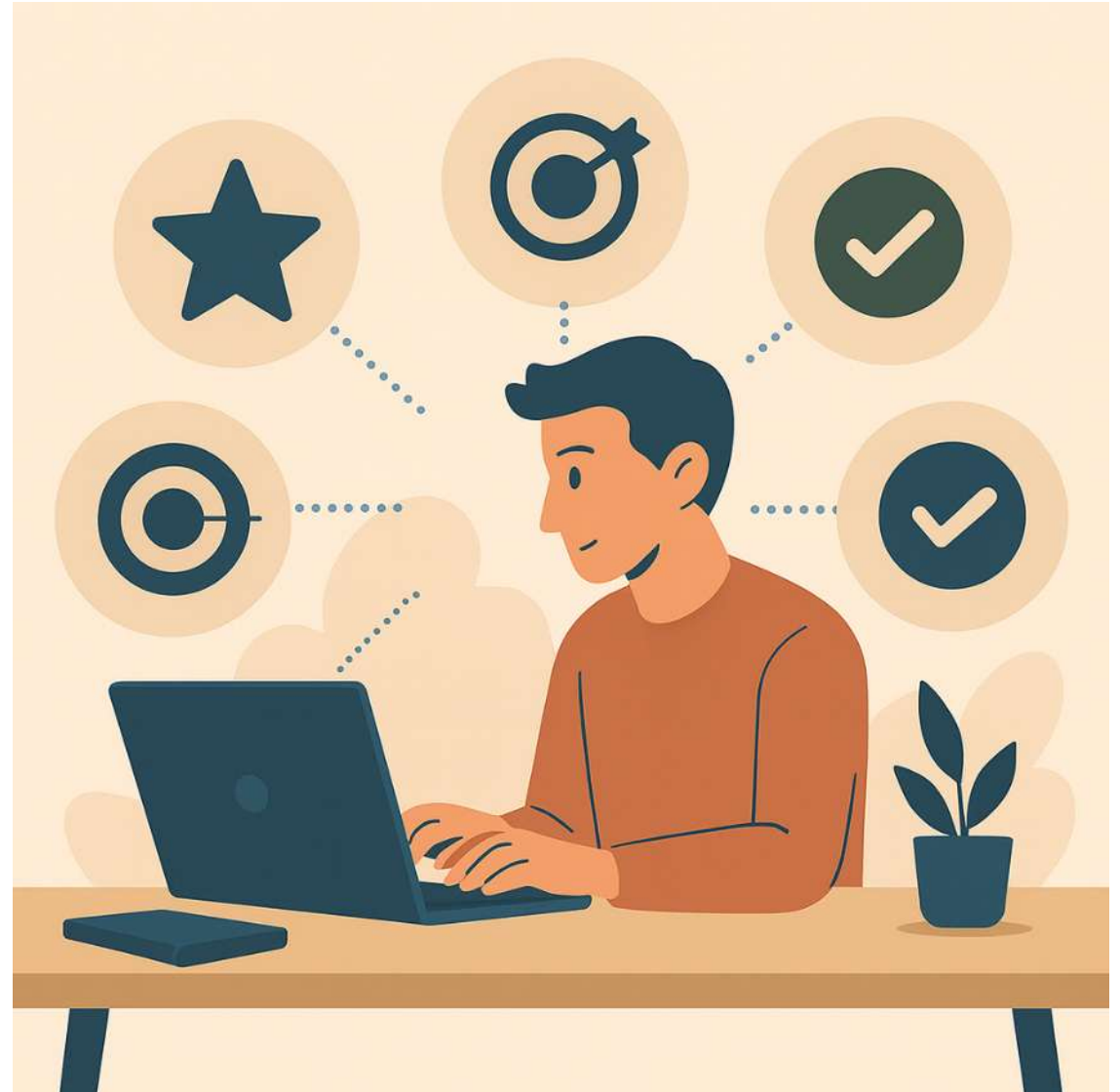


Chapitre 4 – QUALITÉ, LIVRABLES, OUTILS TECHNIQUES ET CLÔTURE

- Objectifs

- Identifier les dimensions de la qualité d'un projet informatique.
- Comprendre le rôle de Git dans la collaboration et la traçabilité.
- Comprendre les principes d'une chaîne CI/CD.
- Préparer les livrables et clôturer le projet en capitalisant sur l'expérience.

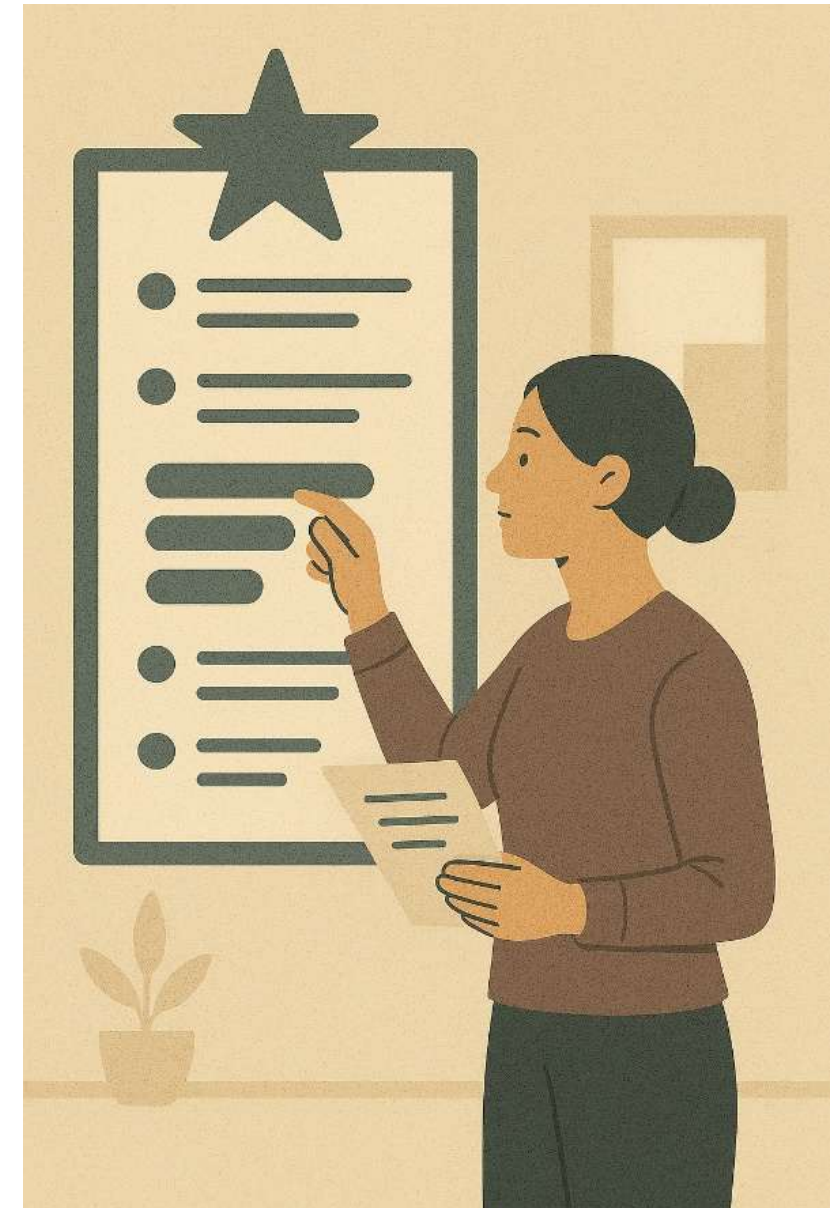


- En résumé

La livraison d'un projet informatique ne se limite pas à remettre du code au client ou aux utilisateurs. Le produit doit être testé afin de vérifier son bon fonctionnement, documenté pour faciliter son utilisation et sa maintenance, versionné pour garantir la traçabilité des évolutions et préparé pour pouvoir être déployé dans de bonnes conditions.

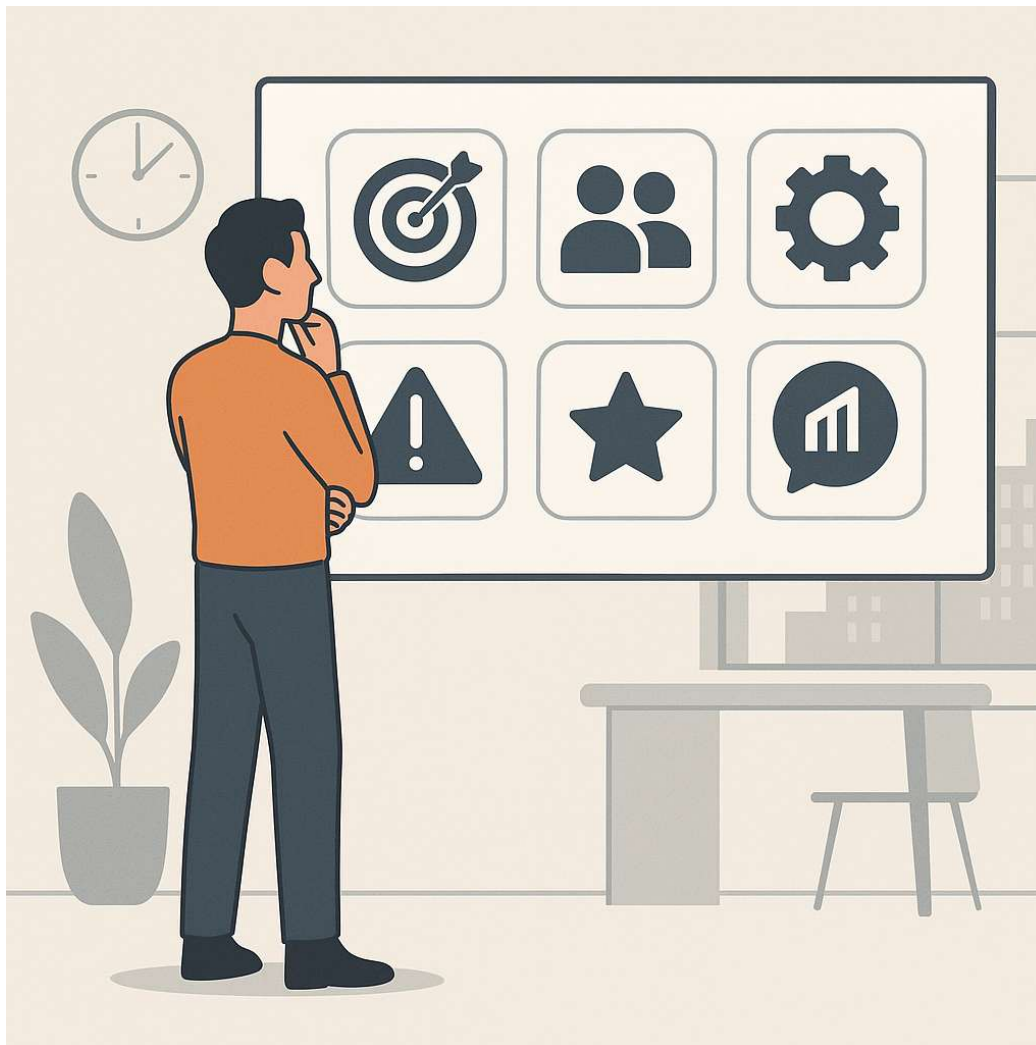
La clôture du projet permet ensuite de valider les résultats obtenus, de vérifier que les livrables correspondent aux objectifs initiaux et d'organiser le transfert des responsabilités vers les équipes qui assureront l'exploitation ou la maintenance.

Elle constitue aussi un moment important pour réaliser un retour d'expérience, identifier les réussites et les difficultés rencontrées, et conserver les enseignements utiles pour les projets futurs.



A – Qualité et stratégie de tests

La qualité se prépare tout au long du projet.

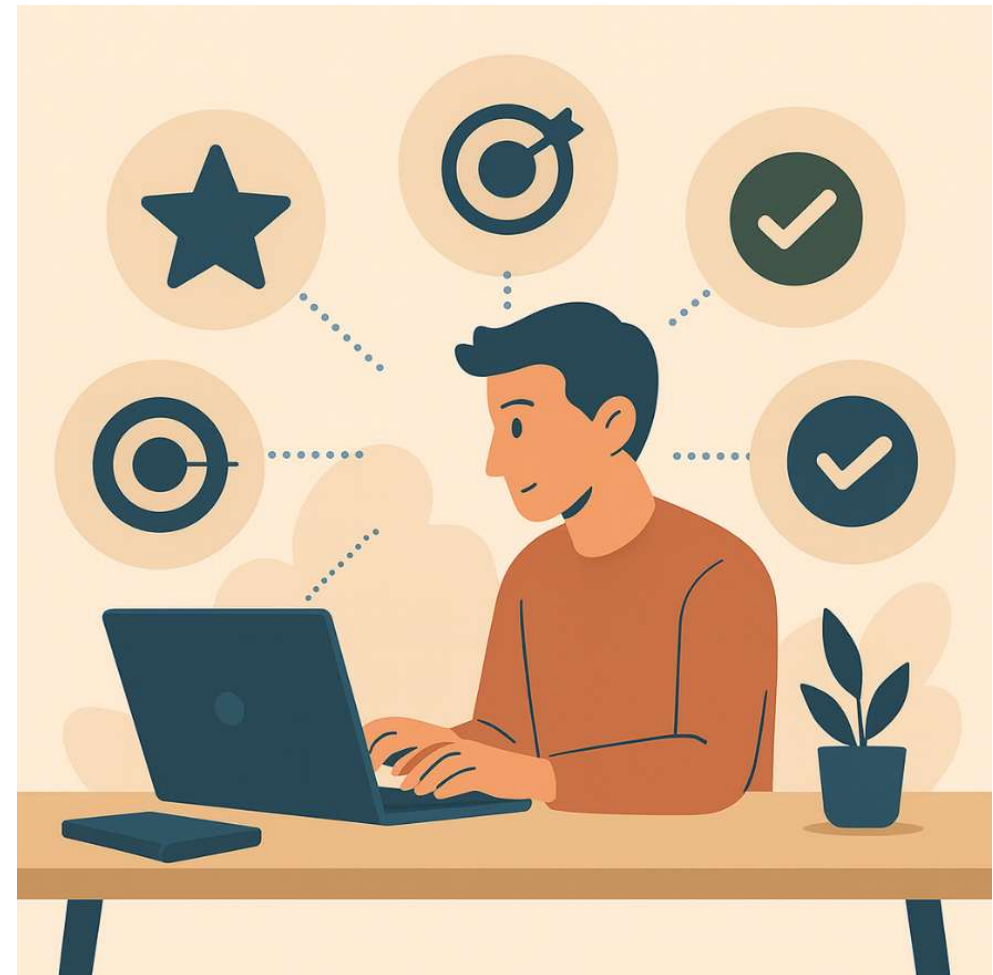


- Tests unitaires : vérifient une petite partie du code.
- Tests d'intégration : vérifient le fonctionnement entre plusieurs composants.
- Tests fonctionnels : vérifient le respect du besoin.
- Tests utilisateurs / recette : valident la solution du point de vue métier.
- Tests de performance ou sécurité lorsque le contexte l'exige.
- Définition de critères d'acceptation et d'une Definition of Done dans une équipe agile.

B – Livrables d'un projet informatique

Les livrables doivent permettre d'utiliser, maintenir et faire évoluer la solution. Leur liste dépend du projet mais doit être définie dès le cadrage.

- Code source et versions livrées.
- Documentation technique et architecture.
- Documentation utilisateur ou guides d'exploitation.
- Scripts ou procédures d'installation et de déploiement.
- Jeux de tests, rapports de recette et liste des anomalies connues.
- Configurations, schémas de données et informations nécessaires à la maintenance.



C – Git et gestion collaborative du code



Git est un système de gestion de versions qui conserve l'historique du code et facilite le travail à plusieurs. Il apporte de la traçabilité : qui a modifié quoi, quand et pourquoi.

- Repository : espace contenant le projet et son historique.
- Commit : enregistrement d'un ensemble cohérent de modifications.
- Branch : ligne de travail isolée permettant de développer sans perturber la version principale.
- Merge : intégration de modifications provenant de branches différentes.
- Pull Request / Merge Request : proposition de modification permettant la revue du code avant intégration.

D – CI/CD et automatisation de la livraison

La CI/CD désigne un ensemble de pratiques visant à automatiser l'intégration, les tests et la livraison.

Lorsqu'un développeur pousse du code, un pipeline peut lancer automatiquement des contrôles avant d'autoriser une mise en production.

- Continuous Integration : intégration fréquente du code et vérifications automatiques.
- Build : création d'une version exécutable ou déployable.
- Tests automatisés : vérification systématique de la qualité.
- Continuous Delivery : version toujours prête à être déployée.
- Continuous Deployment : déploiement automatique après validation des étapes du pipeline.



Précision : Schéma type : Code -> Git -> Build -> Tests -> Validation -> Déploiement.



Exemple : Une Merge Request déclenche automatiquement la compilation et les tests. Si un test échoue, la fusion est bloquée jusqu'à correction.

E – Clôture et retour d'expérience

La clôture officialise la fin du projet. Elle vérifie que les livrables sont acceptés, que les responsabilités sont transférées et que les documents nécessaires sont accessibles.



- Validation finale et recette.
- Transfert vers l'exploitation ou l'équipe de maintenance.
- Archivage des documents et fermeture des accès temporaires.
- Bilan des objectifs, délais, qualité et principaux écarts.
- Analyse des difficultés, réussites et décisions utiles.
- Actions d'amélioration pour les prochains projets.