

## 008 – Qualité & Validation

Projet : Refonte du SI DIGICHEESE

---

### 8.1. Objectif

Garantir la qualité **technique** et **fonctionnelle** du système livré, en intégrant la qualité au cycle de développement (tests, validation Swagger, revue de code, respect des conventions).

Alignement avec l'objectif initial. :contentReference[oaicite:6]{index=6}

---

### 8.2. Stratégie qualité (principes)

La qualité repose sur :

- **Tests** (unitaires / intégration légère),
- **Validation Swagger** (testabilité fonctionnelle),
- **Revue de code**,
- **Respect des conventions** (structure, endpoints, erreurs, nommage),
- **Traçabilité** (critères d'acceptation → preuve).

Alignement avec la stratégie initiale. :contentReference[oaicite:7]{index=7}

---

### 8.3. Tests (périmètre & exigences)

#### 8.3.1. Couverture attendue (minimum)

Les tests couvrent :

- cas nominaux (CRUD),
- erreurs fonctionnelles (404, 400, 409 selon règles),
- validations (formats, champs obligatoires, valeurs).

Alignement avec le document initial. :contentReference[oaicite:8]{index=8}

#### 8.3.2. Exécution

- Exécution obligatoire avant validation d'un ticket (local, et CI si disponible).
- Les échecs de tests bloquent la mise en "Done".

#### 8.3.3. Preuves associées (Sprint Review)

Pour chaque lot livré :

- tests exécutés (résultat attendu : OK),
  - endpoints démontrables dans Swagger,
  - critères d'acceptation atteints.
- 

#### 8.4. Revue de code (quality gate)

Toute fonctionnalité :

- relue par un pair (PR),
- conforme aux standards projet,
- documentée si impact (tech/fonc).

Alignement avec l'intention initiale. :contentReference[oaicite:9]{index=9}

---

#### 8.5. Validation fonctionnelle (definition)

Une fonctionnalité est considérée validée si :

- elle respecte les critères d'acceptation,
- elle est testable via Swagger,
- elle correspond au backlog (périmètre + priorité).

Alignement avec le document initial. :contentReference[oaicite:10]{index=10}

---

#### 8.6. Qualité outillée (recommandations “entreprise”)

Cette section formalise les pratiques recommandées pour la suite (ou pour les options de la promo) :

- Lint & format : ruff (et/ou formatter) appliqué avant merge,
- Mesure de couverture : `pytest-cov` ,
- Automatisation CI : pipeline minimal (lint + tests) sur PR,
- “Quality gates” : échec CI = merge bloqué.

Remarque : ces éléments sont cohérents avec l'objectif d'industrialisation et de professionnalisation du cycle de dev.

---

#### 8.7. Synthèse

La qualité est un **processus continu intégré** au développement : tests + revue + validation Swagger + respect des conventions.

Alignement avec la synthèse initiale. :contentReference[oaicite:11]{index=11}

