EPITECH.
PARIS GRADUATE SCHOOL OF DIGITAL INNOVATION

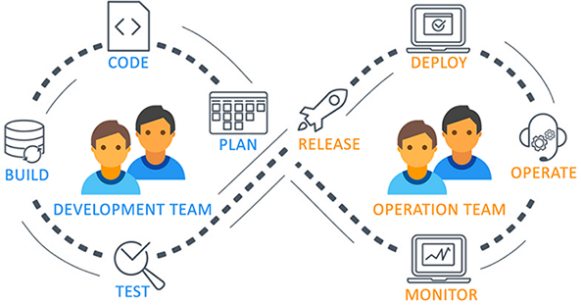
DevOps

Kick-off

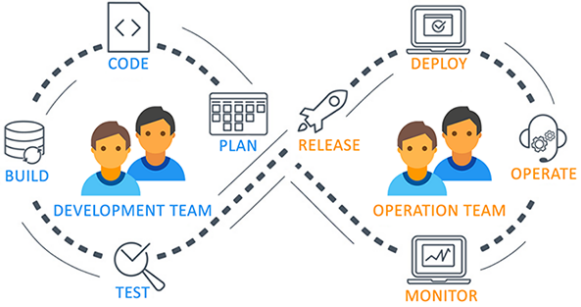
T7 - Network & Sys Admin

T-NSA-700

DevOps

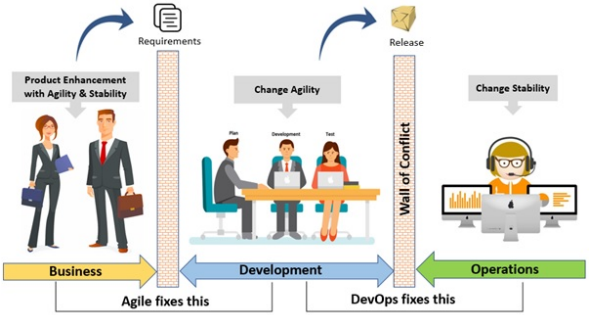

CODE
PLAN
BUILD
DEVELOPMENT TEAM
TEST
DEPLOY
RELEASE
OPERATE
OPERATION TEAM
MONITOR

DevOps

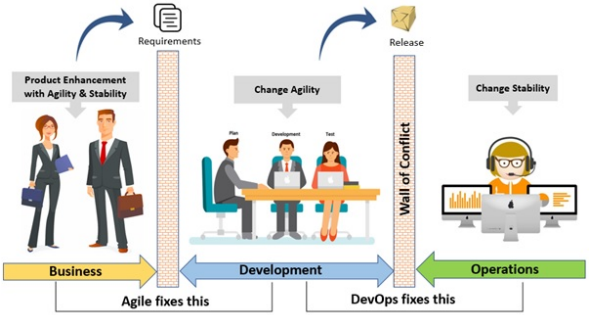


Developing software is much more than writing code.

Challenges

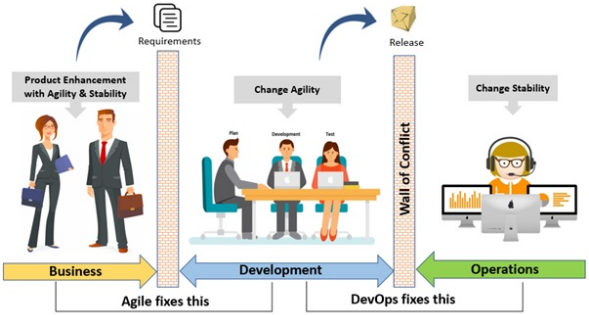

Requirements
Release
Product Enhancement with Agility & Stability
Change Agility
Change Stability
Plan
Development
Test
Wall of Conflict
Business
Development
Operations
Agile fixes this
DevOps fixes this

Challenges



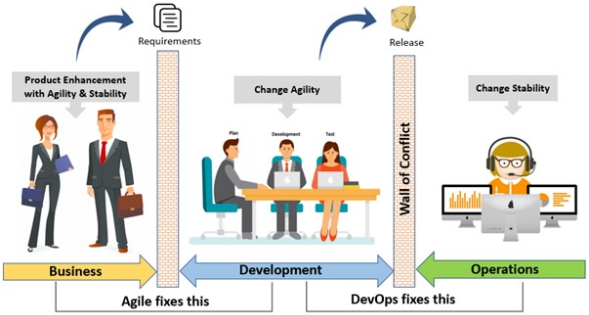
- Organisation

Challenges



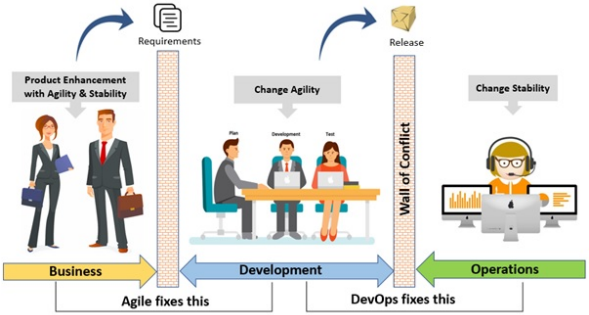
- Organisation
- Collaboration

Challenges



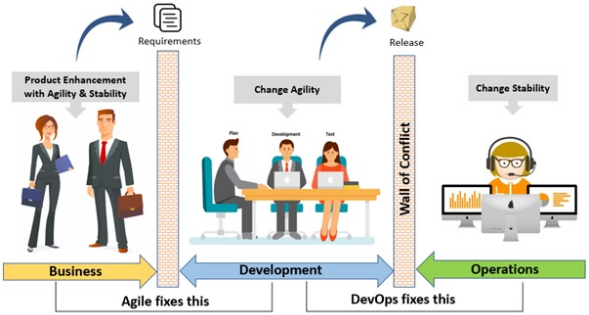
- Organisation
- Collaboration
- Deployment

Challenges



- Organisation
- Collaboration
- Deployment
- Compliance

Challenges



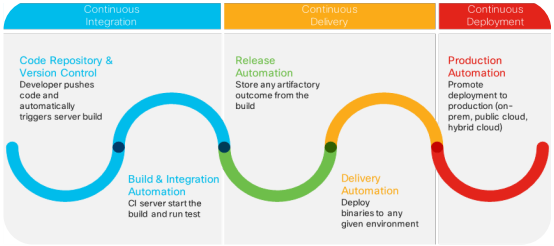
- Organisation
- Collaboration
- Deployment
- Compliance
- Performance

CI/CD

Continuous **I**ntegration / **C**ontinuous **D**elivery

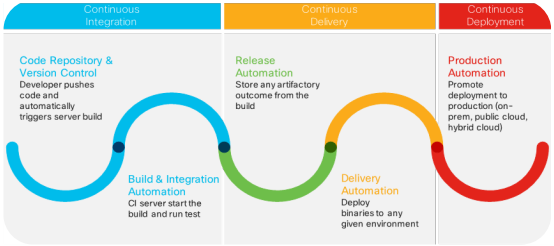
CI/CD

Continuous **I**ntegration / **C**ontinuous **D**elivery



CI/CD

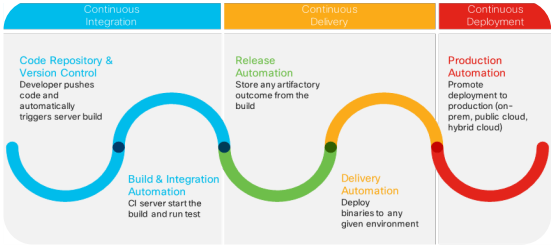
Continuous **I**ntegration / **C**ontinuous **D**elivery



- Manage project's lifecycle

CI/CD

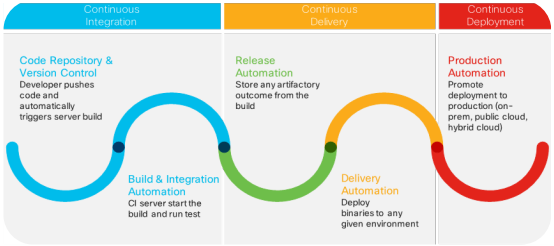
Continuous **I**ntegration / **C**ontinuous **D**elivery



- Manage project's lifecycle
- Test & validate phases

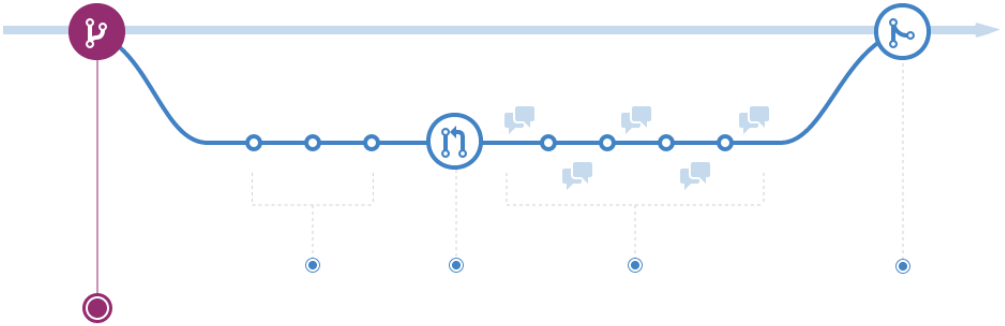
CI/CD

Continuous **I**ntegration / **C**ontinuous **D**elivery



- Manage project's lifecycle
- Test & validate phases
- Automate processes

Git branching: Github flow



Git branching: Github flow

Github flow: master branch is always deployable

Git branching: Github flow

Github flow: master branch is always deployable

- QA must be done on feature-branch, before merging to master

Git branching: Github flow

Github flow: master branch is always deployable

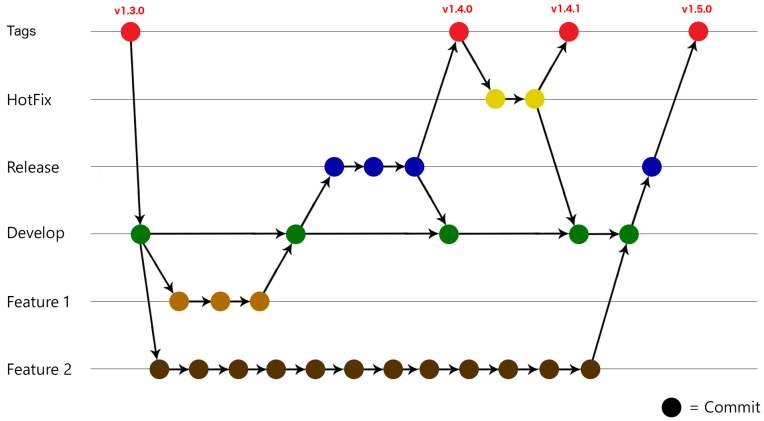
- QA must be done on feature-branch, before merging to master
- Merging to master can trigger a deployement automatically

Git branching: Github flow

Github flow: master branch is always deployable

- QA must be done on feature-branch, before merging to master
- Merging to master can trigger a deployement automatically
- Easy and efficient process

Git branching: Git flow



Git branching: Git flow

Git flow: only releases are deployed (git tags)

Git branching: Git flow

Git flow: only releases are deployed (git tags)

- Easier to maintain multiple software versions in production: 1.42.0, 2.0.0, 2.1.0…

Git branching: Git flow

Git flow: only releases are deployed (git tags)

- Easier to maintain multiple software versions in production: 1.42.0, 2.0.0, 2.1.0…
- Developers merge into a QA branch, then the QA team delivers to production

Git branching: Git flow

Git flow: only releases are deployed (git tags)

- Easier to maintain multiple software versions in production: 1.42.0, 2.0.0, 2.1.0…
- Developers merge into a QA branch, then the QA team delivers to production
- Hotfixes are applied to releases, then master branch (reverse order)

Git branching: Git flow

Git flow: only releases are deployed (git tags)

- Easier to maintain multiple software versions in production: 1.42.0, 2.0.0, 2.1.0…
- Developers merge into a QA branch, then the QA team delivers to production
- Hotfixes are applied to releases, then master branch (reverse order)
- Slow down development processes

Git branching: Git flow

Git flow: only releases are deployed (git tags)

- Easier to maintain multiple software versions in production: 1.42.0, 2.0.0, 2.1.0…
- Developers merge into a QA branch, then the QA team delivers to production
- Hotfixes are applied to releases, then master branch (reverse order)
- Slow down development processes
- Convenient for large teams

Tools

versionning

Tools

versionning
testing

Tools

versionning
testing
deployment

Tools

versionning
testing
deployment
virtualization

Tools

versionning
testing
deployment
virtualization

workflow

Tools

versionning
testing
deployment
virtualization

workflow
scenarios

Tools

versionning
testing
deployment
virtualization

workflow
scenarios
templates

Tools

versionning
testing
deployment
virtualization

workflow
scenarios
templates
community modules
or custom scripts

Tools

versionning
testing
deployment
virtualization

workflow
scenarios
templates
community modules
or custom scripts

Gitlab & Ansible

Gitlab & Ansible

- Organize your work

Gitlab & Ansible

- Organize your work
- Manage repositories

Gitlab & Ansible

- Organize your work
- Manage repositories
- Create roles

Gitlab & Ansible

- Organize your work
- Manage repositories
- Create roles
- Setup templates

Gitlab & Ansible

- Organize your work
- Manage repositories
- Create roles
- Setup templates
- Remotely control actions

Any questions

?