

TP Gestion de Projets — JIRA / GitHub Projects

Table des matières

- [Organisation du TP](#)
- [Énoncé du TP](#)
- [Objectifs pédagogiques](#)
 - [Gestion de projet](#)
 - [Retour sur expérience](#)
- [Configuration recommandée des outils](#)
 - [Option A — Jira](#)
 - [Option B — GitHub Projects](#)
 - [Recommandation](#)
- [Analyse de la collaboration](#)
- [Organisation des groupes](#)
- [Livrables attendus](#)
 - [Pour chaque groupe](#)

Organisation du TP

- Le travail se fait par groupe sauf demande particulière à cgermain@diginamic.fr
 - Ce TP est noté et remplace les notes de **2 QCM** et **1 TP** (du cours gestion de projet et du cours JIRA)
-

Énoncé du TP

- En faisant un « feed back » de votre travail en groupe, vous allez mettre en place en répondant aux objectifs pédagogiques attendus dans ce document,
 - Vous pourrez simuler des daily meeting en suivant la gestion de rôles,
 - Vous avez un canevas à suivre dans ce document.
-

Objectifs pédagogiques

Gestion de projet

- Décomposer un projet en **User Stories, tâches, critères d'acceptation**

- Organiser un **board agile** (Scrum/Kanban)
- Utiliser des outils modernes : **Jira** ou **GitHub Projects**
- Gérer le **temps**, la **communication**, la **répartition des rôles**

Retour sur expérience

- Analyser leur collaboration durant les **5 jours Python**
 - Identifier ce qui a fonctionné / ce qui a bloqué
 - Développer une posture d'**amélioration continue**
-

Configuration recommandée des outils

Option A — Jira

- Type de projet : **Scrum** (ou **Kanban** si tu préfères)
- Colonnes :
 - To Do
 - In Progress
 - Code Review
 - Done
- Champs à activer :
 - Description
 - Story points
 - Acceptance criteria
 - Assignee
- Burndown chart automatique

Option B — GitHub Projects

- Créer un **Project (v2)**
- Ajouter un **Backlog view**
- Ajouter un **Board Kanban**
- Champs recommandés :
 - Status
 - Priority
 - Assignees
 - Labels (frontend/backend/docs/bug...)

Recommandation

-  **Jira** si l'objectif est la gestion de projet professionnelle
 -  **GitHub** si l'objectif est la cohérence avec du code / Python
-

Analyse de la collaboration

- Communication
 - Répartition des rôles
 - Gestion du temps
 - Problèmes rencontrés
-

Organisation des groupes

- **2 à 4 personnes** par groupe
- Rôles à répartir :
 - Product Owner (cadrage, stories)
 - Scrum Master (animation, gestion du board)
 - Développeurs (tâches techniques, documentation)

Chaque rôle doit être tournant ou partagé si les groupes sont petits.

Livrables attendus

Pour chaque groupe

- Lien vers le projet **Jira** ou **GitHub** (inviter Christophe Germain : pseudo : **germain72**)
- Backlog complet (**user stories + tâches + estimations**)
- Board mis à jour (avec un **sprint fictif**)