DevOps; class 1

Arthur Mauvezin

What is DevOps?

Waterfall model :

Steps to deliver:

* Design (agile)
* Code (agile)
* Test : unit or functional (agile)
* Release (sometimes agile)
* Deploy
* Operate
* Feedback

Agile with scrum :

It is based on incremental iterations.

It is a framework to work all together.

* Design/code/test on repeat, then Deploy.

⇒ at each step we deliver something new to the client; he can give you feedback

3 roles :

* Product owner : responsible for working with the team, is the gateway between the team and the external world
* Development team : all developers, testers,
* Scrum master : promotes scrum among team

Scrum limitations :

* Only the dev team is agile.

Devop steps : operate step added between the design/code/test step

Calms framework :

* Culture
* Automation
* Lean : removing something not necery
* Measurement
* Sharing

Cross team mindset

Better deliverable from input team

Better deliverable for output team

Automation & continuous improvement

Talk and put self into others place

Develop to the target

Shorten time to market

Paradagims :

Continuous integration

Continuous delivery

Continuous deployment

Continuous operations

Most known VCS : git, Linus Torvalds <3

Team practices :

* Pull request
* Code review
* Pair programming
* Coding dojo : meet up to optimal solutions so can move on
* Technical breakfast : present feedback, share and deepen knowledge
* TDD : Test Driven Dev
* BDD : Behavior Driven Dev
* Feature flipping
* Test A/B

CI : Build

Download dependencies

Build application

CI : Tests

Test all you can automatically

Unit testing

Functional, end to end tests

Load test

Security tests

Not optional

Reduce time to validate deliverable

Strengthen confidence

Value creation instead of debugging apps

Code analysis

Skills / Tools :

Continuous integration server : Jenkins, Gitlab,...

Scripting : python, bash, ruby

Building : maven, npm

Dependency management

Unit test : Junit

Integration test : postman

Code analysis : sonar

A release is delivered in an environment

Deliverable traceability

* Issue / ticket

Versioning problematic

Test rules:

Test validation must be perfect

Test maturation must be sufficient

Skills / Tools

Deployment / Orchestrator : puppet, docker

Load balancer : nginx

Metric collection

Logs centralization

Infrastructure as code

Skills / Tools

Monitoring : Centreon, Prometheus, influxdB, grafana

Lofs centralisation : elastic search, logstash

Infra as code : terraform, puppet, ansible

Skills

Be curious

Adapt quickly

Gather information