Lesson 1 – introduction and overview

Waterfall Model – Disadvantages

* No working software till late in the development stage
* High risk
* Not good for complex and o-o products
* Not suitable for projects where requirements are constantly changing
* Very hard to make changes later in the stages

Agile methodology

* Broken into several iterations
* Each iteration is between 2 to 8 weeks
* Each iteration will deliver a working product

A diagram of a process

Description automatically generated

Waterfall vs agile

A diagram of a software development process

Description automatically generated

Agile – limitations

* Conflict between develop team and operation team
* During the deploy stage is not streamline and time consuming

DevOps

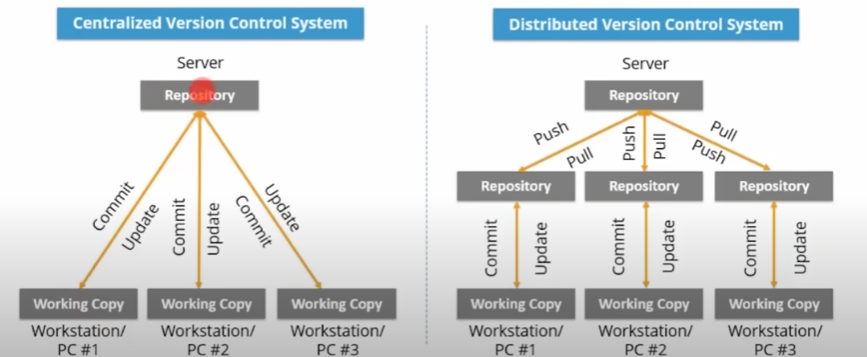
* Integration the developers and the operation team to ensure both sides are working fine
* Something like the middleman?
* A diagram of a company

  Description automatically generated

DevOps stages

* Version control
* Continuous integrations
  + Compile, validate, code review, unit testing, integration
* Continuous delivery
  + Deploy on test server
  + Do UAT
* Continuous Deployment
  + Deploy on production server for release
  + Usually not automatically
  + Configuration management and containerization
* Continuous monitoring FOR all stages

Version control



Centralized vs distributed version control

* Centralized are high risk as it can’t be recovered when the server cashed
* Decentralized are better as its faster for the developers to make changes to their local repo.

Git for version control

* A diagram of a diagram

  Description automatically generated

Continuous integrations using Jenkins

* A diagram of a computer system

  Description automatically generated

Continuous delivery

* A diagram of a delivery process

  Description automatically generated

Continuous deployment

* A computer screen shot of a computer

  Description automatically generated

Configuration management

* A group of text boxes

  Description automatically generated
* Helps to revert
* Easily sync all the machine for all the same software
* Earily revert version

A diagram of a puppet person

Description automatically generated

Containerizations

* A light version of VM
* A diagram of a docker image

  Description automatically generated
* Same images are used to run on many machines so that the code can work on all machine

Continuous monitoring

* Monitor network, software, entire infrastructure
* A diagram of data warehouse

  Description automatically generated