# DevOps

## Gestion de développement collaboratif de logiciel

Thomas Ropars

thomas.ropars@univ-grenoble-alpes.fr

2019

# Usine de production de logiciel

Est ce qu'un éditeur de texte et un compilateur sont suffisants pour développer un gros code?

Qu'en est-il de la diffusion et de la gestion des utilisateurs?

# Usine de production de logiciel

Est ce qu'un éditeur de texte et un compilateur sont suffisants pour développer un gros code?

Qu'en est-il de la diffusion et de la gestion des utilisateurs?

Forge: Usine de production de logiciel

# Qu'est ce qu'une forge?

*Un système de gestion de développement collaboratif de logiciels. Il fournit une interface unifiée à une série de logiciels serveur et intègre plusieurs applications à code source ouvert. [wikipedia]*

## Des services pour:

- le développeur
- le manager

# Services fournis

## Les services principaux

# Services fournis

## Les services principaux

- Gestionnaire de version pour le code source
- Listes de diffusion et forums
- Wiki (gestion de la documentation)
- Service de téléchargement
- Système de gestion des incidents (tickets)
  - Suivi de rapports de bugs
  - Demandes d'évolutions

# Liste exhaustive de services

- Gestionnaire de version pour le code source
- Listes de diffusion et forums
- Wiki
- Service de téléchargement
- Système de gestion des incidents (tickets)
- Gestion des droits utilisateurs
- Authentification
- Rapports d'activité
- Gestionnaire de sondages
- Support d'intégration continue
- Support de revue de code
- Support de gestion de projet
- ...

# Les forges existantes

## Exemples de services de forge

On parle aussi de service web d'hébergement et de gestion de développement de logiciels.

- Launchpad.net
- Google code (closed)
- GNU Savannah
- SourceForge.net
- GitHub
- GitLab
- Benstalk
- Bitbucket
- AWS CodeCommit
- Gogs/Gitea
- ...

Voir aussi `https://en.wikipedia.org/wiki/Comparison_of_source_code_hosting_facilities`

# GitHub

- Inclus tous les outils principaux attendus pour une forge
  - Git utilisé comme gestionnaire de versions
- Ajoute un aspect réseau social
  - Possibilité de suivre des personnes/des projets
  - Notion de graphe entre les *forks* d'un projet
- Modèle:
  - S'inscrire permet de créer gratuitement des dépôts publics
  - Compte payant pour créer des dépôts privés
  - Les entreprises peuvent aussi payer pour héberger en interne une instance de la plate-forme
    - Base du modèle économique de GitHub

# D'autres solutions existent

- Gitlab
  - Open Source
  - Chacun peut héberger son instance du service
  - Intégration de mécanismes d'intégration et de déploiement continue
- AWS CodeCommit
  - Hébergé dans le Cloud Amazon: Haute disponibilité et passage à l'échelle
- Beanstalk
  - Revue de code
  - Déploiment du code sur différentes plateformes
- Gogs/Gitea
  - Service à la GitHub/GitLab mais plus léger

# Les fichiers à inclure dans un projet

## Le fichier README

- A la racine du projet
- Inclus un ensemble d'informations:

# Les fichiers à inclure dans un projet

## Le fichier README

- A la racine du projet
- Inclus un ensemble d'informations:
  - Auteurs et remerciements
  - Informations sur Copyright et Licence
  - Instructions pour la configuration/l'installation
  - Un manifest (liste des fichiers)
  - Bugs connus
  - Aide au dépannage (Troubleshooting)
  - ...

Sur Github, le fichier README est automatiquement convertit en HTML pour être affiché sur la page d'accueil du projet

- Format Markdown (README.md) pour une conversion simple en HTML

# Les fichiers à inclure dans un projet

## Ensemble de fichiers "readme"

- README: informations générales
- AUTHORS: crédits
- THANKS: remerciements
- CHANGELOG: journal détaillé des changements, destiné aux développeurs
- NEWS: journal simplifié des changements, destiné aux utilisateurs
- INSTALL: instructions pour l'installation
- COPYING / LICENSE: Copyright/ Licence
- BUGS: bugs connus et instructions pour en rapporter de nouveaux

# Ameliorer la qualité du logiciel avec des merge requests

## Dénomination

- Merge requests est le nom utilisé par Gitlab
- Pull requests est le nom utilisé par Github

## Principe

- Soumettre un ensemble de modifications à intégrer à un projet
- Les modifications vont devoir être approuvées par les responsables de ce projet
- Avant d'être acceptée les développeurs du projet peuvent:
  - Relire le code
  - Faire des commentaires sur des modifications proposées
  - Demander des modifications du code proposé
  - Exécuter un ensemble de tests

# Merge requests: les étapes principales

## Le contributeur

- Créer une branche de travail et y ajouter ces modifications
- Publier une *merge request* incluant les modifications
- Répondre aux commentaires et ajouter des commits si besoin

## L'ensemble des développeurs

- Voir l'ensemble des modifications associées à la requête
- Faire une revue de code et ajouter des commentaires
- Approuver la requête

# Merge requests

## Pour aller plus loin

- https://blog.zenika.com/2017/01/24/pull-request-demystifie/
- https://docs.gitlab.com/ee/user/project/merge_requests/
- https://help.github.com/en/articles/about-pull-requests

# Références


- Notes de cours de D. Donsez