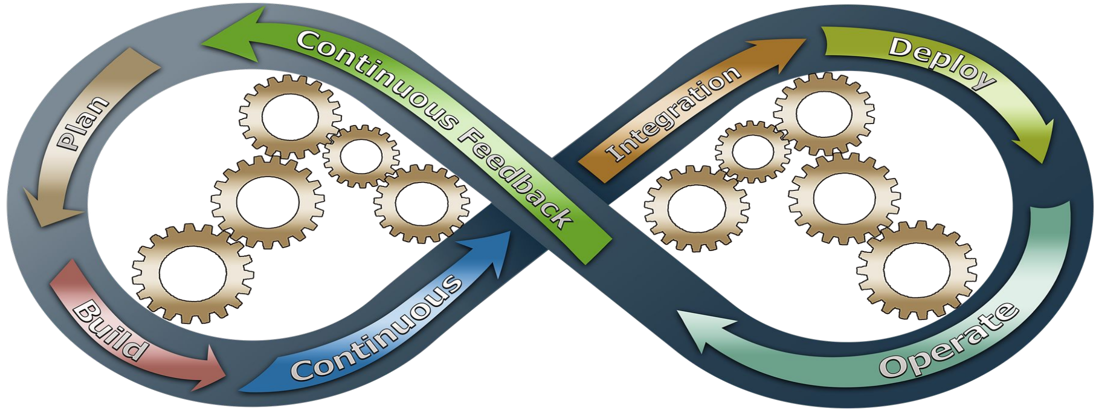
DevOps Tools and Services on AWS



AWS Refresher

- EC2
- S3
- VPC
- Lambda
- Cloudformation
- Cloudwatch
- IAM and security
- SNS

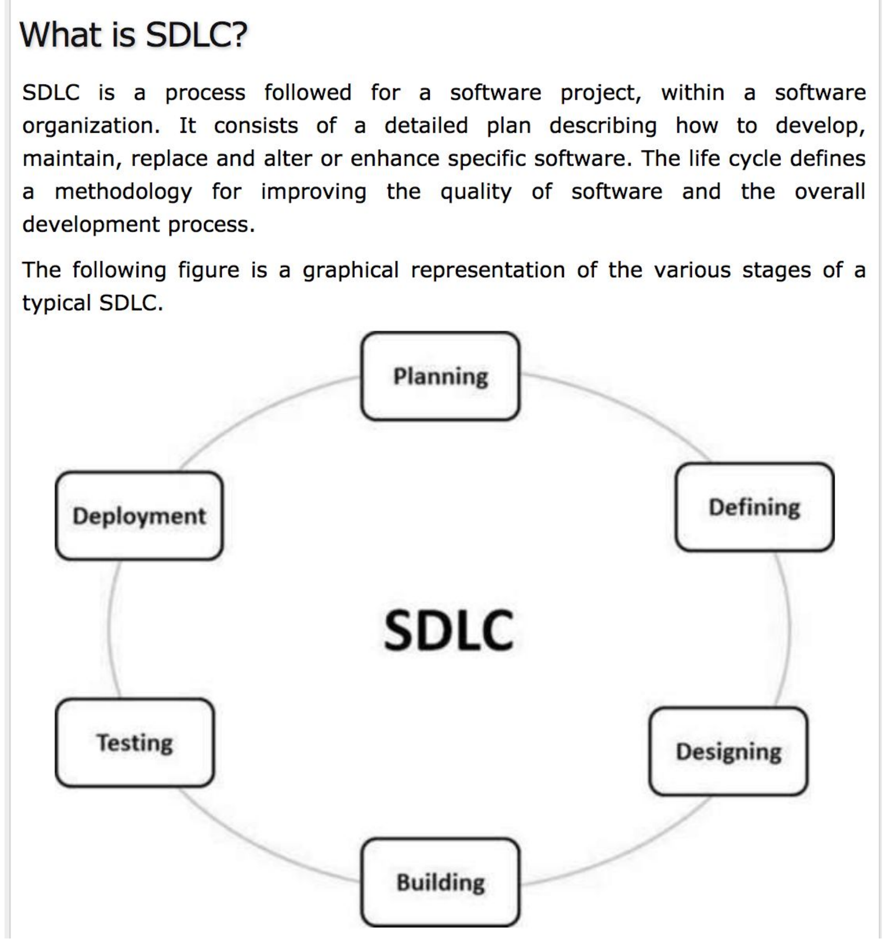
Waterfall, Agile and DevOps

https://www.thoughtworks.com/insights/blog/path-devops

https://www.guru99.com/waterfall-vs-agile.html

https://www.guru99.com/agile-vs-devops.html

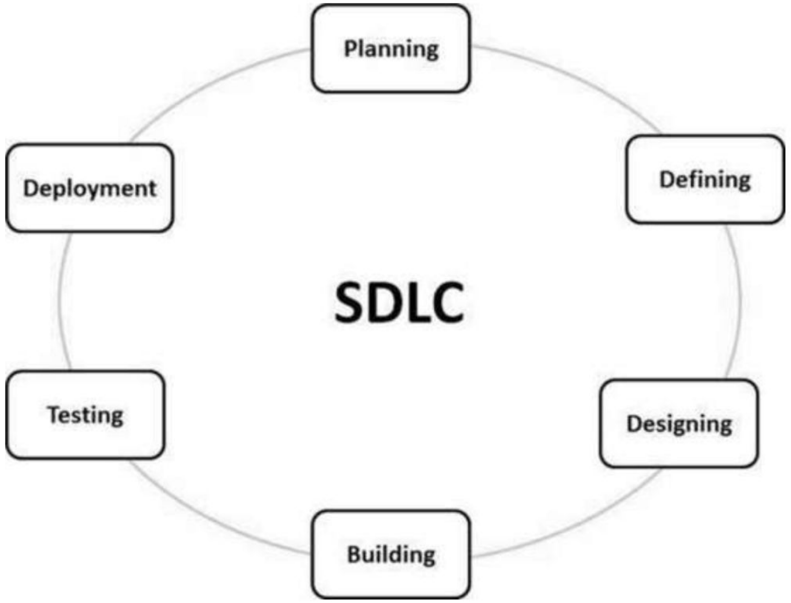
SDLC



What is SDLC?

SDLC is a process followed for a software project, within a software organization. It consists of a detailed plan describing how to develop, maintain, replace and alter or enhance specific software. The life cycle defines a methodology for improving the quality of software and the overall development process.

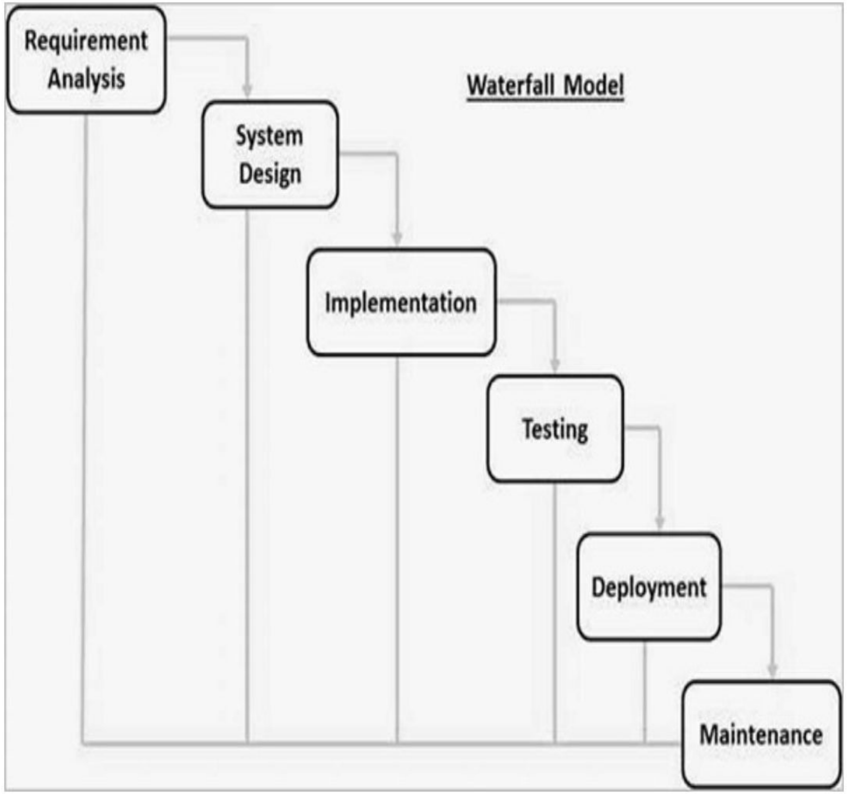
The following figure is a graphical representation of the various stages of a typical SDLC.



Most IT Software delivery in the Old days

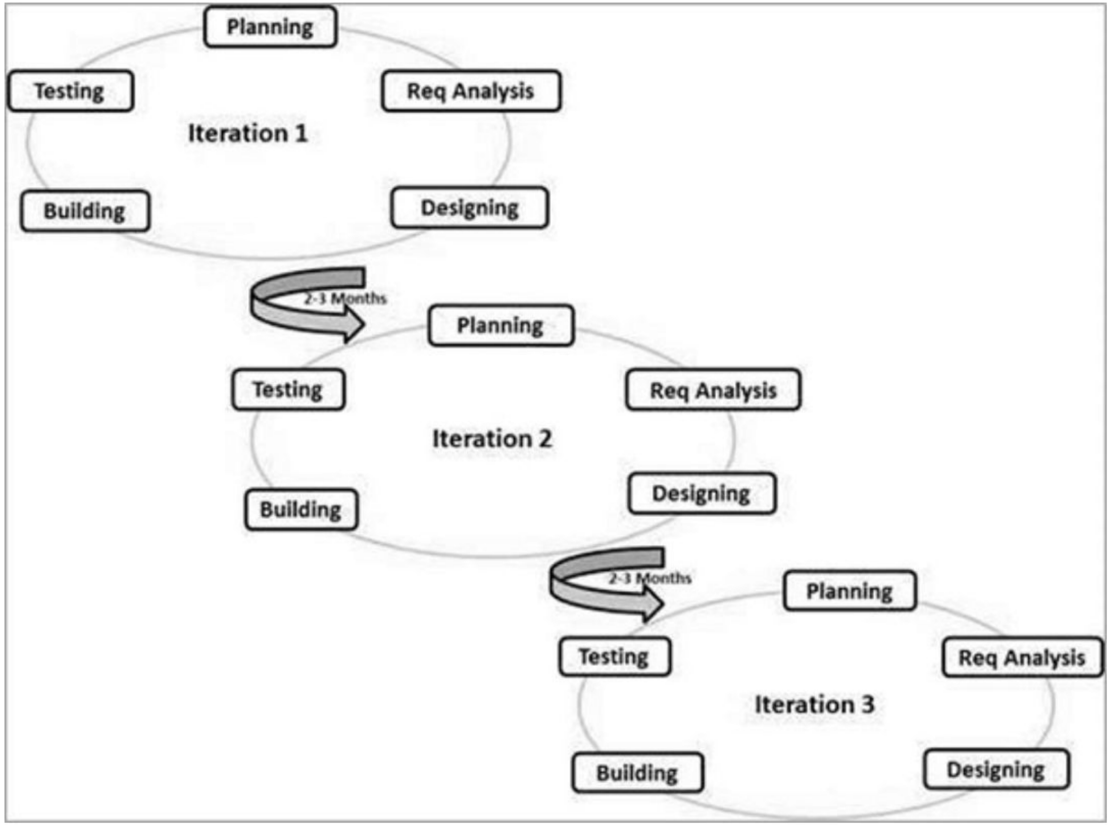
Waterfall method:

There is no scope of changing the requirements once the project development starts.



Agile SDLC

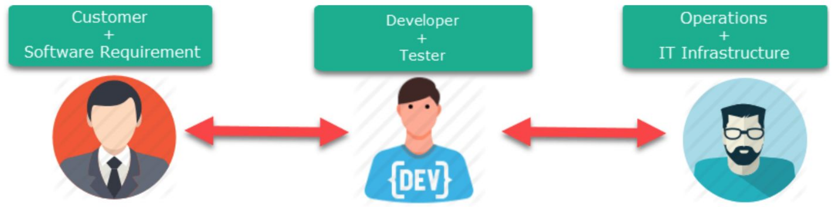
Agile is quite a flexible method which allows changes to be made in the project development requirements even if the initial planning has been completed.



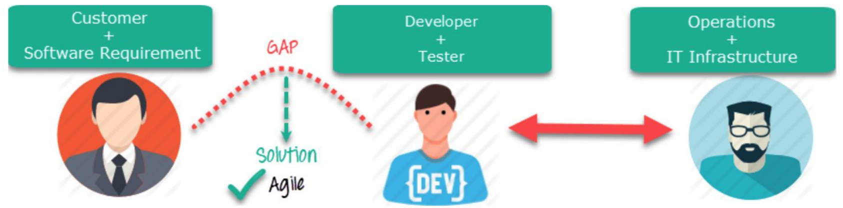
Waterfall, Agile, Devops

- Agile methods brought shorter cycles and faster integration of codebase.
- Still, code wasn't in production, it was just being integrated.
- Devops solves this problem.

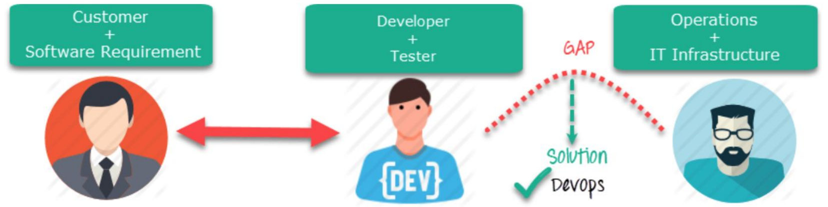
Stakeholders and communication chain in a typical IT process.



Agile addresses gaps in Customer and Developer communications



DevOps addresses gaps in Developer and IT Operations communications



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 Culture Look Like?

Even with the best tools, DevOps is just another buzzword if you don't have the right culture.

The primary characteristic of DevOps culture is **increased collaboration** between the roles of development and operations. There are some important cultural shifts, within teams and at an organizational level, that support this collaboration.

- Developers and Ops team are co-located
- Share responsibilities to get things done & Maintain systems
- Don’t blame each other , work together to solve problems
- Automate to focus on high value tasks

https://www.ca.com/en/blog-automation/what-is-devops-culture.html

https://martinfowler.com/bliki/DevOpsCulture.html

DevOps

The DevOps movement is focused on **delivering software faster and more efficiently**, without breaking things as often. To meet these goals, businesses need to change **organizational culture and structure**, as well as the **tools and processes** they use.

Developers and Operations team work together. Communicate well . This enables software to be delivered faster.

DevOps Introduction Reference

https://aws.amazon.com/devops/what-is-devops/

CONTINUOUS INTEGRATION

Continuous Integration (CI) is a development practice that requires developers to integrate code into a shared repository several times a day. Each check-in is then verified by an automated build, allowing teams to detect problems early.

By integrating regularly, you can detect errors quickly, and locate them more easily

Reference: https://www.thoughtworks.com/continuous-integration

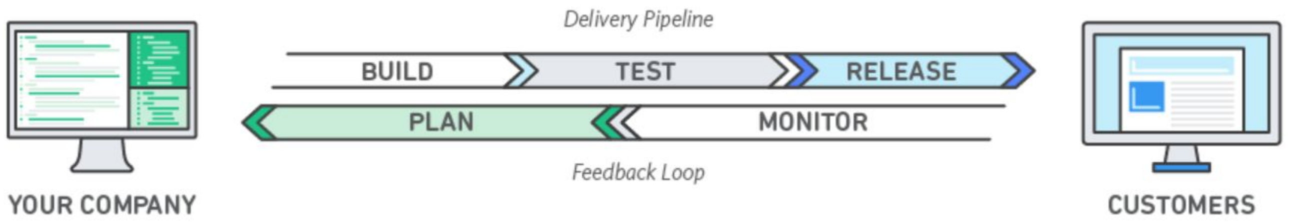
CONTINUOUS DELIVERY

Continuous Delivery is a software development discipline where you build software in such a way that the software can be released to production at any time.

You're doing continuous delivery when:

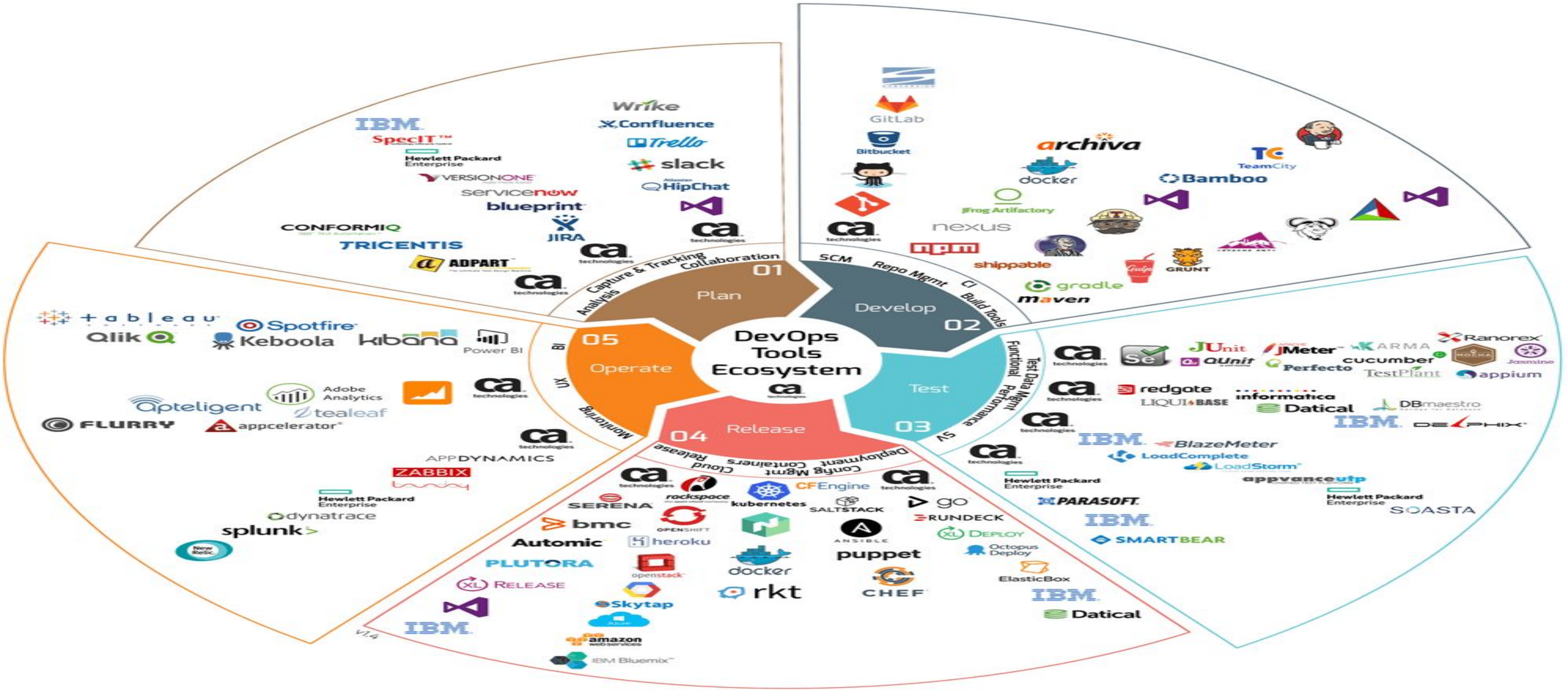
- Your software is deployable throughout its lifecycle
- Your team prioritizes keeping the software deployable over working on new features
- Anybody can get fast, automated feedback on the production readiness of their systems any time somebody makes a change to them
- You can perform push-button deployments of any version of the software to any environment on demand

https://martinfowler.com/bliki/ContinuousDelivery.html


Delivery Pipeline
BUILD
TEST
RELEASE
PLAN
MONITOR
Feedback Loop
YOUR COMPANY
CUSTOMERS

DevOps Services we'll look at

1. Infrastructure as code:Cloudformation (Day 1)
2. Infrastructure as code: Terraform
3. Monitoring and Logging: Cloudwatch Metrics + Logs
4. SCM: Git and Github
5. Serverless Automation: AWS Lambda
6. Configuration Management: Ansible
7. Continuous Integration: Jenkins
8. ChatOps: Slack
9. Alerting : SNS
10. API Automation: AWS Command Line Interface, boto3 AWS SDK
11. Containers: Docker



https://dzone.com/articles/the-ultimate-devops-tools-ecosystem-tutorial-part-2

Every companies have different combo of tools

Most of the tools in the previous slides are open source. We can freely experiment. There are paid alternatives that abstract a lot of management.

A lot of companies combine free and paid tools.

Intro to git

What is git?

https://git-scm.com/book/en/v1/Getting-Started-Git-Basics

Why use git?

LAB 1:

Let’s create a git repo. And push to a remote origin.

1) Create a github account
2) Create a repo and clone it
3) Add some files locally
4) Stage changes, Commit, Push to remote Repo

AWS CLI

The AWS Command Line Interface (AWS CLI) is an open source tool that enables you to interact with AWS services using commands in your command-line shell. With minimal configuration, you can start using functionality equivalent to that provided by the browser-based AWS Management Console from the command prompt in your favorite terminal program.

AWS CLI is used a lot in automation processes.

AWS CLI Commands

List all the s3 buckets

```
aws s3 ls
```

Create an ec2 instance:

```
aws ec2 run-instances --image-id ami-8c1be5f6 --instance-type t2.micro
```

Terraform: Infrastructure as Code

HashiCorp Terraform enables you to safely and predictably create, change, and improve infrastructure. It is an open source tool that codifies APIs into declarative configuration files that can be shared amongst team members, treated as code, edited, reviewed, and versioned.

WRITE

INFRASTRUCTURE AS CODE

PLAN

PREVIEW CHANGES BEFORE APPLYING

CREATE

REPRODUCIBLE INFRASTRUCTURE

Ansible

Ansible is an open-source software provisioning, configuration management, and application deployment tool. It runs on many Unix-like systems, and can configure both Unix-like systems as well as Microsoft Windows. It includes its own declarative language to describe system configuration.

Why Ansible ?

Among the many configuration management tools available, Ansible has some distinct advantages—it’s minimal in nature, you don’t need to install anything on your nodes, and it has an easy learning curve.

Other options are Chef, puppet , Salt

Ansible Highlights

- Uses SSH
- Agentless
- Simple to learn
- Uses Yaml
- Open source