


# Un workflow de développement moderne

Thomas 'Nymous' Gaudin • 2019-01



# Nymous

- C'est moi 😊
- Président du Rézoléo 2015-2016
- Élève en EBM 2017-2018
- Consultant chez Extia, en mission DevOps à Adeo
-  <https://github.com/nymous>

# Plan

1. Quoi ? Pourquoi ?
2. Gestion de version
3. Tests
4. Automatisation
5. Documentation
6. Déploiement

# I. Quoi ? Pourquoi ?

## 1.1 Quoi ?

- Présentation des outils utiles à tout projet de dev moderne
- Intégration de ces outils dans une chaîne
- Projet squelette

## 1.2 Pourquoi ?

- Travail plus fluide
- Cycle *développement* => *mise en production* raccourci
  - => feedback rapide
- Moins peur de faire des erreurs
  - petits incréments faciles à corriger/annuler
  - un workflow qui réduit les erreurs
  - la MEP devient la routine
- Process automatisé
  - => moins d'erreurs humaines

## 2. Gestion de version

## 2.1 Pourquoi ?

Pourquoi faire de la gestion de version ?

- Retour en arrière
- Expérimentation possible
- Travail à plusieurs en parallèle



## 2.1 Outils

Outils de collaboration :

- GitHub
- Gitlab
- Bitbucket

## 2.2 Bonnes pratiques

- Branches (plusieurs workflows possibles)
  - tout dans master □
  - gitflow
  - master + branches de feature
- Branche master protégée
- Pull requests & code review

# Tests

## 3.1 Pourquoi ?

- S'assurer que le code fait ce qui est prévu
  - Tests unitaires
- S'assurer que des bugs corrigés ne reviennent pas
  - Tests de non-régression
- S'assurer de la qualité et sécurité du code
  - Outils d'analyse statique, linters
- Avoir confiance pendant un refactoring
  - Si tests vert => pas de changement de comportement

## 3.2 Outils

- Linter/Formatage :
  - ESLint
  - Prettier
- Tests :
  - Jest
  - Mocha
  - ava

## 3.3 Services

- Code coverage :
  - Codacy
  - Codecov
  - CodeClimate

# Automatisation

## 4.1 Pourquoi ?

- Réduire les erreurs humaines
- Forcer les bonnes pratiques
- Tâche répétitive => automatisisation => gain de temps



## 4.2 Outils

- Travis
- Circle CI
- Gitlab CI
- Codeship
- Drone CI

# Documentation

## 5.1 Pourquoi ?

- Information pour les autres
- Information pour soi
- Respect de sa propre API

## 5.2 Outils

- ApiDoc
- Sawgger

# Déploiement

## 6. Déploiement

- Capistrano
- Ansible, Puppet
- Docker (+ docker-compose, Swarm)
- Kubernetes
- OpenShift, Rancher
  - Traefik
  - Watchtower