Gestion du projet

- Mise en place
- Analyse

Mise en place

OdooAsso — Projet de modules Odoo pour la gestion associative

Contexte

Odoo présente une orientation naturelle vers l'entreprise et un niveau de complexité d'usage trop important pour les utilisateurs d'associations. L'objectif de ce projet est de regrouper tous les besoins en quelques modules cohérents pour simplifier au maximum la gestion, et de produire un travail de qualité suffisante pour être partagé avec d'autres associations.

Stack cible : Odoo 16 Community, déployé en Docker. **Modules tiers utilisés :** modules OCA (vertical association, membership, partner…), modules Deodoo.

Architecture du dépôt

Structure monorepo inspirée du standard OCA :

```
OdooAsso/                          # dépôt public GitHub/GitLab
├── README.md
├── projet.md
├── .pre-commit-config.yaml        # linting OCA
├── setup.cfg
├── oca_dependencies.txt           # versions des modules OCA requis
├── asso_base/                     # socle — groupes, menus racine, mixins
├── asso_membership/               # gestion des membres et cotisations
```

```
├── asso_event/                  # événements et activités
├── asso_accounting/             # comptabilité simplifiée (loi 1901)
├── asso_portal/                 # portail adhérent
└── asso_website/                # front public
```

Principe de séparation

`asso_base` déclare les groupes d'accès, les menus racine et les mixins réutilisables. Tous les autres modules en dépendent. Les modules métier sont indépendants les uns des autres autant que possible, pour permettre une adoption partielle.

Namespace

Le préfixe `asso_` est choisi pour être neutre et réutilisable (pas de nom d'association spécifique). Si le projet vise explicitement le contexte français (loi 1901, plan comptable associatif), le préfixe `fr_asso_` peut être envisagé.

Stratégie de simplification UX

C'est le cœur de la valeur ajoutée du projet. Odoo est simplifié à trois niveaux :

1. Groupes d'accès dédiés

Des groupes propres sont créés, sans réutiliser les groupes Odoo natifs :

- `group_asso_user` — membre du bureau, usage courant
- `group_asso_manager` — responsable, accès étendu
- `group_asso_admin` — administrateur technique

Cela permet de masquer tout ce qui n'est pas pertinent pour une association.

2. Menus simplifiés

- Les menus natifs Odoo sont désactivés (`active="False"`) pour les groupes `asso_*`
- Une arborescence plate et orientée tâches les remplace :

    - Membres → Activités → Finances → Communication

3. Vues allégées

- Les vues natives sont héritées pour masquer les champs inutiles
- L'attribut `groups` est utilisé sur les champs avancés (visibles admins seulement)

4. Wizards pour les actions courantes

Les opérations fréquentes sont regroupées en wizards pour éviter la navigation multi-menus : inscription en masse, relances cotisations, exports, etc.

Qualité et partageabilité

Outils

Outil	Rôle
`pre-commit` (config OCA)	Linting XML, Python, manifest
Tests unitaires Odoo	Validation des modèles et flux métier
`oca_dependencies.txt`	Déclaration explicite des dépendances OCA

Lancer les tests

```
odoo-bin --test-enable --stop-after-init -i asso_membership
```

Conventions

- Chaque module déclare ses dépendances OCA explicitement dans son `__manifest__.py`
- Les données de configuration (`data/`) sont séparées des données de démonstration (`demo/`)
- Les champs custom sont préfixés avec le nom du module (`membership_date_end` plutôt que `x_date_end`)

Environnement de développement

Le projet tourne en Docker. Ajouter un `docker-compose.override.yml` pour monter le dépôt directement dans le `addons_path` :

```
services:
  odoo:
    volumes:
      - ./OdooAsso:/mnt/extra-addons
```

Travailler en mode `--dev xml` pour recharger les vues sans redémarrer le serveur.

Modules — périmètre fonctionnel

`asso_base`

- Groupes d'accès et règles de sécurité
- Menus racine et structure de navigation
- Modèles abstraits et mixins partagés
- Configuration générale de l'association

`asso_membership`

- Gestion des adhérents (hérite `res.partner`)
- Types d'adhésion et tarifs
- Suivi des cotisations et renouvellements
- Relances automatiques

`asso_event`

- Événements et activités de l'association
- Inscriptions et présences
- Lien avec les adhérents

`asso_accounting`

- Simplification de la comptabilité pour le contexte associatif (loi 1901)
- Plan comptable associatif
- Suivi des subventions

`asso_portal`

- Espace adhérent en ligne
- Consultation et renouvellement d'adhésion
- Inscription aux événements

`asso_website`

- Pages publiques de l'association
- Formulaire d'adhésion en ligne
- Agenda public des événements

Étapes de démarrage

1. Initialiser le dépôt Git avec la structure monorepo
2. Configurer `pre-commit` avec la config OCA standard
3. Lister les fonctionnalités par module en partant des douleurs utilisateurs actuelles
4. Créer `asso_base` : groupes, menus racine, modèles abstraits
5. Migrer progressivement les développements existants dans cette structure

Analyse

Voici une méthode **simple, efficace et adaptée au monde associatif** pour structurer votre phase d’analyse et rédiger un cahier des charges léger, sans alourdir le processus. L’idée est de rester pragmatique et centré sur l’essentiel, tout en préparant le terrain pour un développement modulaire et réutilisable.

---

1. Phase d’Analyse : 4 Étapes Clés

Étape 1 : Identifier les Parties Prenantes

Objectif : Savoir qui va utiliser le système et quels sont leurs besoins réels.
Actions :

- Lister les rôles dans l’association (ex : trésorier, secrétaire, bénévole événementiel, membre).
- Pour chaque rôle, noter :
  - **3 tâches principales** qu’ils doivent accomplir avec Odoo.
  - **3 frustrations** actuelles avec l’outil (ex : "trop de champs inutiles", "je ne trouve pas où renouveler une adhésion").

Exemple de tableau synthétique :

Parties prenantes et besoins

Rôle	Tâches Principales	Frustrations Actuelles
Trésorier	Suivre les cotisations, générer des rapports	Trop de menus comptables inutiles
Secrétaire	Gérer les membres, envoyer des emails	Pas de vue d’ensemble des adhésions
Bénévole Événement	Inscrire des participants, envoyer des infos	Processus d’inscription trop long

Étape 2 : Cartographier les Processus Clés

Objectif : Visualiser les workflows actuels pour identifier les simplifications possibles.
Méthode :

- Pour chaque tâche principale, dessiner un **parcours utilisateur simplifié** (3 à 5 étapes max).
  - Exemple pour "Renouveler une adhésion" :
    1. Le membre reçoit un email de rappel.
    2. Il clique sur un lien vers Odoo.
    3. Il voit un formulaire pré-rempli avec ses infos.
    4. Il paie en ligne (ou valide un virement).
    5. L'adhésion est mise à jour automatiquement.

Outils :

- Un tableau blanc (Miro, Draw.io) ou même un papier pour schématiser.
- **Astuce** : Filmer un utilisateur qui effectue la tâche actuelle pour repérer les points de friction.

Étape 3 : Prioriser avec la Matrice "Impact/Effort"

Objectif : Concentrer les efforts sur ce qui apporte le plus de valeur avec le moins de complexité.
Méthode :

- Lister toutes les fonctionnalités souhaitées.
- Les classer dans un tableau à 4 cases :

Matrice Impact/Effort

	Effort Faible	**Effort Élevé**
Impact Élevé	À faire en priorité	À planifier (phase 2)
Impact Faible	Optionnel	À éviter

Exemple :

- **Priorité haute** : Simplifier le renouvellement des adhésions (impact élevé, effort faible si on utilise un wizard).
- **Phase 2** : Intégrer un système de paiement en ligne (impact élevé, mais effort technique important).
- **Optionnel** : Ajouter un champ "allergies" pour les événements (impact faible).

Étape 4 : Rédiger un Cahier des Charges Léger

Structure proposée (1 à 2 pages max) :

1. **Contexte** :
   - "Notre association utilise Odoo 16, mais l'outil est trop complexe pour nos bénévoles. Nous souhaitons simplifier l'interface et regrouper les fonctionnalités essentielles dans des modules dédiés."
2. **Objectifs** :
   - "Réduire de 50% le nombre de clics pour les tâches quotidiennes (ex : renouvellement d'adhésion)."
   - "Masquer les fonctionnalités non utilisées (ex : comptabilité analytique)."
   - "Permettre à d'autres associations de réutiliser nos modules."
3. **Périmètre** :
   - **Modules à développer** (liste priorisée) :
     - Gestion des membres (adhésions, profils).
     - Gestion des événements (inscriptions, participations).
     - Tableau de bord simplifié pour le trésorier.
   - **Hors périmètre** :
     - "La gestion des stocks (non utilisée par l'association)."
4. **Exigences Fonctionnelles** :
   - "Un bouton 'Renouveler' visible sur la fiche membre, déclenchant un wizard en 3 étapes."
   - "Une vue liste des membres avec filtres par statut (actif/inactif/retard)."
5. **Exigences Techniques** :
   - "Compatibilité avec Odoo 16 et les modules OCA/Deodoo existants."
   - "Code documenté et publié sur GitHub sous licence MIT."
6. **Livrables** :
   - "Modules Odoo prêts à installer (avec README et guide utilisateur)."
   - "Formation vidéo de 10 min pour les bénévoles."

---

2. Outils pour Aller Vite

- **Pour les schémas** : Draw.io (gratuit, export en PNG/PDF).
- **Pour la priorisation** : Un tableau Trello ou GitHub Projects.
- **Pour le cahier des charges** : Un Google Doc partagé ou un fichier Markdown dans le repo GitHub.

---

3. Exemple Concret : Module "Gestion des Membres"

Besoin : "Simplifier le renouvellement des adhésions."
Analyse :

- **Processus actuel** : 7 clics, passage par 3 menus différents.
- **Processus cible** : 3 clics via un wizard.
  Spécifications :
- Bouton "Renouveler" sur la fiche membre → Ouvre un formulaire avec :
  - Montant de la cotisation (pré-rempli).
  - Méthode de paiement (virement ou CB).
  - Case à cocher pour accepter le règlement intérieur.
- À la validation : envoi automatique d'un email de confirmation + mise à jour du statut.

---

4. Prochaines Actions

1. **Valider la liste des parties prenantes** avec 2-3 membres de l'association.
2. **Choisir 1 processus clé** (ex : renouvellement d'adhésion) et le cartographier ensemble.
3. **Remplir la matrice Impact/Effort** pour prioriser.

Question pour vous, Nicolas :
Souhaitez-vous commencer par affiner la liste des parties prenantes et leurs besoins, ou préférez-vous cartographier directement un processus type (ex : gestion des événements) ? Je peux vous fournir un template prêt à l'emploi pour l'un ou l'autre.