

## 007 – Suivi & Collaboration

Projet : Refonte du SI DIGICHEESE

### 7.1. Objectif

Décrire les mécanismes de **suivi**, de **communication** et de **coordination** permettant à l'équipe de piloter l'avancement, de sécuriser la qualité et d'assurer la traçabilité du projet.

Ce chapitre formalise le “cadre de travail” : outils, rituels, règles de développement et preuves associées. (Version entreprise : orientée gouvernance + auditabilité).

Référence : objectifs et éléments présents dans la version initiale. :contentReference[oaicite:1]{index=1}

### 7.2. Outils projet (et usage attendu)

Outil	Usage	Livrables / preuves attendues
Jira	Pilotage opérationnel : backlog, sprint, avancement, priorités (MoSCoW), critères d'acceptation	tickets à jour, statut conforme, sprint board
Confluence	Documentation officielle : cadrage, règles, décisions, qualité, itération	pages structurées, décisions tracées
Git (GitHub)	Gestion de versions : branches, PR, historique, revue de code	branches par ticket, PR, commits liés
Swagger (OpenAPI)	Validation fonctionnelle et démonstration des endpoints	endpoints testables, schémas, exemples
Visual Studio Code	Développement	config projet, extensions, exécution

		locale
phpMyAdmin	Administration DB (selon environnement)	création/contrôle tables, vérifs données
Docker / Docker Compose	Reproductibilité environnement (API + DB)	lancement “one command”, cohérence équipe

Alignement avec le document initial : Jira, Confluence, Git, Swagger, VS Code, phpMyAdmin, Docker. :contentReference[oaicite:2]{index=2}

---

### 7.3. Rituels Agile (cadence & attendus)

#### 7.3.1. Sprint Planning (planification)

- **Entrées** : backlog priorisé, capacité équipe (temps), dépendances.
- **Sorties** :
  - objectif de sprint explicite,
  - liste de tickets engagés,
  - critères d’acceptation validés,
  - risques/points d’attention identifiés.

#### 7.3.2. Daily meeting (suivi)

Format standard : **Avancement / Blocages / Actions du jour**

- Focus : synchronisation rapide, levée des blocages, protection du sprint.
- Les décisions et changements importants sont reportés dans Jira/Confluence.

#### 7.3.3. Sprint Review (revue)

- Démonstration basée sur :
  - Swagger (fonctionnel),
  - exécution de tests (preuve qualité),
  - tickets “Done” et critères d’acceptation atteints.
- Les éléments “non livrés” sont explicitement tracés : report / re-priorisation.

#### 7.3.4. Sprint Retrospective (amélioration continue)

- Analyse structurée : **process, outillage, qualité, collaboration, documentation**

- Sortie : **plan d'action** pour le sprint suivant (actions concrètes, owner, échéance)
- 

## 7.4. Règles de développement (workflow Git)

### 7.4.1. Branching & intégration

- **1 branche par ticket** (nommage recommandé : **feature/DGD-xx-...**).
- **Pull Request obligatoire** pour intégrer sur la branche cible.
- **Revue de code avant fusion** (au moins 1 pair).
- **Interdiction de commit direct sur la branche principale** (protection).

Alignment avec les règles présentes dans la version initiale. :contentReference[oaicite:3]{index=3}

### 7.4.2. Definition of Done (DoD) minimale (appliquée à chaque ticket)

Un ticket est considéré “Done” si :

- critères d’acceptation atteints,
  - endpoint(s) testables via Swagger,
  - tests exécutés localement (et CI si disponible),
  - code conforme aux conventions (lint),
  - documentation mise à jour si impact.
- 

## 7.5. Traçabilité (auditabilité)

Chaque ticket Jira doit :

- référencer une User Story / Epic,
- contenir des **critères d’acceptation**,
- porter une priorité MoSCoW,
- être relié à la documentation (Confluence) si le ticket modifie le fonctionnel/technique,
- être relié au code (branche/PR/commit).

Alignment avec les principes posés dans la version initiale. :contentReference[oaicite:4]{index=4}

---

## 7.6. Communication & décisions

- échanges réguliers (daily + points ad hoc si blocage),
  - décisions documentées (Confluence : “Décisions projet”),
  - conventions partagées (Confluence : page “Conventions API / DEV”).
-

## **7.7. Synthèse**

Le dispositif de suivi garantit :

- coordination et transparence,
- traçabilité Jira ↔ code ↔ documentation,
- reproductibilité et qualité de livraison.

Alignment avec la synthèse initiale. :contentReference[oaicite:5]{index=5}

---