Références

- https://www.gnu.org/ (système d'exploitation et mouvement du logiciel libre)
- https://producingoss.com (livre sur le développement open source)
- https://openhatch.org (archive: organisation aidant à contribuer dans l'open source)
- http://teachingopensource.org/ (actualité / blog posts sur l'enseignement open source)
- http://open-advice.org/ (conseils de développeurs open source)
- http://oss-watch.ac.uk (conseil sur l'utilisation, le développement et les licences open source)

Ce chapitre va se focaliser sur des exemples de bonnes attitudes sociale à avoir pour qu'un projet perdure dans le temps !

- **Capacité opérationnelle** : capacité du projet à incorporer de nouvelles contributions de code, de nouveaux développeurs, ...
- **Capacité de survie** : capacité à survivre au départ de membres fondateurs, ...

Exemple d'échec à cause d'un manque dans ces capacités : incapacité à gérer la croissance d'un projet; à gérer le départ d'individus charismatiques, ...

Ce chapitre devrait apporter des réponses aux questions suivantes : Comment intégrer des contributeurs réguliers du projet dans votre flux de travail ? Comment / quand donner à quelqu'un un accès de *commit* ? Comment résoudre les débats de la communauté ? ...

IV – Infrastructure sociale

IV.1 - Modèles de gouvernances

IV.2 – Consensus et vote

IV.3 – Commiters

IV.4 – Autres rôles

IV.1 - Modèles de gouvernances

Qu'est ce qu'un modèle de gouvernance ?

Qu'est ce qu'un modèle de gouvernance ?

C’est une description où l’on retrouve :

- une description du projet et de la manière dont il est géré;
- les **rôles** et **responsabilités** que peuvent endosser les participants du projet
  💭 Les rôles sont souvent des termes larges pour éviter l’auto censure (*user* , *contributor*, *committer*, ...)
- des **règles de base de la participation** au projet;
- les processus de prise de décision;
  💭 Atteste qu'une décision prise est *valide* et assure aux contributeur que leurs efforts seront traités équitablement;
- les **processus de communication** et **de partage** au sein de l'équipe et de la communauté du projet.

=> Modèle qui empêche un projet open source de sombrer dans le chaos !

Pourquoi est-il important de définir un modèle de gouvernance ?

Pourquoi est-il important de définir un modèle de gouvernance ?

Il existe beaucoup de modèles de gouvernance, il est dont important de communiquer clairement ses politiques et stratégies !

Pour les développeurs potentiels : comprendre comment ils peuvent s'engager dans le projet et ce que l'on attend d'eux

Pour les utilisateurs potentiels : description des processus de contrôle de la qualité du projet
💭 C'est un indicateur de viabilité du projet

⚠️ Puisqu'un seul collaborateur externe peut avoir un effet majeur sur la durabilité d'un projet, autant ne pas risquer de le perdre !

Quelles sont les barrières au fait qu'un projet adopte un modèle de gouvernance ?

Quelles sont les barrières au fait qu'un projet adopte un modèle de gouvernance ?

Malgré l'importance de définir un modèle de gouvernance dès le départ, de nombreux projet ne le font pas :

1. Processus vue comme un exercice bureaucratique inutile (*"paperasse"*) car souvent présumé comme complexe alors que c'est l'inverse :
   - **L'objectif est d'avoir un modèle simple et efficace** !
     Par exemple: processus de décision simple pour ne pas ralentir le projet
   - Pas besoin d'avoir des règles pour gérer toutes les situations
   - Doit fournir un moyen léger de guider la gestion du projet de manière claire et transparente

⚠️ Un modèle de gouvernance doit faciliter, et non compliquer, le processus de création et d'évaluation des contributions des tiers !

Quelles sont les barrières au fait qu'un projet adopte un modèle de gouvernance ?

2. Crainte de la perte d'orientation du projet, i.e. crainte de contraindre le projet à s'adapter à un environnement changeant

💭 C'est un phénomène fréquent dans les projets mal gouvernés car manque d'agilité, i.e. problème lié à l'absence d'un modèle != l'existence de celui-ci

En réalité, le modèle apporte l'effet inverse :

- **Apporte de l'agilité** : définit comment les nouveaux contributeurs peuvent amener le projet dans des directions inattendues sans interférer avec ses objectifs fondamentaux
- Fournit à la communauté un mécanisme pour définir l'orientation qu'elle souhaite donner au projet ET **garanti que l'équipe centrale du projet ne perde pas le contrôle**

Quelles sont les barrières au fait qu'un projet adopte un modèle de gouvernance ?

3. La prise de contrôle du projet par des tiers

💭 Mais encore une fois c'est totalement l'effet inverse), cela garantit que le contrôle reste précisément là où la direction du projet le souhaite, que ce soit la prise de décision centralisée ou distribuée !

⚠️ Lors du choix du niveau de contrôle, il est important de prendre en compte comment les efforts de l'équipe seront soutenus :

- un produit rentable : maintient du contrôle et encouragement des collaborations
- un produit peu rentable : promouvoir l'utilisation la plus large possible

Quelles sont les barrières au fait qu'un projet adopte un modèle de gouvernance ?

4. Projet jugé trop jeune / petit pour attirer des utilisateurs / développeurs actifs

⚠️ Ce n'est jamais le cas ET son absence réduit drastiquement les chances de contributions

Exemple de raisons liées à cette réduction : ne pas savoir comment contribuer ou ce qu'il adviendra de la contribution; le projet communique l'impression de ne pas vouloir s'engager avec des tiers; ...

Idées courantes (**fausses**):

- projet trop jeune alors qu'une contribution qui améliore la qualité/maniabilité du logiciel est importante à n'importe quel moment !
- projet trop petit alors que c'est un manque de compréhension de l'open source (rarement plus d'une poignée de développeurs actifs à un moment donné et une communauté bien gérée est autosuffisante)

Connaissez-vous des modèles de gouvernance ?

Connaissez-vous des modèles de gouvernance ?

Spectre des possibles modèles allant du contrôle centralisé au contrôle distribué :

- le **dictateur bienveillant** : centralisé autours d'une personne / organisation;
- la **méritocratie** : contrôle distribué accordé en reconnaissance des contributions;
- la **contribution libre** : totalement distribué en accordant du crédit à toute contribution.

Des **différences structurelles** apparentes **MAIS** elles **suivent les principes du libre / open source** : le partage du code et l'encouragement de chacun à contribuer à la communauté !

Connaissez-vous des modèles de gouvernance ?

💭 Il arrive souvent qu'un modèle de gouvernance évolue à mesure que le projet mûrit. En effet, un projet peut, par exemple, passer d'une dictature à une démocratie.

- Dictateur qui se retire du projet
- Dictateur souhaitant répartir la responsabilité des décisions

💭 La transition est rarement inversée. Si on considère la transition d'une dictature à une méritocratie cela se traduirait par :

- N-1 personnes d'accord pour renoncer à leurs influence individuelle
- N-1 personnes ayant élue le dictateur mais pouvant aussi facilement le destituer

Le modèle de gouvernance peut être exposé dans un fichier dédié du code source, par exemple https://github.com/nodejs/node/blob/master/GOVERNANCE.md

Quelles sont vos expériences de modèle de gouvernance ?

- Projet en entreprise ?
- Projet personnel ?
- Projet universitaire (Nuit de l'info / ...) ?

Peronnelement:

- MAESTRO : à mi chemin entre la dictature et la méritocratie
- HM : dictature
- PaDaWAn / Elo / O2 / Coupe : méritocratie

Dictateur bienveillant

Terme un peu rude mais proche de la réalité : une autorité qui a le dernier mot sur toutes les décisions importantes du projet.

⚠️ **Utilisation de cette autorité peu fréquente** car :

- difficile d'avoir suffisamment d'expertise partout;
- contributeurs souhaitent avoir une influence sur la direction du projet;
- **risque de Fork** !

💭 Laisser les débats s'arranger via discussion et expérimentation ! **Intervenir lorsqu'un consensus ne peut être atteint** pour clôturer le débat et aller de l'avant

Exemples: Linus Torvalds (Noyau Linux), Guido van Rossum (Python) BDFL (Benevolent dictator for life), Matz (Ruby), Bram Moolenaar (Vim), ...

Quelles caractéristique peut-on attendre d’un dictateur bienveillant ?

Quelles caractéristique peut-on attendre d'un dictateur bienveillant ?

- **Conscience de son influence** dans le projet (mesurer son discours)
  ⚠️ Ne pas donner l'impression qu'il n'est pas utile de s'opposer
- Capacité à **reconnaître ses erreurs** : accepter que l'on peut avoir de mauvaises idées (sinon elle resteront longtemps dans le projet)
- **Capacité technique** : pouvoir travailler sur le code; pouvoir participer à une discussion sur les changements envisagés; pas besoin d'avoir les compétences les plus pointues !
- [Optionnel] **Fondateur du projet** (ancienneté automatique)

💭 Si aucun candidat n'est évident pour être le dictateur, alors une autre organisation devrait être envisagée !

Globalement

Le rôle du dictateur bienveillant relève moins de la dictature que de la diplomatie

Clefs du succès :

- s'assurer que les bonnes personnes ont une influence sur le projet
- s'assurer que la communauté se rallie à la vision du dictateur

Rôle : s'assurer que les committers prennent les bonnes décisions au nom du projet

⚠️ Tant que tout est en accord avec la stratégie du projet : ne pas utiliser son pouvoir !

Exemple de document décrivant la gouvernance : les grandes lignes

Ce qu'il faudrait retrouver dans le document :

- Vue d'ensemble
- Rôles et responsabilités (varient selon le modèle de gouvernance)
- Processus de contribution & processus de décision

Section vue d'ensemble

- Projet dirigé par un dictateur bienveillant et géré par la communauté
- **Ce qui est attendu de la communauté** : contribuer activevement à la maintenance quotidienne du projet
  💭 Mission : guider les décisions du dictateur bienveillant en s'engageant et contribuant
- **Ce qui est attendu du dictateur bienveillant** : tracer la ligne stratégique générale du projet
  💭 Mission : résoudre les conflits au sein de la communauté et veiller à la progression coordonnée du projet

Section rôles et responsabilités

Utilisateur : membre de la communauté qui a besoin du projet

- Pas d'exigences spécifiques (excepté la compréhension du projet)
- Important pour un projet : sans eux le projet n'aurait pas de raison d'être
- Actions courantes des utilisateurs :
  - prêcher les louanges du projet;
  - donner du retour sur les forces / faiblesses vu d'un nouvel utilisateur;
  - apporter un soutient morale;
  - apporter un soutient financier.

Section rôles et responsabilités

Contributeur : membre de la communauté qui ne souhaite pas devenir commiter ou n'en a pas encore eu l'occasion.

- Généralement pas le pouvoir d'apporter des modifications directes : Workflow où le contributeur envoie un patch par mail / PR / MR
  💭 Dans l'historique Git il y aura alors un commiter != auteur

```
git log -p -2 –-pretty=’Commiter = %cn, Author =%an ‘
```

- Pas de compétence spécifiques attendues
- Réaliser des actions bénéfiques au projet : apporter du support aux nouveaux utilisateurs; signaler / corriger des bogues; rédiger de la documentation; programmer (tests, nouvelles fonctionnalités, ...); ...

Section rôles et responsabilités

Commiter : contributeur ayant apporté des contributions précieuses et sur lesquel on compte pour écrire du code directement dans le référentiel / filtrer les contributions

- N'est pas libre de faire ce qu'il veut : son travail continue d'être examiné par la communauté avant d'être accepté dans une version officielle
- Souvent concentré sur un aspect spécifique du projet
- Apporte un grand niveau de connaissances et d'expertises
- Position associée aux membres influents de la communauté
- Pas d'autorité sur la direction du projet MAIS a l'oreille du dictateur

💭 Différence entre commiter et contributeur : moment où l'approbation d'une contribution est demandée à la communauté (commiter après plutôt qu'avant)

Section processus de contribution et de décision

Comment sont gérés les processus de contribution et de décision dans une dictature bienveillante ?

Section processus de contribution et de décision

Comment sont gérés les processus de contribution et de décision dans une dictature bienveillante ?

- Processus de contribution classique, i.e. n'importe qui peut contribuer au projet puisqu'il y a une myriade de tâches : donner du retour utilisateur; tester et reporter ou corriger des bogues; traduire; écrire ou mettre à jour de la documentation, ...
- Processus de décision classique, i.e. basé sur le consensus:
  - ne nécessite pas forcément un processus formel;
  - en cas de conflit, le dictateur tranche.

Méritocratie

Terme explicite : **le gouvernement par le mérite**

💭 Mot d'ordre : *"C'est ceux qui font qui décident"*

Modèle qui peut sembler céder le contrôle aux membres de la communauté en réponse à leurs contributions au projet mais ce n'est pas le cas :

- le contrôle n'est pas donné sans considération;
- démocratie != méritocratie => tout le monde n'a pas un vote contraignant;
- capacité de vote gagnée au mérite : construit sur la disponibilité, l'implication, la réactivité et le travail bien fait !

Méritocratie

Une structure à plat : une fois le pouvoir de décision obtenu, l'autorité est la même pour tout le monde !

💭 Le modèle méritocratique permet de s'assurer que le pouvoir de décision est octroyer à ceux qui partagent la vision commune du projet et sont disposés à travailler vers cette vision.

Quelques exemples de méritoracratie:

- Apache : extension de la communauté des développeurs lorsque le groupe d'administration estime une personne https://www.apache.org/foundation/how-it-works/#meritocracy

Méritocratie

- Communauté Mozzila : autorité distribuée entre les bénévoles et employés de la communauté (dépend des capacités de contribution au projet) https://www.mozilla.org/en-US/about/governance/

- Debian : élection annuelle du leader et positions clefs au mérite https://debian-handbook.info/browse/hr-HR/stable/sect.debian-internals.html

Exemple de document décrivant la gouvernance : les grandes lignes

Ce qu'il faudrait retrouver dans le document :

- Vue d'ensemble
- Rôles et responsabilités (varient selon le modèle de gouvernance)
- Processus de contribution & processus de décision

Section vue d'ensemble

- Projet communautaire méritocratique et consensuel
- **Ce qui est attendu de la communauté** : contribuer activevement à la maintenance quotidienne du projet
  💭 Mission : contribuer à la conception du projet et participer aux processus de prise de décisions
- **Ce qui est attendu du comité de gestion de projet** : tracer la ligne stratégique générale du projet
  💭 Mission : participer à la planification stratégique et approuver les modifications apportées au modèle de gouvernance (e.g. l'ajout d'un membre)

Section rôles et responsabilités

- Utilisateurs, Contributeur (cf slides précédents)
- **Commiters** dans le contexte d'une **méritocratie**
  Les nouveaux commiters sont nommés par d'autres commiters MAIS doivent être validés par le comité de gestion :
  - la procédure de vote est privée afin de laisser le comité de gestion s'exprimer librement sur un candidat et éviter de froisser la personne
  - le résultat est rendu publiquement par la suite, le candidat peut demander des explications

⚠️ Le statut de commiter est un privilège et pas un droit, il doit être gagné et peut être retiré par le commité de gestion !

Section rôles et responsabilités

Commité de gestion du projet : identifié comme *propriétaire du projet*

- Eventail de responsabilités plus large que celui d'un commiter (bon déroulement du projet)
- Actions réalisés par les membres : examiner les contribution au code; participer à la planification stratégique du projet; valider les modifications apportées au modèle de gouvernance, ...
- Pas d'autorité significative sur les autres membres de la communauté sauf sur le choix des nouveaux commiters et la prise de décision lors de débat sans consensus communautaire
- Nouveaux membres par validation et invitation des membres du comité
- A accès à des éléments privés du projet (liste de diffusion & archives)

Section rôles et responsabilités

💭 Les éléments privés sont utilisés par le commité uniquement pour des questions sensibles, e.g. vote pour un nouveau commiter, questions juridiques, ... mais ne sont pas utilisés pour la gestion / planification du projet !

[Optionnel] Président du commité de gestion :

- **Personne unique**, élue par les membres du comité
- Poste conservé jusqu'à ce qu'il décide de se retirer / le comité vote pour le destitué (exemple: au moins 2/3 du comité souhaite la destitution)
- Pas d'autorité supplémentaire aux autres membres du comité sauf dans le cas où le commité n'arrive pas à un consensus, i.e. voix prépondérante
- Rôle de coordinateur du commité
- Doit vérifier que les processus de gouvernances sont respectés

Section processus de contribution et décision

- N'importe qui peut contribuer au projet indépendamment de ses compétences (cf dictateur bienveillant)
- La majorité des décisions sont prises par obtention de consensus *paresseux*
- Les décisions concernant l'avenir du projet sont prises en considérant la communauté, i.e. de l'utilisateur le plus récent au membre du commité le plus expérimenté
  - les décisions sur l'orientation stratégique du projet ou sur des aspects juridiques doivent obtenir une approbation par vote;
  - tous les membres sont encouragés à participer aux discussions / vote;

Section processus de contribution et décision

- ◦ les membres du comité ont des voix contraignantes sur la prise de décision finale (déclarent le débat clôt, prennent une décision finale, ...).
- Les discussions de gestion non sensible ont lieu sur la liste de diffusion des contributeurs au projet
- Les discussions de gestion sensible ont lieu sur une liste privée

Contribution libre

- Modèle de gouvernance plus récent
- Organisation proche du modèle Méritocratie :
  - les personnes qui font le plus de travail sont reconnues comme les plus influentes MAIS cela se base sur la travail courant et non sur les contributions historiques;
  - la prise de décision est réaliée par recherche de consensus;
  - recours au comité technique lorsqu'un consensus n'est pas trouvé;
  - discussion en interne et vote si aucun consensus interne n'est obtenu.

Exemples de projets associés à ce type de modèle : Rust, Node Js

Divergence avec la *tradition* du libre

Différence sur l'obtention du privilège de commiter : tout contributeur proposant une contribution non triviale doit être ajouté en tant que *commiter* et avoir un accès en écriture au référentiel !

Pourquoi évoluer en ce sens ?

1. **La limitation du droit de validation n'est plus autant nécessaire**

- historiquement cela avait un sens avec les anciens gestionnaires de version tel que Subversion mais ce n'est plus le cas avec Git
- avec Subversion un contributeur pouvait mettre le bazar par inadvertance
- il n'y a pas grand chose qui ne puisse être corrigé via Git

Divergence avec la *tradition* du libre

2. Tous les commiters n'ont pas le doit de prendre des décision de haut niveau !

- augmente la base de commiter pour la revue de code et le triage des bogues
- les petits changements sont examinés et ajustés sans l'intervention des contributeurs les plus techniques !
- contributeurs techniques peuvent se consacrer à d'autres tâches dédiées

IV.2 – Consensus et vote

Quels éléments sont communs aux modèles de gouvernance précédents ?

Quels éléments sont communs aux modèles de gouvernance précédents ?

1. Le groupe fonctionne majoritairement par consensus
2. Lorsque le consensus n'est pas possible, utilisation d'un mécanisme de vote

Qu'est ce qu'un consensus ?

Quels éléments sont communs aux modèles de gouvernance précédents ?

1. Le groupe fonctionne majoritairement par consensus
2. Lorsque le consensus n'est pas possible, utilisation d'un mécanisme de vote

Qu'est ce qu'un consensus ? Accord que tout le monde est prêt à accepter !

💭 Pas d’ambiguïté sur l'atteinte d'un consensus : en général, la discussion s'arrête après que quelqu'un propose qu'un consensus a été atteint.

⚠️ Dernière chance pour que quelqu’un d’autre exprime son désaccord !

💭 Lors de la proposition d'atteinte d'un consensus, il est important de rappeler quel est le consensus et les conséquences qu'il aura.

Consensus implicite

Les consensus implicites apparaissent dans les décisions non controversées :

- correction de bogue
  - le commit en lui même est une proposition de consensus
  - dans le tête du développeur il signifie *"je suppose qu'on est tous d'accord que ce bogue doit être corrigé, et que c'est la manière de le faire"*
- la documentation d'une interface, ...

Consensus implicite

Les petites décisions non controversées sont des consensus implicite :

- le commit est une proposition de consensus implicite;
- s'il n'y a pas de réponse à ce commit, un silence équivaut à un consentement
- s'il y a discussion, les changements sont discutés comme s'il n'étaient pas commit

💭 Les consensus implicites nécessitent une certaine période de temps avant de considérer qu'il n'y a pas d'objection (généralement au moins 72 heures)

Pourquoi est ce que ce n’est pas pas gênant de procéder ainsi (consensus implicite) ?

Pourquoi est ce que ce n’est pas pas gênant de procéder ainsi (consensus implicite) ?

Les logiciels de versionnement : **un garant du retour en arrière** !
Permet de modifier un changement que l'auteur pensait satisfaisant pour tous
Compromis entre le démarrage d'une contribution et le consensus : pas de crainte de démarrer une contribution; demande de consensus non systématique

Le processus d'établissement d'un consensus n'a pas besoin d'être très formel:

- modif mineurs : pas de discussions OU discussions avec "hochement de tête"
- modif importantes : discussions et attente de quelques jours avant consensus

⚠️ Psychologiquement plus difficile d'annuler un changement que de l'empêcher de se produire

Le vote : un autre moyen pour décider

Quand doit-on voter ?

Le vote : un autre moyen pour décider

Quand doit-on voter ?

C'est une des choses les plus difficile à déterminer à propos du vote !
💭 En dernier recours, si les autres options ont échouées.

Pourquoi éviter le vote si possible ?

Le vote : un autre moyen pour décider

Quand doit-on voter ?

C'est une des choses les plus difficile à déterminer à propos du vote !
💭 En dernier recours, si les autres optiosn ont échouées.

Pourquoi éviter le vote si possible ?

- le vote n'est pas un bon moyen de résoudre un débat !
- le vote met fin à la discussion et donc à la réflexion créative
- le vote règle le débat par décompte != débat rationnel aboutissant à une solution globale

Pourquoi est-il préférable de négocier un compromis que de tenir un vote ?

Pourquoi est-il préférable de négocier un compromis que de tenir un vote ?

- Compromis : tout le monde est un peu mécontent
- Vote : certains sont heureux d'autres mécontents

💭 Plus intéressant d'avoir tout le monde un peu insatisfait mais ayant la sensation d'avoir tiré un prix de son malheur !

Comment éviter un vote prématuré ?

Comment éviter un vote prématuré ?

- Simplement dire *"je ne pense pas que nous soyons encore prêts pour un vote"* et expliquer pourquoi
- Demander un vote informel, par exemple à main levée
  💭 Psychologiquement, si le vote penche d'un côté, facilite les réticents à faire des compromis
- Relancer la discussion, par exemple proposer une nouvelle solution ou un nouveau point de vue sur une ancienne suggestion

⚠️ Ne pas faire de compromis à tout prix ! Les solutions avec compromis peuvent être pire que celles sans !

💭 Un bon moment pour voter : conduit à une meilleur solution que les compromis & les contributeurs ne seront pas trop mécontents.

Le vote c'est bien mais qui doit voter ?

Le vote c'est bien mais qui doit voter ?

⚠️ Sujet sensible car il oblige à reconnaître officiellement que certains sont plus impliqués / ont un meilleur jugement que d'autres !

💭 Une bonne solution : accorder le droit de vote au contributeurs ayant les droits de commit

⚠️ Le droit de commit peut être partiel (exemple: modifications autorisées dans la documentation, les bindings vers d'autres langages, ...) ou complet !

- Si droit partiel : définir si le droit d'accès implique un droit de vote ou si c'est différent
- Si droit complet : prudence lors de l'accord au droit de commit complet !
  ⚠️ Octroiement droit complet : capacité technique ET membre de l'électorat

IV.3 – Commiters

Rôle retrouvé dans tous les projets libre, peu importe le modèle de gouvernance ET rôle lié au contrôle de la qualité du code

💭 **Conviction personnelle** : fonction nécessaire et je ne pense pas qu'un projet open source puisse réussir sans elle ! En général, les gens se sentent compétent pour apporter des changement au code.

Le projet ne peut s'appuyer sur le propre jugement de chacuns et doit imposer des normes. Il doit donner l'accès aux commits uniquement à ceux qui respectent les normes !

Dans cette partie nous allons discuter :

- des normes pour juger de nouveaux commiters potentiels;
- du processus de jugement par rapport à la communauté.

Comment choisit-on un commiter ?

Comment choisit-on un commiter ?

Les compétences techniques / connaissances du code ne sont pas forcément le critère de sélection principal !

Application du principe d'Hippocrate : ne pas nuire !

- Savoir **faire preuve de jugement**, i.e. être conscient de ses capacités et savoir quoi entreprendre ou non
- Les compétences techniques peuvent être acquises et enseignées, ce n'est pas toujours le cas pour le jugement

💭 Un commiter peut proposer la candidature d'un nouveau commiter même si celui-ci n'a fait que des petits correctifs MAIS que ceux-ci sont suffisamment nombreux pour décrire un pattern et : s'appliquent proprement; ne contiennent pas de bogue; suivent les conventions de codage.

Comment choisit-on un commiter ?

Dans certains projet, il est attendu qu'un individu souhaitant devenir commiter propose des correctifs non triviaux. Ces projets veulent être sûre que la personne :

- ne sera pas *"mauvaise"*;
- devrait pouvoir faire du *"bien"* sur toute la base du code.

⚠️ Ce n'est pas un mauvais critère mais il faut être prudent :

- Ne pas transformer *l'engagement* en *"appartenance à un club sélectif"*
- La question devrait être *"qu'est ce qui apportera le plus au code ?"* et pas *"ne va-t-on pas dévaloriser le statut de commiter ?"*

Comment choisit-on un commiter ?

Un autre aspect à prendre en compte lors du choix d'un nouveau commiter est le **comportement** !

Dans les projets open source, **les compétences sociales sont aussi importantes que la capacité technique brute** !

💭 Les compétences techniques et le suivi des directives du projet ne doivent pas assurer le privilège de commiter !
Par exemple : individu querelleur sur les listes de diffusions / IRC

💭 Avec les logiciels de gestion de versions, le fait d'ajouter un *"mauvais"* commiter n'est pas trop problématique vis-à-vis du code
⚠️ Cependant, cela peut amener à révoquer les droits d'accès d'une personne, i.e. une action désagréable et parfois conflictuelle !

Révoquer les droits de commit

Première chose à savoir : éviter cette situation avant d'y être contraint !
⚠️ Les discussions associées à cette action peuvent être très conflictuelles, diviser la communauté, nuire à la productivité par son aspect distractif

Si cette action doit être réalisée :

- la discussion sur la **révocation doit être réalisée en privée**;
- la discussion ne doit **pas inclure la personne concernée**.

💭 **Contradiction avec l'idée de transparence du libre / open source** mais c'est nécessaire. Cela permet à tous le monde de parler librement et, si le vote échoue, cela évite que la personne ciblée soulève des questions créant des *"factions"*, e.g. *"qui a voté pour moi ? contre moi ?"*

Révoquer les droits de commit

Pour servir d'avertissement, la personne ciblée peut être notifiée pendant / après la délibération.

⚠️ Cette notification doit venir d'une décision prise par le groupe

💭 Lorsque la révocation a lieu, celle-ci est rendu publique

⚠️ Besoin délicatesse dans la façon de présenter les choses aux monde extérieur

Commiter partiel

Dans certains projets, il existe différent *"grades"* de commiter :

- accès libre sur la **documentation mais pas sur le code**,
- accès sur les **tests mais pas sur le développement de nouvelles fonctionnalité**,
- ...

Zone communes d'accès : documentation, traduction, bindings, ... (des zones où l'insertion d'une erreur n'induira pas de problème pour le cœur du projet)

Souvent, le droit de commit ne se résume pas à l'écriture de correctifs et inclut aussi l'appartenance à un électorat (cf section précédente)

Commiter partiel

Sur quoi les commiters partiels peuvent-ils votés ?

Commiter partiel

Sur quoi les commiters partiels peuvent-ils votés ?

💭 Pas de bonne réponse, typiquement cela peut dépendre des grades de commiter

💭 Cela ne doit pas empêcher quelqu'un de s'exprimer, e.g. faire part de son approbation / sa désaprobation avec des "+1" / "-1" au lieu de bulletin de vote !

Comment sont élus les nouveaux commiter partiel ?

Commiter partiel

Sur quoi les commiters partiels peuvent-ils votés ?

💭 Pas de bonne réponse, typiquement cela peut dépendre des grades de commiter

💭 Cela ne doit pas empêcher quelqu'un de s'exprimer, e.g. faire part de son approbation / sa désaprobation avec des "+1" / "-1" au lieu de bulletin de vote !

Comment sont élus les nouveaux commiter partiel ?

- Tout commiter complet peut *"parrainer"* un nouveau commiter partiel
- Les commiters partiels d'un domaine peuvent choisir des nouveaux commiters partiels pour ce même domaine

Commiter partiel

Selon les projets, les points précédents peuvent variés mais les principes suivant s'appliquent à la plupart des projets :

- Un commiter devrait pouvoir voter sur des questions de son domaine
- Un commiter ne devrait pas pouvoir voter sur des questions en dehors de son domaine
- Les votes sur les procédures devraient être limités aux commiteurs complets

Commiter dormant

Doit-on retirer les droits de commit aux commiters dormant ?

Commiter dormant

Doit-on retirer les droits de commit aux commiters dormant ?

💭 **Conviction personnelle** : non pour deux raisons :

- Cela **peut inciter les commiters à valider des modifications inutiles** (mais acceptables) pour ne pas être considéré inactif
- Cela **ne sert à rien de retirer les droits**. Le jugement se détériore en étant absent d'un projet ? Le commiter ne serait pas capable de comprendre qu'il est déconnecté du projet ?

💭 Un peu le même principe que le télétravail et les jours pour le faire

💭 Les droits peuvent évidemment être retirés à la demande de la personne

Transparence

Les discussions autours d'un nouveau commiter doivent être privées mais les règles et procédures pour le devenir n'ont pas besoin de l'être !

- Permet de clarifier que n'importe qui peut prétendre à ce privilège en proposant des bon patchs
- Démystifie le groupe des commiters comme n'étant pas des illuminatis

La liste des commiters doit être publiée !

- Traditionnement dans un fichier dédié appelée COMMITERS / MAINTAINERS / AUTHORS à la racine de l'arborescence du projet
- Liste les commiters complets, les domaines de commits partiels, les membres de chaque domaine
- Chaque commiteur doit être répertorié par son nom et adresse électronique

IV.4 - Autres rôles

Autres rôles

A mesure qu'un projet devient plus complexe, le travail consiste à gérer les personnes et les flux d'informations.
⚠️ Besoin de partager les tâches de gestion et les tâches techniques !

Nous allons voir quelques rôles qui peuvent apparaître parfois consciemment (par les membres du projet) et d'autres fois spontanément. Aucun des rôles ne nécessitera un contrôle exclusif sur le domaine associé, i.e. le responsable n'empêche pas les autres d'apporter des modifications dans son domaine.

Le travail d'un responsable consiste à : remarquer quand des personnes travaillent dans son domaine; former ces personnes à faire les choses de manière cohérente.
💭 Renforce les efforts multiples et évite les conflits

Responsable de correctifs

Quel est le rôle du responsable de correctifs ? Que peut-on attendre de lui ?

Responsable de correctifs

Quel est le rôle du responsable de correctifs ? Que peut-on attendre de lui ?

Le rôle du responsable de correctif est de **s'assurer que les correctifs ne *"glissent pas entre les mailles du filet"***.

Exemples concrets de correctifs qui pourraient glisser :

1. Un correctif est envoyé et relues par un examinateur qui y voient des erreurs

Celui-ci demande des modifications à l'auteur du correctif ... processus itératif qui converge lorsque l'examinateur n'a plus rien à critiquer et valide le correctif.

❓ Comment faire lorsque l'examinateur ne le fait pas (e.g. manque de temps) ?

Responsable de correctifs

2. Un correctif est envoyé et déclenche une discussion en roue libre

Typiquement, le correctif corrige un bogue mais le concept utilisé n'est pas jugé comme étant le plus adapté car le projet préférerait utiliser une autre manière pour résoudre une classe de problème plus vaste.

💭 Cet événement n'est pas connu à l'avance et c'est le correctif qui amène à cette découverte !

❓ Comment faire pour stopper la discussion ?

Responsable de correctifs

3. Un correctif est envoyé et c'est le néant absolu, personne ne répond

Fréquent lorsque tous le monde à peu de temps à ce moment-là et que chacun espère que quelqu'un d'autre s'en chargera

❓ Comment s'assurer que ce correctif sera examiné lorsque d'autres correctifs / actions prioritaires se présentent par la suite ?

⚠️ Ce cas de figure est préjudiciable au projet car il décourage l'auteur du correctif et communique un sentiment de *"déconnection"* aux autres contributeurs potentiels

Responsable de correctifs

Action réalisée par le responsable de correctifs : **suivi des correctifs jusqu'à un état *"stable"***. Par exemple dans le cas :

1. Il signale un problème en pointant vers la version finale du correctif et la liste de diffusion associée.
💭 Si le correctif résout un problème existant, il annote ce problème avec les informations pertinentes au lieu d'ouvrir un nouveau ticket

3. Il attend quelques jours et, s'il n'y a toujours pas de réponse, demande si quelqu'un compte examiner le correctif
💭 Soit cela déclenche une réaction (c'est généralement le cas) et il y a itération, soit il ouvre (ou non) un ticket sur le correctif et au moins l'auteur a eu une réponse

Responsable de correctifs

Intérêt du responsable de correctifs : économie de temps et d'énergie mentale

Sans cela, chaque développeur peut se demander constamment :

- s'il peut compter sur quelqu'un d'autres pour répondre à un correctif;
- s'il doit garder un œil sur le correctif, ...

💭 Potentiel duplication des efforts qui est totalement inutile
💭 Le gestionnaire de correctif supprime la mise en doute de la situation !

⚠️ Ce système ne fonctionne que si les gens peuvent compter sur la présence du responsable sans faute ! Le rôle doit être tenu formellement.

Responsable de traduction

Quel est le rôle du responsable de traduction ? Que peut-on attendre de lui ?

Responsable de traduction

Quel est le rôle du responsable de traduction ? Que peut-on attendre de lui ?

Le rôle du responsable de traduction est d'**aider les traducteurs à se coordonner entre eux et de faire l'intermédiaire avec le reste du projet**

La notion de traduction peut englober la documentation du logiciel dans d'autres langues mais aussi traduire le logiciel lui même, e.g. les messages d'aides ou d'erreurs

💭 Il peut être judicieux d'avoir un seul gestionnaire ou bien d'avoir un gestionnaire par aspect de la traduction

Responsable de traduction

Rôle n'impliquant **pas nécessairement de faire des traductions soit même**.
Mais, c'est bien évidemment possible.

Rôle essentiel car les traducteurs sont un groupe d'individus très différents des développeurs :

- pas forcément d'expérience de travail dans une équipe;
- rarement une expérience de travail incluant une outil de contrôle de version !

💭 Compétences importantes du poste : **plus diplomatiques que techniques**

Responsable de traduction

Exemple : cas d'un projet réalisant de la traduction croisée, i.e. plusieurs personnes doivent travailler sur une même traduction:

- le gestionnaire met en contact des personnes traduisant le même langage;
- si personne n'est disponible pour la même langue, il invite courtoisement le volontaire à trouver un/d' autre(s) volontaire(s)

Exemples d'actions réalisées par le responsable pour les traducteurs :

- configurer les droits d'accès approprié au référentiel;
- informer les traducteurs des conventions (e.g. le template des commits);
- suivre les commits pour vérifier que les conventions sont respectées;
- mettre en place des listes de diffusions pour chaque langue.

Responsable de documentation

Quel est le rôle du responsable de documentation? Que peut-on attendre de lui?

Responsable de documentation

Quel est le rôle du responsable de documentation? Que peut-on attendre de lui?

Le rôle du responsable de documentation est d'aider à **maintenir la documentation organisée, à jour et cohérente avec elle-même** !

💭 Tenir à jour la documentation est une tâche sans fin

- chaque fonctionnalité / amélioration du code entraîne des changements;
- une fois la documentation ayant une certain exhaustivité, une grande partie des correctifs proposés seront sur la documentation et pas sur le code. La raison étant qu'il y a beaucoup plus de personnes pour faire de la prose que de personnes compétentes pour corriger un bogue dans le code !

Responsable de documentation

💭 Champ d'actions qui peut s'intersecter avec le responsable de correctifs !

- Intersection possible au niveau des correctifs de documentation
- Ces deux rôles peuvent être attribués à une même personne mais lorsque la quantité de correctif est trop important : deux gestionnaires distincts !

💭 Quelques propriétés des correctifs de documentation :

- en général, il ne nécessitent pas de test à réaliser;
- le rapport *"frais administratif"* et *"productivité élevé"* est plus élevé pour les correctifs de documentation que de code;
- les correctifs peuvent avoir besoin d'ajustement pour maintenir le ton de l'auteur cohérent.

Responsable de documentation

En résumé, les exigences de gestion des correctifs de documentation et le fait que la base de code doit être surveillée pour maintenir la documentation amènent au besoin d'avoir une personne / équipe dédiée à cette tâche !

En pratique le responsable de documentation :

- enregistre où/comment la documentation est en retard par rapport au projet;
- connaît intimement la documentation;
- surveille les modifications apportées à la documentation;
- surveille l'évolution du code.

💭 Il maintient la documentation en bonne santé !

Responsable de tickets

Quel est le rôle du responsable de tickets ? Que peut-on attendre de lui ?

Responsable de tickets

Quel est le rôle du responsable de tickets ? Que peut-on attendre de lui ?

Le rôle du responsable de tickets est d'aider à **clarifier la base de données de tickets**

💭 Le nombre de tickets suivis dans l'outil de gestion des bogues augmente proportionnellement au nombre d'utilisateurs
=> Augmentation *"sans fin"* du nombre de tickets ouverts

Une des tâches courante du gestionnaire est de **gérer les *"mauvais"* tickets** :

- limiter le nombre de tickets incomplets ou mal écrits, e.g. des champs du formulaire ne sont pas correctement remplis
- limiter le nombre de tickets doublons

💭 Il est donc intéressant que le gestionnaire soit familier de la bdd des bogues!

Responsable de tickets

Une autre tâche classique consiste à **établir les connections entre problèmes et développeurs** :

- les développeurs ne peuvent pas être conscient de toutes les issues !
- le gestionnaire connaît l'équipe de développement et peut diriger l'attention de certains développeurs sur un ticket spécifique

💭 Il est intéressant que le gestionnaire soit lui même un développeur. Ainsi, il devrait être conscient de la sensibilité des autres développeurs, de leur tempérament et désirs.

Responsable de tickets

[Optionnel] Le responsable peut *"arranger"* la base de données pour refléter les priorités du projet

- Définir qu'un bogue sera corrigé dans une version spécifique future du logiciel
- Définir qu'une version du logiciel sera utilisée comme jalon de résolution d'un ticket

💭 Cela donne aux utilisateurs une visibilité sur la base de données des tickets au travers de la planification des versions !

⚠️ Ces cibles ne sont pas forcément statique car de nouveaux bogues dont la résolution est plus prioritaire peuvent arrivés à tout moment !

Responsable de tickets

Une autre tâche du responsable consiste à **constater qu'un problème devient obsolète**

- Problème corrigé accidentellement par une modification en dehors du projet
- La modification du statut de comportement bogué en comportement attendu

A mesure que la base de données grandi, le responsable va de plus en plus adopter la stratégie de diviser pour régner :

- catégoriser les problèmes à leur arrivé;
- diriger les problèmes vers le développeur / équipe approprié(e);
- être épaulé par d'autres (*"Triagers"*)

💭 Le gestionnaire devient un coordinateur général !

Responsable de tickets

Exemple de documentation associée à ce rôle :

- https://github.com/nodejs/node/blob/main/README.md#triagers
- https://github.com/nodejs/node/blob/main/doc/contributing/issues.md#triaging-a-bug-report

Responsable de FAQ

Quel est le rôle du responsable de FAQ ? Que peut-on attendre de lui ?

Responsable de FAQ

Quel est le rôle du responsable de FAQ ? Que peut-on attendre de lui ?

Le rôle du responsable de FAQ est d'**aider à clarifier la FAQ**

⚠️ La maintenance d'une FAQ diffère des autres documents d'un projet : le contenu n'est pas planifiable à l'avance ET le document est totalement réactif !

Difficultés associées à la maintenance d'une bonne FAQ :

- le document peut facilement devenir incohérent et désorganisé;
- il est facile d'y ajouter des entrées double ou *"semi-double"*;
- trouver les liens qui devraient être fait entre éléments interdépendant (notamment à cause du temps d'intervalle entre les éléments)

Responsable de FAQ

En résumé, le rôle du responsable est de :

- **Maintenir la qualité globale de la FAQ (doublons et liens)**
  💭 Nécessite d'être familier avec les sujets des questions qu'elle contient
- Surveiller la liste de diffusion du projet et les forums pour **trouver des questions récurrentes et étendre la FAQ** avec ces questions
  💭 Nécessite d'être capable de :
  - suivre des fils de discussion;
  - extraire les questions fondamentales;
  - proposer une entrée dans la FAQ;
  - intégrer les commentaires des autres;

Responsable de FAQ

Le responsable de FAQ est généralement l'expert en formatage FAQ, i.e. il *"passe derrière"* les contributions pour nettoyer les petits détails oubliés

⚠️ Le rôle de responsable FAQ est associé à une grande quantité de travail qui **tend à être automatisé MAIS attention à la sur-automatisation** !!
Par exemple : l'automatisation du processus de maintenance en permettant à tout le monde de contribuer et d'éditer la FAQ (comme un wiki)

💭 La **décentralisation induit une réduction de la charge de travail de maintenance** MAIS **au dépend de la qualité de la FAQ**

Responsable de FAQ

Pourquoi cette perte de qualité ?

Responsable de FAQ

Pourquoi cette perte de qualité ?

- Personne ne surveille les interdépendances entre éléments
- Personne n'a de vue d'ensemble de la FAQ
- Personne pour constater le besoin de mettre à jour un élément
- Personne pour constater qu'un élément est devenu obsolète

=> La FAQ ne fournit pas ce que cherche les utilisateurs et parfois peut même les induire en erreur !