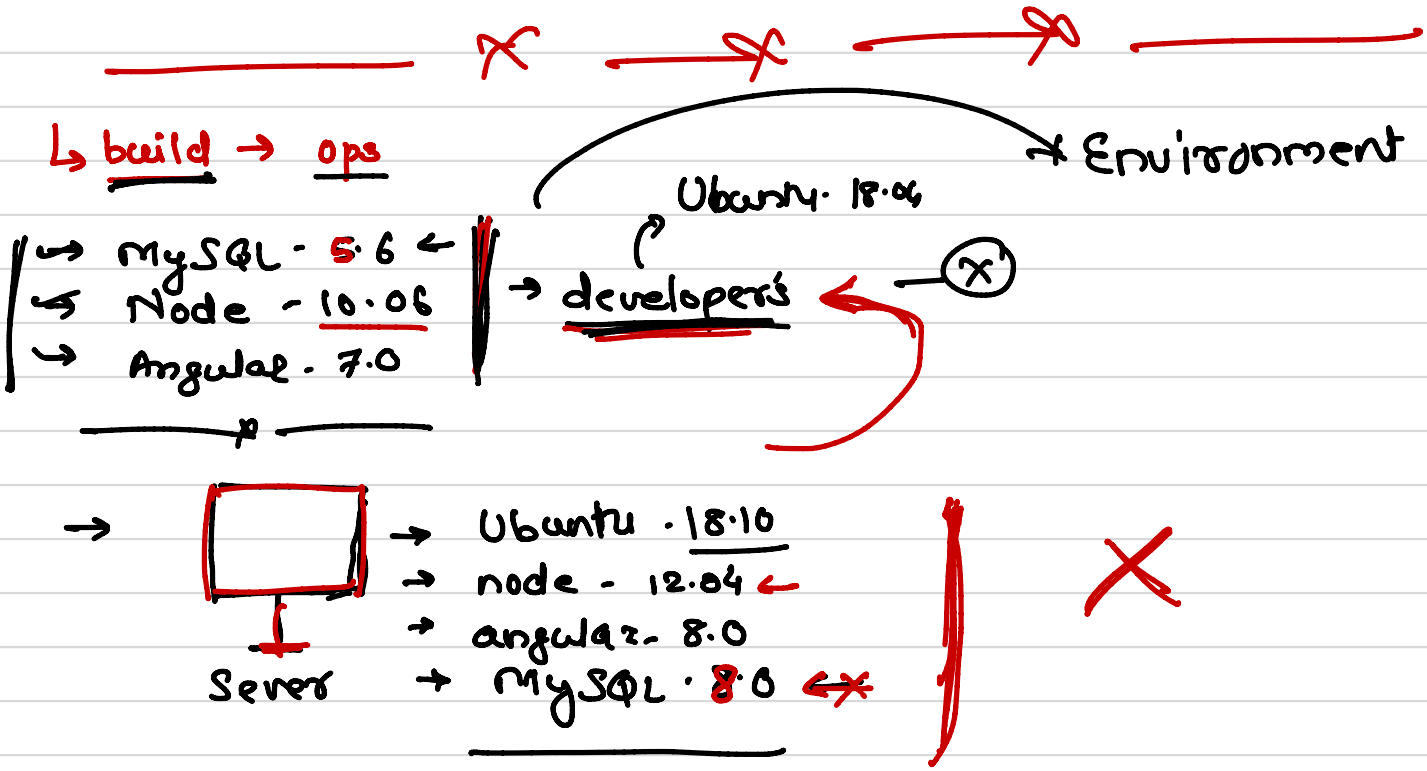
Developer + Operations

DevOps

1. Requirement gathering
2. Requirement Analysis
3. Design
4. Coding ← developer [software stack]
   ↳ containerization
5. Testing
6. Deployment ← operations, monitoring
7. Maintenance



DevOps (Automation)

1. Node . 10.04
2. MySQL . 5.6

Dockerfile

↳ container → VM

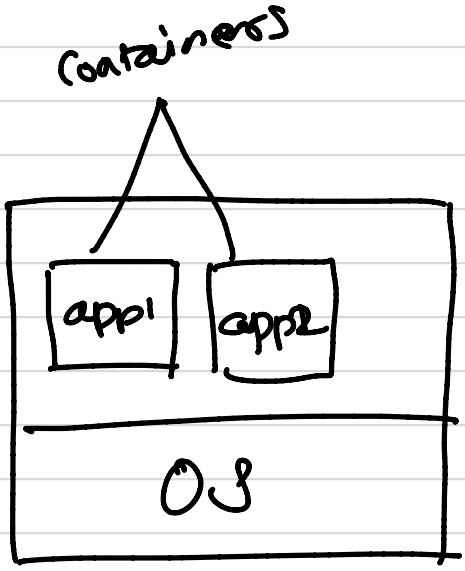
MySQL 5.6
Node. 10.04
⋮

Operations

→ Dockerfile

↳ Container (VM)

MySQL 5.6
Node . 10.04
⋮



Overview → Not a tool / software / application
→ a process / reference / guideline

- DevOps is a combination of two words development and operations
- Promotes collaboration between Development and Operations Team to deploy code to production faster in an automated & repeatable way (Agile)
- DevOps helps to increases an organization's speed to deliver applications and services
- It allows organizations to serve their customers better and compete more strongly in the market
- Can be defined as an alignment of development and IT operations with better communication and collaboration

Why DevOps is Needed?

- Before DevOps, the development and operation team worked in complete isolation
- Testing and Deployment were isolated activities done after design-build. Hence they consumed more time than actual build cycles.
- Without using DevOps, team members are spending a large amount of their time in testing, deploying, and designing instead of building the project.
- Manual code deployment leads to human errors in production
- Coding & operation teams have their separate timelines and are not in synch causing further delays

Reasons to use DevOps

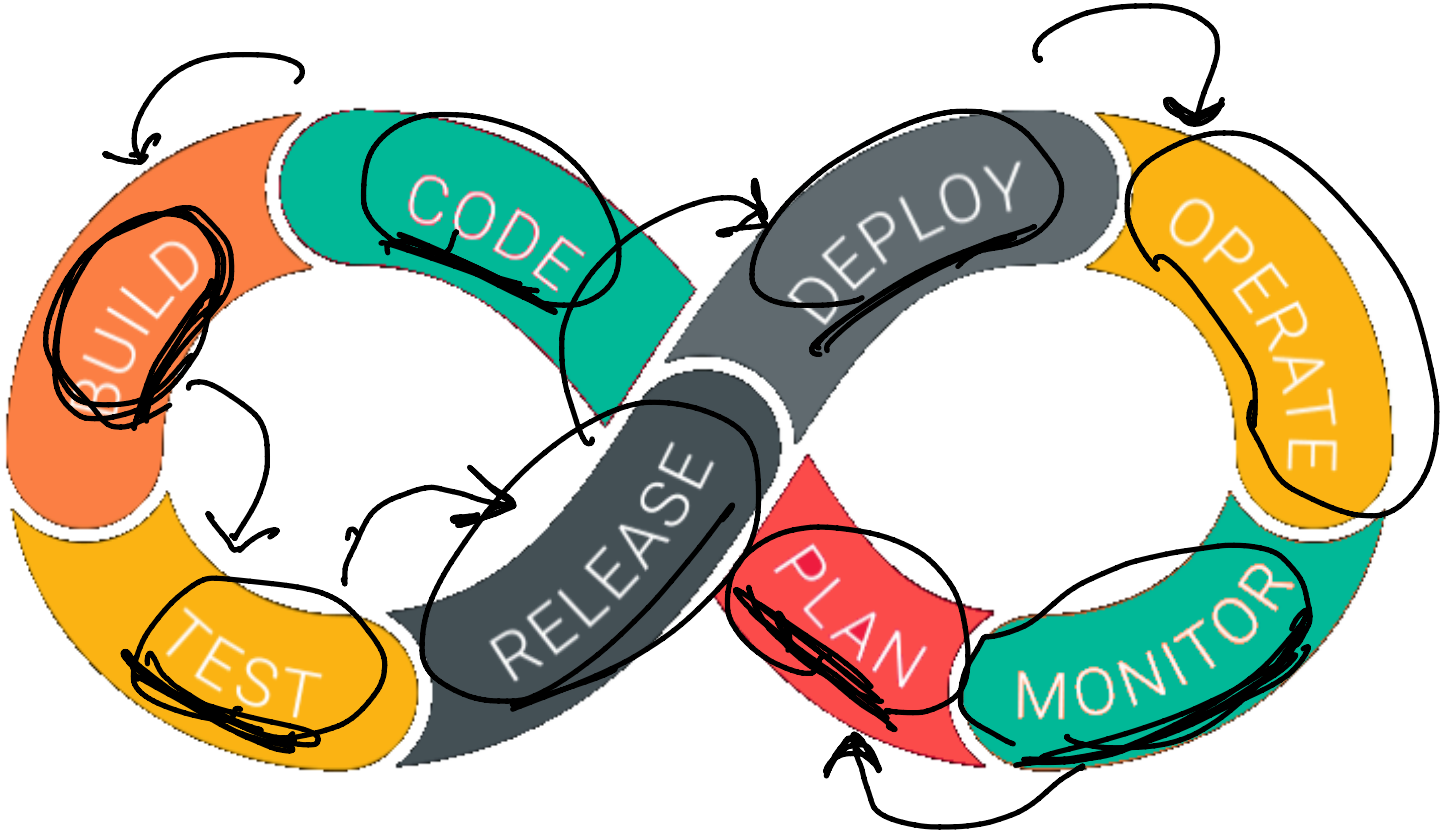
- **Predictability:** DevOps offers significantly lower failure rate of new releases
- **Reproducibility:** Version everything so that earlier version can be restored anytime
- **Maintainability:** Effortless process of recovery in the event of a new release crashing or disabling the current system
- **Time to market:** DevOps reduces the time to market up to 50% through streamlined software delivery. This is particularly the case for digital and mobile applications
- **Greater Quality:** DevOps helps the team to provide improved quality of application development as it incorporates infrastructure issues

Reasons to use DevOps

- **Reduced Risk:** DevOps incorporates security aspects in the software delivery lifecycle. It helps in reduction of defects across the lifecycle
- **Resiliency:** The Operational state of the software system is more stable, secure, and changes are auditable
- **Cost Efficiency:** DevOps offers cost efficiency in the software development process which is always an aspiration of IT companies' management
- **Breaks larger code base into small pieces:** DevOps is based on the agile programming method. Therefore, it allows breaking larger code bases into smaller and manageable chunks

DevOps Lifecycle

Agile → sprint



DevOps Lifecycle

CI / CT / CD

- **Development**
  - In this DevOps stage the development of software takes place constantly
  - The entire development process is separated into small development cycles
- **Testing**
  - Testing team use tools like Selenium to identify and fix bugs in the new piece of code
- **Integration**
  - In this stage, new functionality is integrated with the prevailing code, and testing takes place.
  - Continuous development is only possible due to continuous integration and testing

DevOps Lifecycle

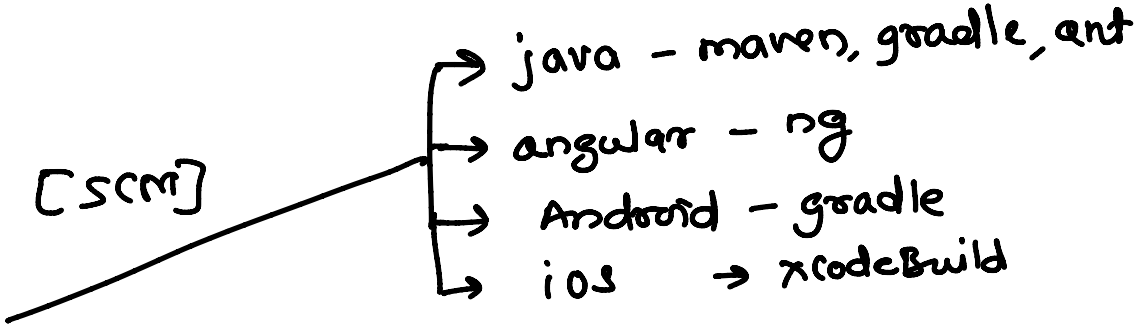
Agile → weeks
Iterative → months
Waterfall → years

- **Deployment**
  - In this phase, the deployment process takes place continuously
  - It is performed in such a manner that any changes made any time in the code, should not affect the functioning of high traffic website
- **Monitoring**
  - In this phase, operation team will take care of the inappropriate system behavior or bugs which are found in production

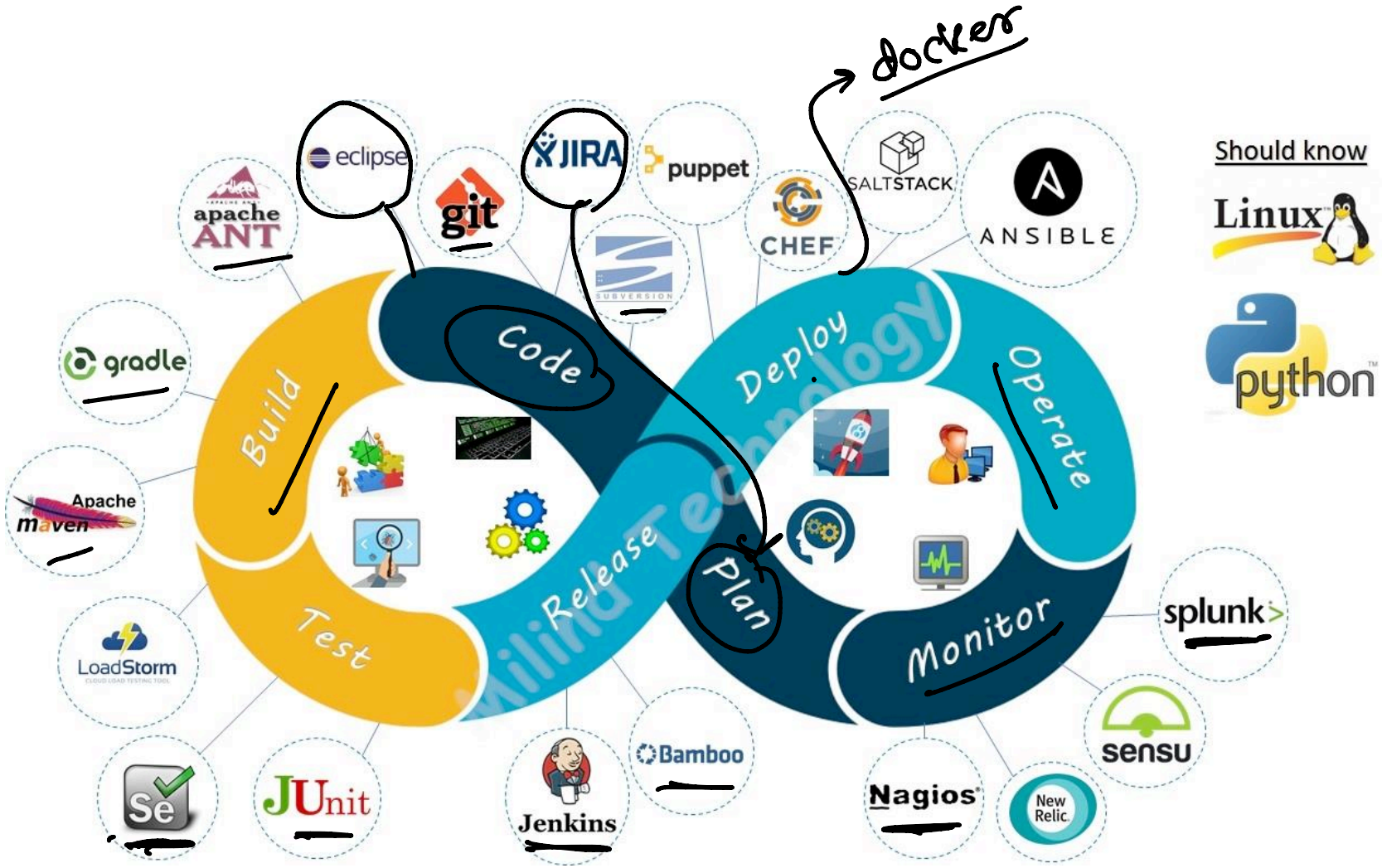
[automation]

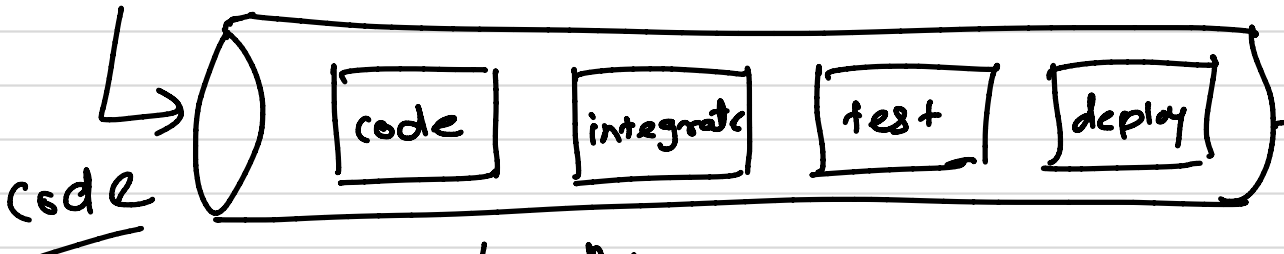
Tools used by DevOps engineers

- **Planning**: Jira
- **Code management**: Git, SVN
- **Building**: maven, gradle, ant
- **Testing**: selenium, TestNG
- **Integration**: Bamboo, Hudson, Jenkins [free]
- **Deploy**: Docker, Kubernetes, Vagrant
- **Configuration Management**: Puppet, Chef, Ansible
- **Monitor**: Slack, Splunk, Nagios



Tools used by DevOps engineers





code

jenkins

(automate)

well tested deployed code