

PROJET *

Production

2025/2026

Mastere Développement Fullstack

M1 - Groupe B

Dernière version : 06/11/25

PROJET POKEDEX



Sprint 0
Sprint planning

**Jour 1
Cadrage**
05/11

**Doc cadrage
Board**

RACI / SWOT
KPI / US

Sprint 1
Rituels SCRUM

**Jour 2
Prod & Maintenance**
06/11

**Comptes rendus
Plan de maintenance**

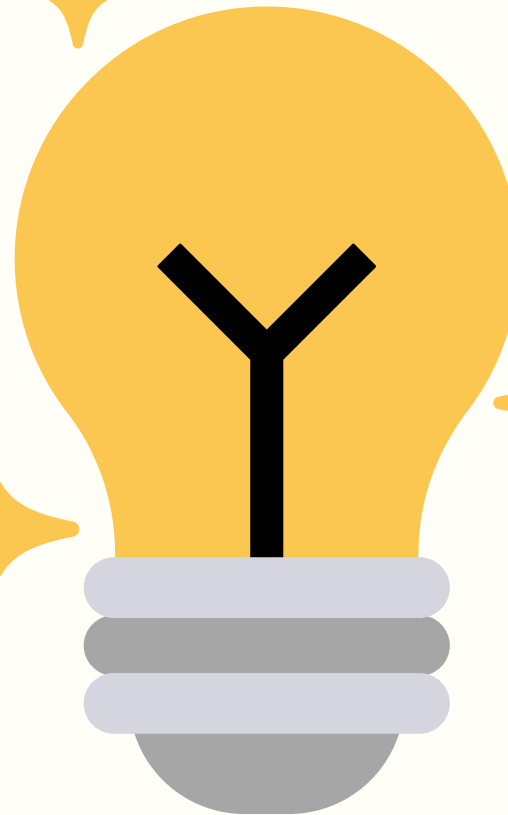
Daily - Review - Retro
Types de maintenance


Estimation des coûts
Planification

**Jour 3
Coûts & Pilotage**
19/12

Cas pratique

Ecrit ~1h



- 
1. Théories des maintenances
 2. DAY ONE SPRINT ONE
 3. Review et retro

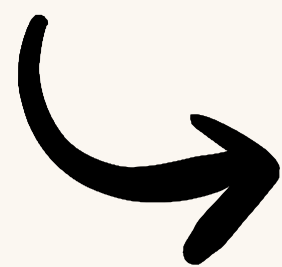
Selon la norme ISO/IEC 14764, la maintenance est :

La **modification** d'un produit logiciel **après sa livraison** pour :

- corriger des fautes
- améliorer les performances
- adapter le produit à un environnement modifié

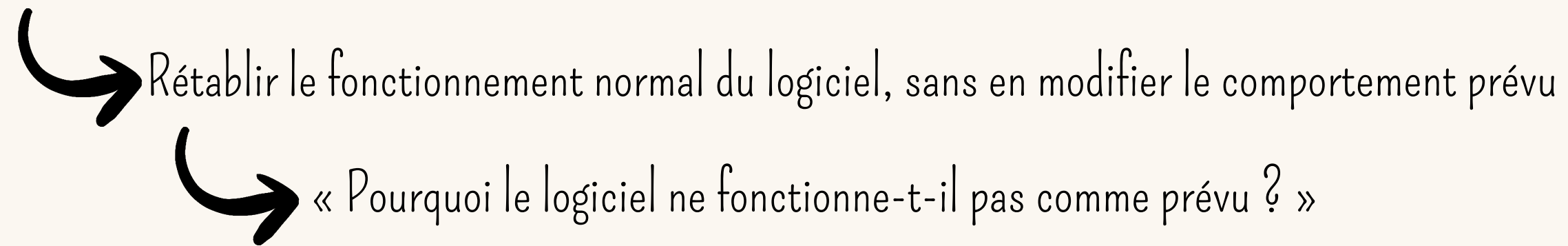
Même après la livraison, un logiciel suit un cycle de vie continu

- Livraison / Mise en production
- Phase d'exploitation
- Détection d'un besoin (bug, évolution, optimisation)
- Analyse et planification de la maintenance
- Implémentation et déploiement du correctif ou de la mise à jour



Un projet logiciel n'est jamais "terminé" : il évolue avec son environnement et ses utilisateurs

L'ensemble des actions visant à corriger les anomalies détectées dans le logiciel après sa mise en service

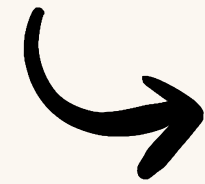


Comment ?

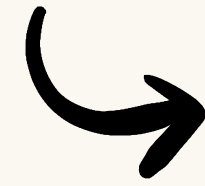
- Signalement du bug
- Analyse de la cause
- Priorisation selon la gravité (mineure, majeure, critique)
- Correction et livraison d'un patch

Exemples : Corriger un bug qui empêche la sauvegarde d'une partie dans ChronoQuest
Résoudre une erreur d'affichage d'une carte Pokémon

L'ensemble des actions réalisées avant l'apparition d'une panne ou d'un dysfonctionnement, dans le but de réduire les risques futurs



Améliorer la fiabilité et la sécurité du système à long terme.



« Que peut-on faire maintenant pour éviter un problème demain ? »

Comment ? Planification de tâches techniques dans le backlog
Surveillance continue (monitoring, alerting)
Mise en place de tests automatisés et d'outils d'audit de code.

Exemples : Nettoyer régulièrement la base de données pour éviter sa saturation
Mettre à jour les dépendances (framework, bibliothèques, serveurs)
Revoir les tests unitaires et les performances pour prévenir des bugs

Elle consiste à faire évoluer le logiciel pour l'adapter à de nouveaux besoins, technologies ou marchés.

→ Accroître la valeur métier et la satisfaction utilisateur.

→ « Comment améliorer le logiciel pour répondre à de nouveaux objectifs ? »

Comment ? Recueil et priorisation des besoins
Étude de faisabilité technique et budgétaire
Estimation des coûts et planification (sous forme d'US, Epics...).
Développement et livraison via des sprints

Exemples : Ajouter un mode multijoueur à ChronoQuest
Intégrer un système de succès pour motiver les joueurs
Refonte du design du Pokédex pour le rendre mobile-first
Migration du backend vers une architecture microservices

Organisation :

8h30 → 12h15 : Production - Mise à jour board

12h15 → 12h30 : Daily - Compte rendu

Attendus :

Screen board début de sprint, heure visible - Annexe

Screen board fin de journée, heure visible - Annexe

Compte rendu daily - Annexe



Daily review

Quand : Tous les jours

Qui : Dev team + SM (+ PO)

But : Synchroniser l'équipe

Déroulé : 15 minutes max

- Tâches d'hier
- Tâches d'aujourd'hui
- Obstacles / Difficultés



Sprint review

Quand : Fin de chaque sprint

Qui : PO + PP (+SM)

But : Montrer ce qui a été réaliser, valider

Déroulé

- PO rappelle objectifs du sprint
- Démo du produit
- Feedback PP

Organisation :

10 min : Préparation de la review : quelles US accomplies ?

30/40 min : Review avec les PO / Rédac de la doc pour la dev team

~30 min : Retrospective



Sprint retrospective

Quand : Après la sprint review

Qui : Dev team + SM (+ PO)

But : Améliorer la manière de travailler de l'équipe

Déroulé

- Point positifs
- Point négatifs
- Points d'amélioration

Attendus :

Compte rendu Review : feedback client - Annexe

Compte rendu Retro - Annexe

Cahier des charges

1 - Contexte et objectifs du projet

2 - Périmètre fonctionnel

Description globale des principales fonctionnalités, intégration des 1^{ers} maquettes possibles

3 - Contraintes techniques

Exigences techniques spécifiques du projet

4 - Livrables attendus

Livrables de fin de projet : prototype, v1, documentation, guide utilisateur...

5 - Critères d'acceptation

Plan de Management Projet

1 - Objectifs du PMP

Reprendre la formule, objectifs du document en interne

2 - Ressources techniques

Liste des ressources techniques et matérielles nécessaires (choisies) pour le projet, justifiées

3 - Ressources humaines

RACI, certains choix justifiés

“Le PO(A/R) supervise la réalisation des maquettes car il participe à leur élaboration avec le client (C), puis il les transmet à l'équipe de développement (I).”

4 - Management projet NON

5 - Planification NON

6 - Prévention des risques

SWOT, Matrice de criticité des risques, plan de prévention

7 - Suivi du budget NON

8 - Indicateurs de pilotage

2 KPI, justifiées. Si possible un screen du KPI après la production.



Annexes : Compte rendus réunions, screen

Plan de maintenance :

préventive (actions mises en oeuvre + US techniques si nécessaire)

évolutive (améliorations envisagée + US associées)

MERCI

