

# Partie 2 — Comprendre la logique du libre et de la collaboration

## Objectif général

Découvrir la culture du **logiciel libre** et les modes de **collaboration ouverts** qui structurent la production logicielle contemporaine.

Comprendre les **valeurs**, les **règles de fonctionnement** et les **modèles économiques** qui permettent au libre de vivre, d'évoluer et de s'auto-organiser.

## Compétences mobilisées

- Identifier les grands principes éthiques et juridiques du logiciel libre.
- Comprendre les rôles, responsabilités et processus collaboratifs dans un projet communautaire.
- Évaluer la viabilité économique d'un modèle open source.
- Participer de manière pertinente à une contribution technique ou documentaire.

## Contenus de cours

### 1. Esprit et histoire du logiciel libre

- Origines du mouvement : Richard Stallman, Free Software Foundation, Open Source Initiative.
- Distinction entre *free software* et *open source* : liberté d'usage, pas seulement gratuité.
- Quatre libertés fondamentales : exécuter, étudier, modifier, redistribuer.
- Impact culturel : partage, transparence, apprentissage collectif, gouvernance ouverte.

### 2. Licences et droits d'usage

- Comprendre les licences : pourquoi elles existent et ce qu'elles protègent.
- Principales licences :
  - **MIT** : permissive, simple, très répandue.
  - **GPL / AGPL** : viralité du code et partage obligatoire.
  - **Apache 2.0** : protection contre les brevets.
  - **Creative Commons** : pour les contenus non logiciels (textes, visuels, docs).
- Savoir lire et choisir une licence : compatibilités, obligations, usages commerciaux.
- Présentation des outils d'aide au choix (ChooseALicense.com, SPDX).

### 3. Modèles économiques du libre

- Financement du logiciel libre : mythes et réalités.
- Modèles existants :
  - **Sponsoring / mécénat** (individuel ou entreprise).
  - **Services associés** (support, formation, intégration).
  - **Open core** : cœur libre, modules propriétaires.
  - **Dual licensing** : licence libre + licence commerciale.
  - **Produits dérivés et SaaS** : GitLab, Nextcloud, Elastic.
- Problèmes de soutenabilité : fatigue des mainteneurs, dépendances critiques, modèle bénévole.

### 4. Gouvernance et collaboration communautaire

- Organisation d'un projet open source :
  - Rôles : contributeur, mainteneur, core team, reviewer.
  - Processus : *issue, fork, branch, pull request, review, merge*.
- Bonnes pratiques :
  - Rédiger une documentation claire et à jour.
  - Respecter le *code of conduct* et les conventions de contribution ( *CONTRIBUTING.md* ).
  - Communiquer avec bienveillance : importance du ton, du feedback et de la transparence.

## Activités proposées

### Atelier 1 — Étude de cas : anatomie d'un projet open source

- Choisir un projet libre majeur (ex : Symfony, Laravel, Blender, Nextcloud, VS Code).
- Analyser :
  - Son modèle économique et sa licence.
  - Sa structure organisationnelle (core team, communauté).
  - Ses outils de contribution (issues, PR, docs, roadmap).
- Restituer l'analyse sous forme d'une courte synthèse écrite.

## Atelier 2 — Contribution minimale

- Identifier une petite tâche ouverte sur un projet libre : faute de frappe, amélioration de documentation, ajout de lien manquant.
- Réaliser cette contribution via un fork et une pull request.
- Décrire le processus suivi et les échanges éventuels avec les mainteneurs.

## Livrables attendus

- Fiche d'analyse d'un projet open source (1 à 2 pages) présentant :
  - Licence, modèle économique, structure de gouvernance.
  - Observations sur la collaboration et la qualité documentaire.
- Rapport de contribution (capture, lien ou description du processus).

## Critères d'évaluation

Critère	Description	Pondération
Compréhension du modèle libre	Clarté et précision de l'analyse du projet choisi	40 %
Qualité de la documentation de la contribution	Pertinence, rigueur et mise en contexte	40 %
Posture collaborative	Respect des codes du libre (communication, transparence, bienveillance)	20 %

# Pistes d'approfondissement

- Étudier le fonctionnement de plateformes comme **GitHub Sponsors** ou **Open Collective**.
- Observer les modèles hybrides de projets récents (ex : Sentry, Strapi, Supabase).
- Explorer la notion de **communautés de pratique** : contribution non-technique (traduction, doc, design).

## Message clé

Le logiciel libre n'est pas qu'une méthode de diffusion,  
c'est une **culture de coopération** où la valeur du code vient de sa **transparence et de sa transmission**.

Savoir collaborer, c'est apprendre à coder avec et pour les autres.