



**POLYTECHNIQUE  
MONTRÉAL**

UNIVERSITÉ  
D'INGÉNIERIE

LOG1000

# Ingénierie logicielle Processus de développement d'un projet logiciel open source

## Travail pratique #3

Polytechnique Montréal Remis le 10 février 2019

**Trimestre :** Hiver 2019

equipe= 32

Équipier1: *Sami Belaidi - 1900787*

Équipier2: *Lucas Simoneau – 1885195*

## 4 Partie 1

### 4.1 Révision technique de code source

1. Gerrit est un outil web qui permet de passer en revue les modifications sur le code source grâce à un navigateur web. Ce logiciel est utilisé avec Git afin d'apporter des modifications au code source. Ces modifications peuvent être lues, approuvées ou rejetées.
2. Dans un contexte de contribution à un projet, un patch est une modification apportée par une tierce personne à ce projet. Cette modification pourra être revue et appliquée par les membres de l'équipe
3. La section open sont des modifications pas encore revues par les propriétaires du projet la section Merged montre les modifications qui ont été acceptées, et la section Merged montre les modifications qui ont été abandonnées.
4. L'auteur de cette révision est Jörg Mayer .
5. Un seul fichier a été modifié, soit packet-mint.c.
6. Cette révision renomme "mint type" en "mint port", et améliore la liste de ports.
7. +2 Anders Broman cela signifie que le code a été revu deux fois par Anders Broman.
8. Répondez aux questions 4 à 7 inclusivement, mais cette fois avec la révision `Id48cd9b876b4d272eb6dbdadf8a0859190cc0ce8`.
  - 8.1. L'auteur de cette révision est Peter Wu
  - 8.2. Trois fichiers ont été modifiés qui sont : `preferences_dialog.cpp`, `rsa_keys_frame.h`, `rsa_keys_frame.cpp`.
  - 8.3. Cette revision charge la liste des URI des clés, supprime les clés connues, demande un code Pin et ajoute les informations valides. Appliquez cette liste en appuyant sur OK dans la boîte de dialogue des préférences ou annulez les modifications en appuyant sur Annuler.
  - 8.4. +2 Peter Wu cela signifie que Peter Wu a revu le code deux fois.

## 4.2 Gestion des bogues

9. Nick Calus est la personne qui a trouvé ce bug. Il s'agit d'un humain.
10. Le niveau d'importance est bas normal ( "low normal")
11. Le niveau d'importance d'un bug est déterminé par des personnes qui votent pour celle-ci.
12. Non, il ne risque pas de compromettre le fonctionnement du logiciel, car le bug est seulement le manque de documentation pour un paramètre.
13. Deux personnes ont émis un commentaire.
14. Oui , le bug est confirmé car Alexis La Goutte demande s'il peut push le fix sur gerrit.
15. Oui, au debut de la page c'est marqué "Status:RESOLVED FIXED".

## 4.3 Intégration continue

16. Un buildbot permet de compiler automatiquement un projet lorsque des changements sont effectués afin de vérifier son bon fonctionnement. Dans un projet open-source, vu que le code est souvent modifiée, on veut savoir si les changements apportés ne nuisent pas au bon fonctionnement de l'exécutable.
17. Il y a 6 builders pour ce projet:
  - Clang Code Analysis
  - Ubuntu 18.04 x64
  - Visual Studio Code Analysis
  - Windows Server 2016 x64
  - Windows Server 2016 x86
  - macOS 10.14 x64
18. Le rouge : le test a echoue.  
Le vert: le test a ete reussi  
Le orange: le test presente des warning

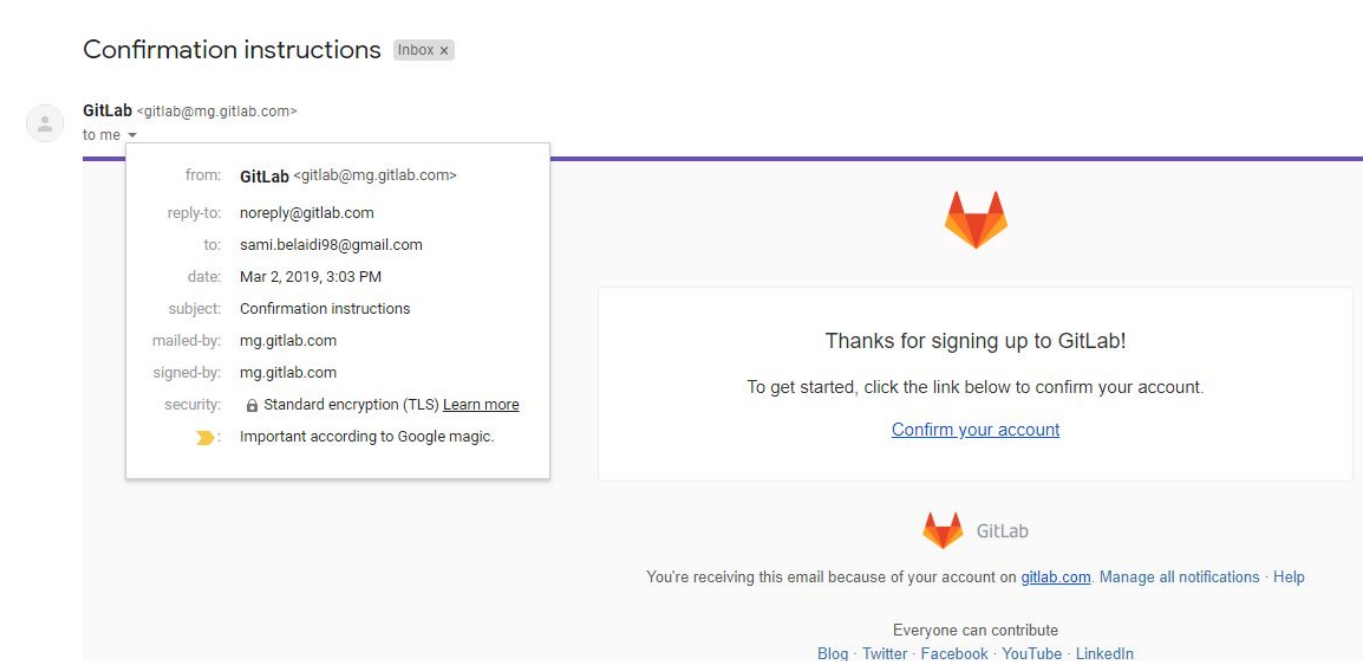
19. Voici un build qui contient un avertissement.



20. En regardant le message de build, le warning est le suivant: "ICE86: ICE86 - Post warning for the use of AdminUser instead of Privileged property in conditions." Cela veut dire que dans les conditions, AdminUser a été utilisé à la place de Privileged property.
21. Le numéro technique est le suivant: ICE86. L'auteur est le builder msbuild.

## 5 Partie 2

### 5.1 Prérequis





### 5.2 Développement






22. Log1000\_h19 correspond au nom d'un projet sur GitLab.

23. Fork fait une copie du projet. Cela permet de pouvoir modifier le projet sans affecter le projet source.

24.

You pushed to Equipe32 at Sami Belaidi / Log1000\_h19 1 hour ago [Create merge request](#)



 **Log1000\_h19**   
Project ID: 11113438





 Add license  10 Commits  2 Branches  0 Tags  123 KB Files



Projet pour le TP3 de LOG1000






Forked from [soumaya Medini / Log1000\\_h19](#)



---


master  log1000\_h19 / 

 History  Find file  Web IDE 

 Add TP3 Files  
soumaya Medini authored 3 weeks ago 9d599e83 

 Add README  Add CHANGELOG  Add CONTRIBUTING  Enable Auto DevOps  Add Kubernetes cluster

 Set up CI/CD  Security Dashboard











Name	Last commit	Last update
 TP3-H19	Add TP3 Files	3 weeks ago

Sami Belaidi > Log1000\_h19 > Repository > Branches

Overview **Active** Stale All  [Delete merged branches](#) [New branch](#)

Protected branches can be managed in [project settings](#).

Active branches

 master <b>default</b> <b>protected</b>		
 9d599e83 · Add TP3 Files · 3 weeks ago		
 Equipe32		
 9d599e83 · Add TP3 Files · 3 weeks ago	0   0	<a href="#">Merge request</a> <a href="#">Compare</a>  

25. Il faut utiliser git clone ainsi que git checkout Equipe32

```

gigl@fedora-gigl TP3-H19]$ make
g++ -o main.o -c main.cpp
g++ -o moteur.o -c moteur.cpp
g++ -o piston.o -c piston.cpp
g++ -o voiture.o -c voiture.cpp
g++ -o mecanicienne.o -c mecanicienne.cpp
g++ -o moteur_putput.o -c moteur_putput.cpp
g++ -o moteur_beepbeep.o -c moteur_beepbeep.cpp
g++ -o garage main.o moteur.o piston.o voiture.o mecanicienne.o moteur_putput.o moteur_beepbeep.o
/garage
Voitures :
Voiture A : Moteur #1123 d'une puissance de 2 cv
Voiture B : Moteur #2558 d'une puissance de 24 cv
Voiture C : pas de moteur
Moteurs :
Moteur #1123 : 2 pistons et d'une puissance de 2 cv
Moteur #2001 : 6 pistons et d'une puissance de 18 cv
Moteur #2558 : 8 pistons et d'une puissance de 24 cv
Mecaniciennes :
Mecanicienne #100 a les assignations suivantes :
Voiture A
Voiture B
Mecanicienne #161 a les assignations suivantes :
Voiture B
Mecanicienne #553 a les assignations suivantes :
Voiture B
Voiture C
gigl@fedora-gigl TP3-H19]$ S

```

26. La commande a utiliser pour ajouter les modifications au staging local est "git add .".
27. Un merge request est une demande au propriétaire du projet de faire un "merge" de deux branches. La première est celle du projet source, et la deuxième est celle de la personne qui a modifié le code.
28. L'utilité d'un merge request a deux grande utilités. Premièrement, il permet à des personnes de pouvoir apporter des modifications à des projets qu'ils ne possèdent pas. Deuxièmement, cela permet également de protéger le code source, afin que ce dernier ne subisse pas des modifications non voulues.
29. La branche source est SamiBelaidi/log1000\_h19 et la branche de destination est somedPoly/log1000\_h19

## 5.2.1 Révision de code

30. Equipe 79: [https://gitlab.com/somedPoly/log1000\\_h19/merge\\_requests/37](https://gitlab.com/somedPoly/log1000_h19/merge_requests/37)
31. Le commits associée est "ajoute moteur.o dans le makefile" (avec la référence :8017b5933b6116d7d6e1d67c5630aa2c8c92d47b).
32. Le fichier qui a été modifié est le makefile.

- 33. Oui, les modifications demandées sont correctes. Les modifications attendues ont été faites.
- 34. Oui, les objets et l'exécutable du projet ont bien été créés.
- 35. Le score attribué à l'équipe 79 est de 10/10 vu qu'ils ont respecté toutes les consignes.