

# Retours d'un randonneur en C++

Alexis Colin

Télécom SudParis

# Une équipe est un groupe social

## Définition

Un groupe social est un ensemble d'individus formant une unité sociale durable, caractérisé par des valeurs communes, des liens plus ou moins intenses, une situation sociale identique et/ou des activités communes, une conscience d'appartenir à ce groupe ou par la reconnaissance, par d'autres groupes, de son existence.

## Ce qui peut fabriquer un groupe social

- L'auto-constitution du groupe
- Des caractéristiques identifiées par un groupe social plus large
- Des intérêts matériels ou des occupations communes
- L'uniformisation ou l'adaptation des comportements par mimétisme ou nécessité d'interaction entre des individus

# Avec qui je travaille ?

## Des métiers divers

- Des créatifs (auteur·es, musicien·nes, graphistes 2D/3D, designers...), des QA...
- Des programmeur·ses (web, tools, graph, UI, GP, art, system, engine...)
- Des managers, du marketing, des community managers...

## Des humains divers

- Des gens avec la flamme et des gens avec la flemme
- Des gens enthousiaste et des blazés
- Des gens seuls, avec des enfants et/ou un·e conjoint·e
- Des gens passionnés d'optim CPU, de train ou de jardinage
- Des gens avec des problèmes perso, un proche malade...

Des gens quoi.

# Qu'est-ce qu'une culture ?

## Définition - Éthologie et philosophies matérialistes

Tout comportement, habitude, connaissance, croyance, système de sens, modes de raisonnement ou d'expérimentation... appris par un individu biologique et transmis socialement (non biologique).

## Caractéristiques

- Des systèmes comportementaux et perceptifs interdépendants
- Des ensembles en perpétuelle évolution
- Reflet et pilier de l'organisation sociale
- Attention aux cultures hégémoniques !

# Communiquer

## Votre travail...

- ...concerne potentiellement tout le monde
- ...restera plus longtemps que vous

## Vous ne devez pas :

- Prendre vos interlocuteur·ices pour des andouilles
- Rendre votre sujet inaccessible techniquement
- Rendre votre sujet inaccessible symboliquement
- Participer à une culture oppressive / élitiste / excluante

## Communiquer par le code - Nommage

```
void set_default(ConfigManager const & manager) {  
    set(manager.get_default());  
}  
...  
set_default(manager);
```

---

```
void use_default_global_input_configuration(  
    InputConfigurationList const & input_configs) {  
    set_global_input_config(input_configs.  
        get_first_valid_configuration());  
}  
...  
use_default_global_input_configuration(  
    input_configurations);
```

# Communiquer par le code - Commentaires

## Justifier

```
/* On melange aleatoirement les donnees pour limiter le risque de  
tomber sur le cas ordonne */
```

## Expliquer

```
/* On cherche le point d'interêt le plus proche n'ayant pas encore  
été évalué */
```

```
/* Equivaut à une multiplication par 2 en s'assurant que les bits de  
poids fort restent à zéro */
```

## Aider

```
/* A partir d'ici la memoire pointee ne peut plus etre consideree  
comme valide */
```

## Spécifier/Documenter

ex : Doxygen



# Communiquer par les outils - Documentation

## Les questions à se poser

- À qui s'adresse la documentation ?
- Quelles sont les informations utiles ?
- Comment la tenir à jour ?

## Idées de parties à ajouter

- À qui la page de documentation s'adresse principalement !
- Un résumé, un historique des modifications, un sommaire
- Les éléments de contexte et les objectifs au moment de la conception : « Comment et pourquoi on en est là ? »
- Des détails techniques *datés* et vulgarisés → tout le monde doit pouvoir lire la doc

# Communiquer par les outils - Commit et merge-request

## Les bases du commit

- Tous les commits fonctionnent et sont indépendants
- Chaque message de commit permet de naviguer efficacement dans l'historique

## Les bases de la Merge/Pull Request

- Informer les personnes concernées
- Expliquer ce que vous avez fait, ce dont vous doutez

## Mais...

- Observer ce qui se fait dans votre équipe !
- Participer à la répartition de la connaissance du code
- Ne pas participer à l'exclusion / la placardisation de quelqu'un

## Mettre les formes - le formatage

```
auto index_of(span<int> const & v, int value) -> size_t {  
    for (auto const i : indexes_of(v))  
        if (v[i] == value)  
            return i;  
    unreachable;  
}
```

---

```
size_t getIndexOf(const span<int>& v, int value)  
{  
    const size_t size = v.size();  
    for( size_t i = 0; i != size; ++i )  
        if( v[i] == value )  
            return i;  
    assert(false);  
}
```

# Pourrissement du code

## Les causes

- Une accumulation de modifications pas forcément prévues à la conception, un manque d'organisation
- Des départs ou une mauvaise répartition du travail et des connaissances (documentation...)
- De mauvaises conditions de travail, la lassitude...
- Un manque de compétences, d'habitude : C'est NORMAL

## Les conséquences

- Perte de contrôle et de temps
- Frustration, énervement, démotivation

## Mais...

- Un code pourri peut aussi être un code stabilisé

## Path of darkness

### La négativité peut être fonctionnelle (en partie)

- Elle dicte des normes de qualité de code
- Elle évacue et permet de fraterniser autour de la frustration
- Elle nous éloigne de ce qui nous coûte nerveusement

### avec des conséquences dramatiques

- La négativité est très contagieuse
- Création de tensions entre les gens
- Démotivation, appréhension, angoisses...

### et doit être dépassée

- Écrire du code est un artisanat et non une science
- La résignation n'est pas une solution

# Code review - Qu'est-ce que c'est ?

## Objectifs

- Détecter davantage de bugs, d'erreurs
- Ouvrir un espace de discussion et de partage de compétence
- Casser la propriété du code, partager la responsabilité

## Comment ?

- À l'oral ou à l'écrit
- En direct ou via un outil (github, gitlab...)

## Une pratique qui fonctionne si...

- elle est adossée à d'autres outils (CI, unit tests...)
- elle n'est pas l'expression d'un rapport de pouvoir
- elle est réalisée dans la réciprocité et la confiance

# Code review - Comment faire ?

## Toute évaluation est une violence

- Il s'agit de recevoir et porter des jugements !
- Le paradoxe du maître ignorant : conditionnement au statut d'apprenant/aidé

## Soumettre

- Guider les relecteur·ices vers les points importants
- Indiquer quel genre de retours on souhaite

## Commenter

- Est-ce que c'est moi qui ne comprends pas ?
- Ma remarque est-elle pertinente ?
- Ma remarque est-elle claire et correctement justifiée ?