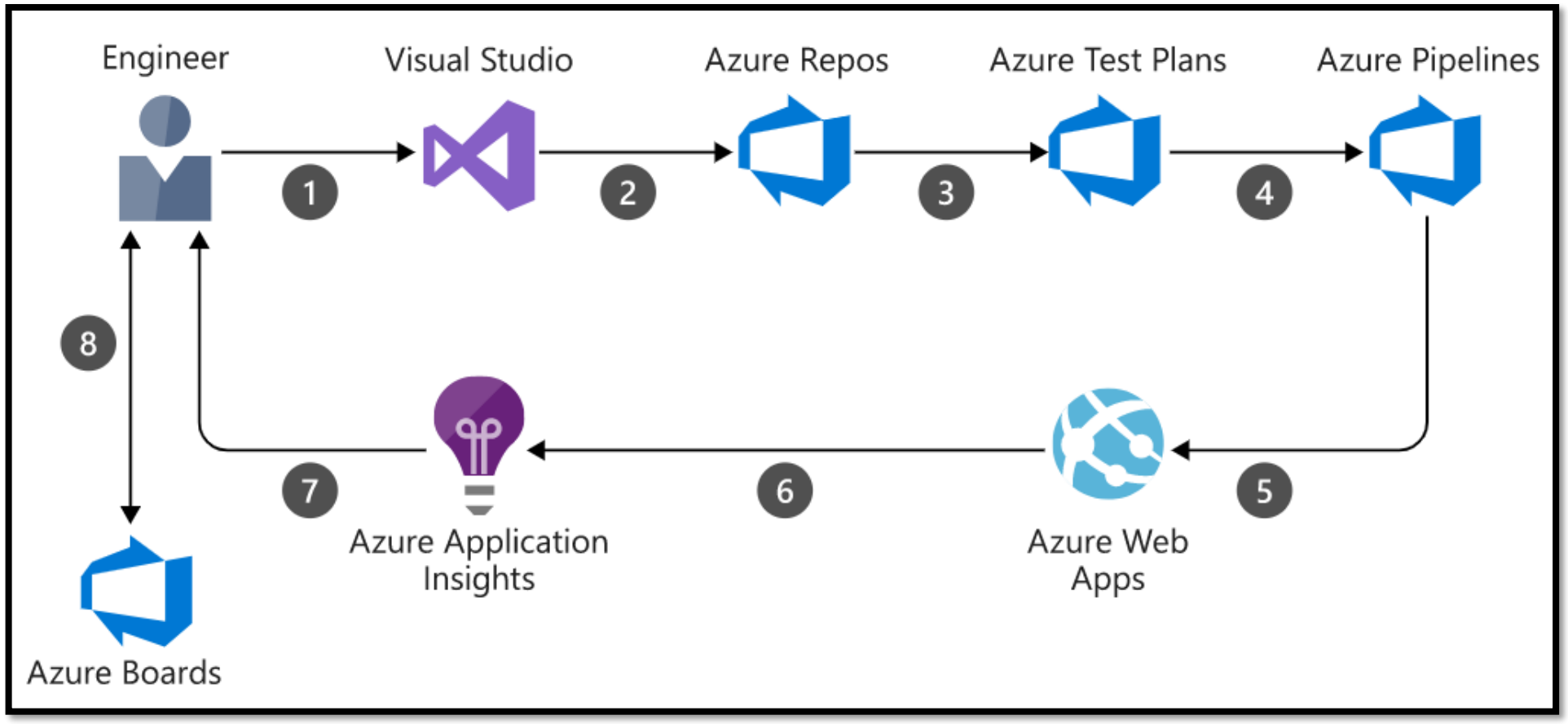
- 4. **Azure Test Plans** provides several tools to test your apps, including manual/exploratory testing and continuous testing.

- 5. **Azure Artifacts** allows teams to share Maven, npm, and NuGet packages from public and private sources and integrate package sharing into your CI/CD pipelines.

Azure DevOps Tools and Project Life Cycle



Azure DevOps Tools Life Cycle



Anonymous Global Hospitability Company DevOps journey

Challenges

- Existing IT infrastructure
- Rapid increase of clients
- Required frequent updates

Challenges

- IT infrastructure management
- Frequent modifications
- Continuous monitoring

Solution

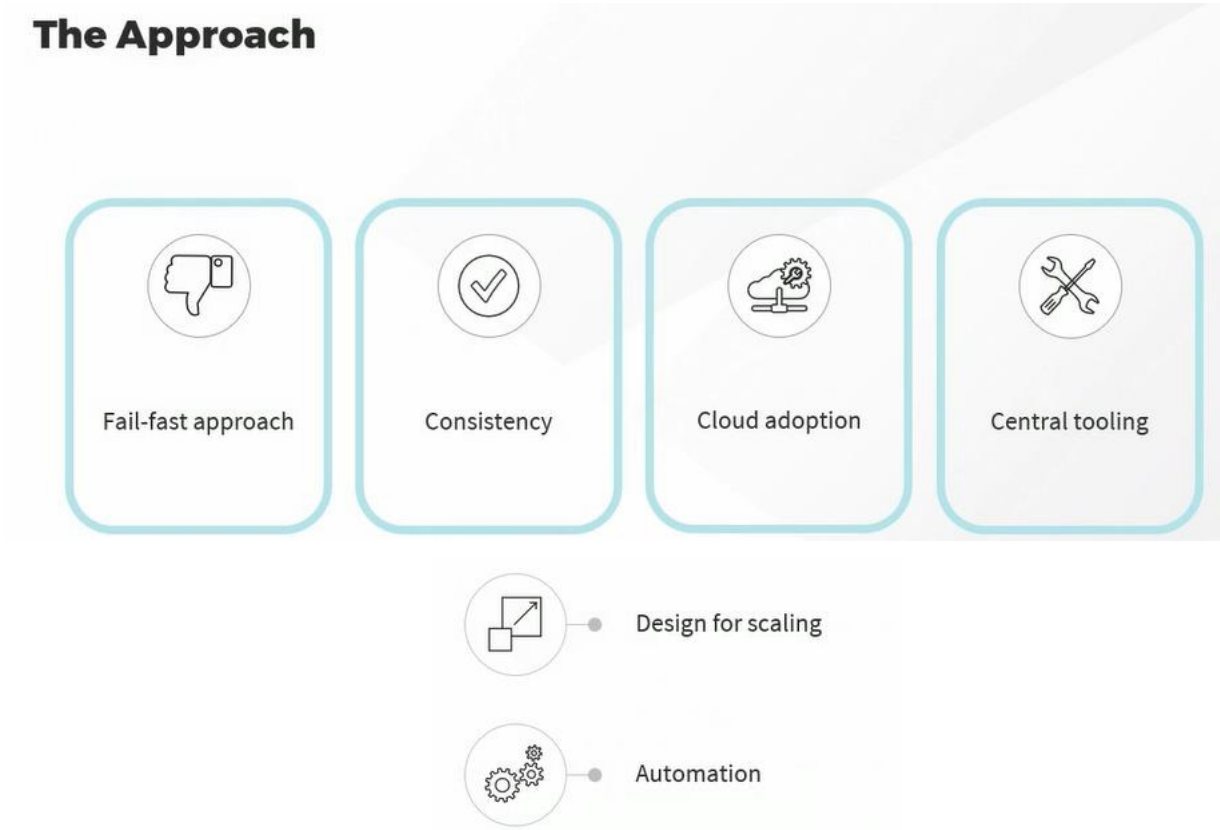
Key takeaways

Get management of their IT infrastructure and software under control.
New-found high availability through load balancing and caching.
Vastly improved their ability to enhance their solution frequently and without disruptions to the business process.

set of tools that needs to be implemented to make that happen.
Now, in this case, the company used

Jenkins for CI/CD,
Ansible for configuration management,
Docker for containerization,
the ELK stack for log management,
and Stash for source code management along with other.

Capital Ones DevOps journey



Solution

was to reduce the time to build their apps, provision their infrastructure, and deploy their solutions. through automation, CI/CD pipelines, and process around where and how to code.

DevOps teams had whatever they needed. Implementing a DevOps strategy.

Attained a much more collaborative culture as a result of their DevOps implementation, and
key focus on customer feedback, having that collaboration means a lot of engagement from other employees within your organization.

The DevOps team is transparent, open to feedback, and communicating a lot.
employees within the organization being much more engaged in the process.
were able to notice a much higher degree of quality in software solution.
DevOps automatically results in a higher quality team.