

Document de Restitution - TP JIRA

1. Introduction

Dans le cadre de notre formation Lead Dev chez Diginamic Formation, nous avons réalisé ce TP lors du cours de gestion de projet. L'objectif était de nous familiariser avec les plateformes JIRA et Confluence d'Atlassian en créant et configurant un environnement projet complet, nous permettant d'explorer en profondeur ces outils essentiels dans l'écosystème de la gestion de projet moderne.

Les objectifs pédagogiques étaient de :

- Comprendre l'interface et les fonctionnalités de base de JIRA
- Configurer un projet et ses workflows
- Créer et organiser des tickets
- Lier JIRA et Confluence pour la documentation
- Gérer les utilisateurs et les permissions

Ce projet a été réalisé collectivement par une équipe de quatre étudiants :

- Sheerin Banu
- Ibrahima Barry
- Florian Marcot
- Lucas Préaux

2. Sommaire

[1. Introduction](#)

[2. Sommaire](#)

[3. Présentation du Projet Exemple](#)

[3.1 Description du cas d'usage choisi](#)

[3.2 Périmètre fonctionnel](#)

[3.3 Besoins et contraintes identifiés](#)

[4. Configuration de l'Environnement JIRA](#)

[4.1 Création du Projet](#)

[4.2 Paramétrage des Éléments de Base](#)

[4.3 Organisation des Sprints/Kanban](#)

[5. Intégration avec Confluence](#)

[5.1 Structure de l'espace documentation](#)

[5.2 Proposition d'arborescence pour notre projet d'API](#)

[5.3 Intégration avec JIRA](#)

[6. Gestion du Projet](#)

[6.1 Backlog et User Stories](#)

[6.2 Suivi et Reporting](#)

[7. Bonnes Pratiques et Recommandations](#)

[7.1 Bonnes pratiques identifiées](#)

[7.2 Pièges à éviter](#)

[8. Conclusion](#)

[8.1 Synthèse des Réalisations](#)

[8.2 Compétences Acquises](#)

[8.3 Points d'Amélioration Identifiés](#)

[8.4 Perspectives d'Évolution](#)

[9. Annexes](#)

[A. Captures d'Écran Complémentaires](#)

[B. Ressources et Références](#)

[C. Glossaire](#)

3. Présentation du Projet Exemple

3.1 Description du cas d'usage choisi

Le cas d'usage retenu pour le TP concerne le développement d'une API. Le projet s'inscrit dans un cadre où l'intégralité des diagrammes UML (use cases, classes, séquences, etc.) a déjà été réalisée en amont. Cela nous a permis de nous concentrer sur la création et la configuration d'un espace de projet dans JIRA et Confluence, en tirant parti de ces schémas pour structurer nos tâches et nos priorités.

3.2 Périmètre fonctionnel

Le projet vise à concevoir une API répondant aux besoins d'une application de gestion des utilisateurs. Les principales fonctionnalités incluent :

- La création, la modification, et la suppression d'utilisateurs.
- L'authentification sécurisée via des tokens JWT.
- La gestion des rôles et des permissions (utilisateur, administrateur).

3.3 Besoins et contraintes identifiés

1. Besoins techniques :

- Garantir la compatibilité avec les frameworks front-end (Angular ou React).
- Assurer la conformité avec les normes REST et les bonnes pratiques de développement.

2. Contraintes de gestion :

- Les tâches doivent être clairement définies et priorisées pour respecter les délais.
- L'équipe doit être en mesure de collaborer efficacement à distance en utilisant JIRA pour le suivi des tâches et Confluence pour centraliser la documentation.

3. Contraintes techniques :

- Intégration obligatoire avec une base de données SQL (MySQL).
- Limitation des temps de réponse des endpoints à moins de 200 ms.
- S'assurer que l'API est facilement extensible pour intégrer de nouvelles fonctionnalités à l'avenir.

4. Configuration de l'Environnement JIRA

4.1 Création du Projet

Pour notre projet, nous avons opté pour un projet de type "SCRUM" dans JIRA. Ce choix nous a donné accès à un ensemble de fonctionnalités spécifiquement adaptées à la méthodologie SCRUM, notamment des templates de tickets prédéfinis comprenant :

- Epic : Pour les grands ensembles fonctionnels
- Story : Pour les fonctionnalités utilisateur
- Bug : Pour le suivi des anomalies
- Tâche : Pour les activités génériques
- Sous-tâche : Pour décomposer les tickets en éléments plus petits

Organisation et Gestion des Sprints

La méthodologie SCRUM est intégrée au cœur du projet avec un backlog projet permettant de centraliser et prioriser l'ensemble des tickets. Ce backlog nous permet de créer et organiser des sprints en y affectant les tickets selon nos priorités. Une vue Chronologie est également disponible, offrant une planification calendaire des tickets pour une meilleure visibilité à long terme.

Le processus de démarrage d'un sprint est intuitif et s'effectue depuis la vue Backlog via le bouton "Démarrer un sprint". Une fois le sprint lancé, nous avons accès à différentes vues pour gérer notre travail :

Vue Tableau (Kanban)

Cette vue, accessible dès le démarrage d'un sprint, présente les tickets du sprint en cours dans un format kanban. Elle utilise un workflow initial simple (TO DO > EN COURS > FAIT) qui permet une gestion visuelle efficace de l'avancement des tâches par l'équipe.

Vue Liste

Une vue alternative sous forme de tableau permet de visualiser et manipuler toutes les tâches dans un format similaire à Excel, facilitant la gestion en masse des tickets et leur modification.

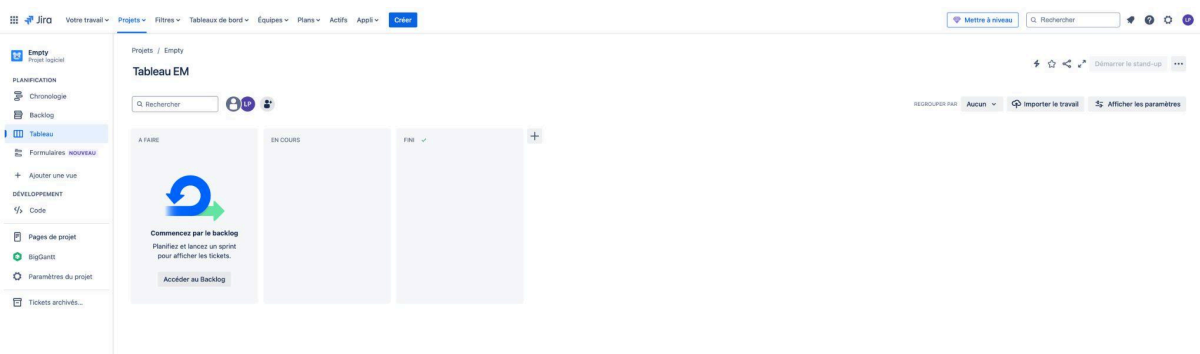
Fonctionnalités Avancées

Le projet SCRUM dans JIRA nous donne également accès à :

- Des rapports automatiques comme le burndown chart pour suivre l'avancement du sprint
- La possibilité de créer des formulaires personnalisés pour standardiser la création des tickets selon les besoins spécifiques de l'équipe ou de l'entreprise

Cette configuration nous permet d'appliquer efficacement les principes de la méthode SCRUM tout en bénéficiant des outils de suivi et de reporting intégrés à JIRA.

Voici la vue sur laquelle on arrive une fois qu'on a créé un nouveau projet :

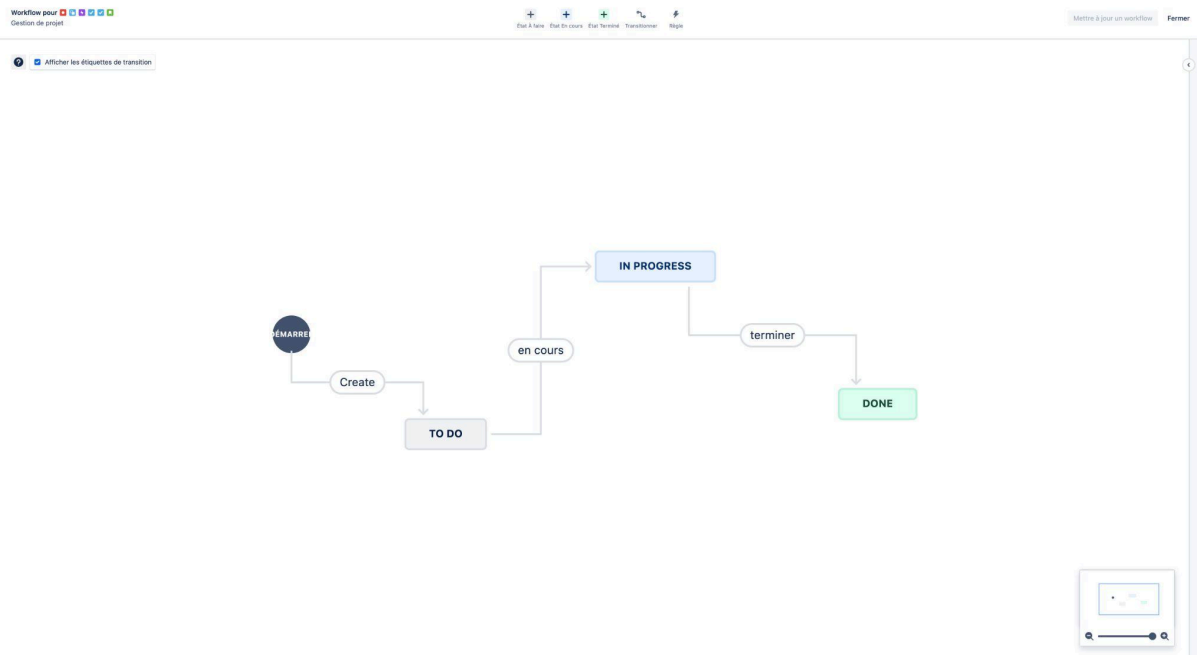


On remarque l'accès aux différentes vues du projet par le menu et que le tableau KANBAN sera accessible dès qu'on aura planifié et lancé un sprint.

4.2 Paramétrage des Éléments de Base

Étant donné que le projet est encore à ses débuts, nous nous sommes principalement concentrés sur la création d'Epics et de Stories. La création de "bugs" a été effectuée uniquement à titre informatif, afin de nous familiariser avec le processus et d'être prêts à les gérer lorsque cela deviendra nécessaire.

Nous n'avons pas non plus configuré de workflow particulier pour le projet et avons gardé le workflow par défaut qui nous convenait :



4.3 Organisation des Sprints/Kanban

Structure des sprints

▼

Tableau Sprint 2

15 déc – 29 déc

(6 tickets)

0

0

Démarrer un sprint

...

SCRUM-19	inviter un utilisateur à rejoindre une équipe	GESTION DES ÉQUIPES	À FAIRE	-	
SCRUM-20	accéder à la liste des membres d'une équipe	GESTION DES ÉQUIPES	À FAIRE	-	
SCRUM-35	modifier le rôle d'un membre	GESTION DES ÉQUIPES	À FAIRE	-	
SCRUM-38	supprimer un membre	GESTION DES ÉQUIPES	À FAIRE	-	
SCRUM-47	accéder à la page de description d'une équipe	GESTION DES ÉQUIPES	À FAIRE	-	
SCRUM-21	créer un projet	GESTION DES PROJETS	À FAIRE	-	

+ Créer un ticket

6 tickets

Estimation 0

▼

Tableau Sprint 3

29 déc – 12 janv.

(10 tickets)

0

0

Démarrer un sprint

...

SCRUM-23	rattacher un document à un projet	GESTION DES PROJETS	À FAIRE	-	
SCRUM-24	supprimer un document du projet	GESTION DES PROJETS	À FAIRE	-	
SCRUM-25	supprimer un document du projet	GESTION DES PROJETS	À FAIRE	-	
SCRUM-26	modifier un document du projet	GESTION DES PROJETS	À FAIRE	-	

- La durée du sprint est de 2 semaines
- Possible entre 1 et 4 semaines
- Durée fixe et constante
- Alignée sur le rythme et la vélocité de l'équipe

Phases Essentielles d'un Sprint

Avant le premier Sprint

Sprint Planning :

- Sélection collective des user stories
- Estimation des charges de travail
- Définition des objectifs du sprint
- Calcul de la vélocité prévisionnelle
- Engagement de l'équipe

Pendant le Sprint

Daily Stand-up (15 minutes max)

- Point quotidien
- Chaque membre répond à 3 questions :
 - Qu'ai-je fait hier ?
 - Que vais-je faire aujourd'hui ?
 - Ai-je des blocages ?
- Favorise la transparence
- Permet une synchronisation rapide

Fin du Sprint

Sprint Review :

- Démo du travail réalisé
- Présentation aux parties prenantes
- Validation des fonctionnalités
- Recueil des feedbacks

Rétrospective :

- Analyse collective du sprint
- Que s'est-il bien passé ?
- Quels sont les axes d'amélioration ?
- Actions concrètes à mettre en place

Préparation du Backlog du sprint suivant :

- Sélection des user stories du product backlog
- Décomposition en tâches
- Estimation en story points
- Respect de la capacité de l'équipe
- Priorisation claire

Points à prendre en compte lors de la préparation d'un Sprint

Critères d'Éligibilité d'une Story :

- Compréhension claire
- Critères d'acceptation définis
- Faisable dans le sprint
- Taille adaptée
- Valeur métier
- Techniquement réalisable

Gestion des Imprévus :

- Garder 10-15% de capacité pour :
- Corrections de bugs
- Tâches techniques
- Incidents

Le but est de rester **flexible sans déstabiliser le sprint**.

Métriques de Suivi

Plusieurs éléments sont à mesurer pour améliorer la qualité du produit (certains sont d'ailleurs mesurés automatiquement par Jira) :

- Vitesse de l'équipe
- Burndown chart
- Taux de complétion
- Nombre de stories livrées
- Qualité des livrables

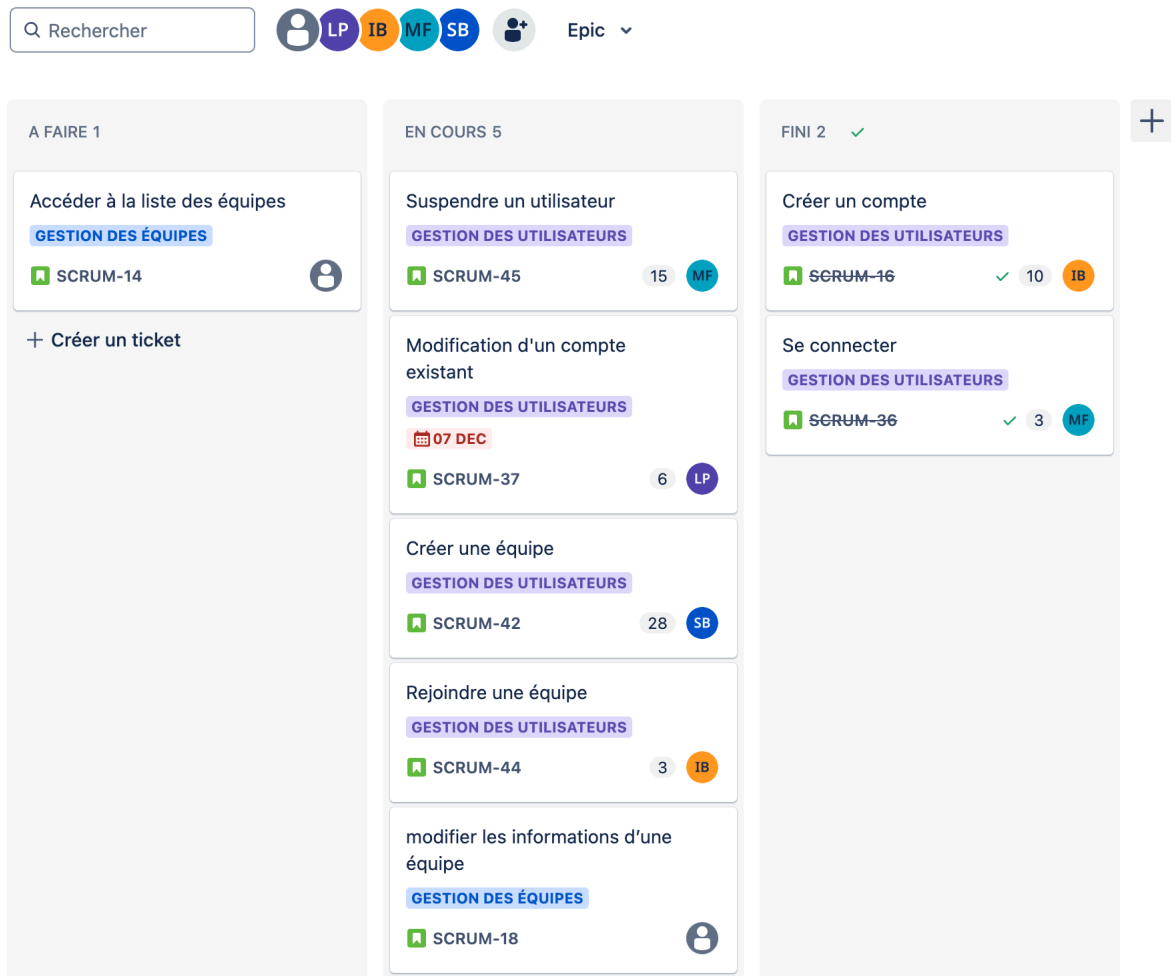
Tableau KANBAN

Le tableau kanban est créé à partir du workflow configuré dans le projet. Comme précisé plus haut nous avons gardé le workflow par défaut : A FAIRE > EN COURS > FAIT

Voici les tableaux sprint :

Projets / Gestion de projet

Tableau Sprint 1



5. Intégration avec Confluence

Dans le cadre de notre TP, nous avons créé une page principale dans Confluence pour notre projet, qui servirait à accueillir la description détaillée du projet. Bien que nous n'ayons pas eu le temps de développer une documentation complète, voici comment nous aurions pu organiser notre espace Confluence de manière efficace.

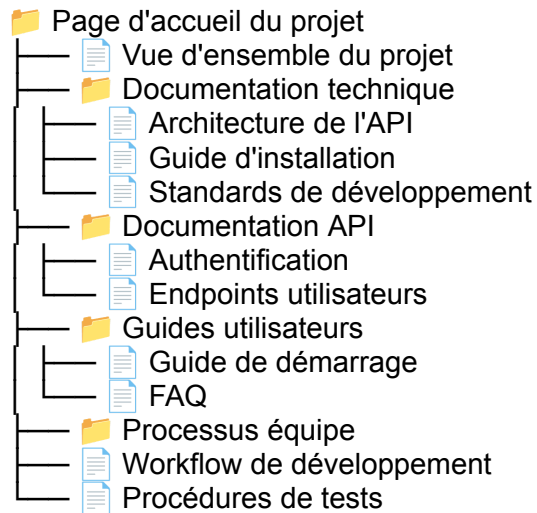
5.1 Structure de l'espace documentation

Confluence fonctionne avec une organisation hiérarchique des pages, permettant une navigation intuitive dans la documentation :

- Chaque espace peut contenir plusieurs pages
- Les pages peuvent avoir des sous-pages, créant ainsi une arborescence

- La navigation latérale permet de visualiser facilement cette hiérarchie
- Les pages peuvent être déplacées et réorganisées facilement par glisser-déposer

5.2 Proposition d'arborescence pour notre projet d'API



5.3 Intégration avec JIRA

Confluence s'intègre naturellement avec JIRA grâce à plusieurs fonctionnalités :

- Possibilité d'insérer des macros JIRA dans les pages Confluence
- Création de liens directs vers les tickets JIRA
- Affichage dynamique de l'état d'avancement des tickets
- Capacité à créer des tableaux de bord combinant informations JIRA et documentation

Cette structure, si elle avait été mise en place, nous aurait permis de :

- Centraliser toute la documentation du projet
- Faciliter l'accès aux informations pour tous les membres de l'équipe
- Maintenir une cohérence entre le code et la documentation
- Simplifier l'intégration de nouveaux membres dans l'équipe

Bien que nous n'ayons créé qu'une page principale dans le cadre du TP, cette expérience nous a permis de **comprendre le potentiel de Confluence comme outil de documentation** et son **intégration naturelle avec JIRA**.

6. Gestion du Projet

6.1 Backlog et User Stories

Organisation du backlog

Pour structurer le backlog, nous avons créé un Epic pour chaque diagramme de cas d'utilisation. Cette approche nous a permis de regrouper les tâches de manière cohérente et de maintenir une vue d'ensemble claire sur les fonctionnalités du projet.

Exemples de User Stories créées

Chaque cas d'utilisation a été décliné en une User Story correspondante. Par exemple :

- *"Créer un compte"* : En tant qu'utilisateur, je souhaite pouvoir créer un compte pour accéder aux fonctionnalités de l'application.
- *"Se connecter"* : En tant qu'utilisateur, je veux pouvoir me connecter à mon compte pour sécuriser mon accès.

Priorisation et estimation

Les tâches ont été priorisées selon une logique de cascade, en débutant par les fonctionnalités critiques. Nous avons choisi de travailler d'abord sur l'authentification (*se connecter*), puis sur la gestion des équipes (*créer une équipe* et *attribuer des membres*). Chaque tâche a été estimée en termes d'effort et de complexité, afin de planifier efficacement le travail et respecter les délais.

Backlog

Dans la vue Backlog nous retrouvons tous les tickets créés pour le projet, classés par sprint quand ils ont été associés ou dans le backlog global du projet si ce n'est pas le cas.

Voici le backlog actuel de notre projet :

Backlog

Q Rechercher

14

Epic

Importer le travail

Analyses

Afficher les paramètres

Tableau Sprint 1

2 déc. - 15 déc.

(8 tickets)

Terminer le sprint

10

3

15

6

28

3

-

SCRUM-16

Créer un compte

GESTION DES UTILIS...

TERMINÉ(E)

SCRUM-36

Se connecter

GESTION DES UTILIS...

TERMINÉ(E)

SCRUM-45

Suspendre un utilisateur

GESTION DES UTILIS...

EN COURS

SCRUM-37

Modification d'un compte existant

GESTION DES UTILIS...

EN COURS

SCRUM-42

Créer une équipe

GESTION DES UTILIS...

EN COURS

SCRUM-44

Rejoindre une équipe

GESTION DES UTILIS...

EN COURS

SCRUM-18

modifier les informations d'une équipe

GESTION DES ÉQUIPES

EN COURS

SCRUM-14

Accéder à la liste des équipes

GESTION DES ÉQUIPES

À FAIRE

Créer un ticket

Tableau Sprint 2

15 déc. - 29 déc.

(6 tickets)

Démarrer un sprint

-

-

-

-

-

-

SCRUM-19

inviter un utilisateur à rejoindre une équipe

GESTION DES ÉQUIPES

À FAIRE

SCRUM-20

accéder à la liste des membres d'une équipe

GESTION DES ÉQUIPES

À FAIRE

SCRUM-35

modifier le rôle d'un membre

GESTION DES ÉQUIPES

À FAIRE

SCRUM-38

supprimer un membre

GESTION DES ÉQUIPES

À FAIRE

SCRUM-47

accéder à la page de description d'une équipe

GESTION DES ÉQUIPES

À FAIRE

SCRUM-21

créer un projet

GESTION DES PROJETS

À FAIRE

Créer un ticket

Tableau Sprint 3

29 déc. - 12 janv.

(10 tickets)

Démarrer un sprint

-

-

-

-

-

-

-

-

-

SCRUM-23

rattacher un document à un projet

GESTION DES PROJETS

À FAIRE

SCRUM-24

supprimer un document du projet

GESTION DES PROJETS

À FAIRE

SCRUM-25

supprimer un document du projet

GESTION DES PROJETS

À FAIRE

SCRUM-26

modifier un document du projet

GESTION DES PROJETS

À FAIRE

SCRUM-27

consulter les commentaires

GESTION DES PROJETS

À FAIRE

SCRUM-28

répondre à un commentaire

GESTION DES PROJETS

À FAIRE

SCRUM-29

attacher un emoji à un commentaire

GESTION DES PROJETS

À FAIRE

SCRUM-30

mettre un lien dans le projet vers un dépôt distant

GESTION DES PROJETS

À FAIRE

SCRUM-31

saisir un commentaire sur le projet

GESTION DES PROJETS

À FAIRE

SCRUM-22

consulter un projet + voir la liste des projets

GESTION DES PROJETS

À FAIRE

Créer un ticket

Backlog

(22 tickets)

Plan dans un tableau blanc

Créer un sprint

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

SCRUM-32

voir la liste des utilisateurs

GESTION DES PROJETS

À FAIRE

SCRUM-33

inviter un utilisateur

GESTION DES PROJETS

À FAIRE

SCRUM-34

attribuer un rôle à un utilisateur

GESTION DES PROJETS

À FAIRE

SCRUM-49

consulter un projet + voir la liste des projets

GESTION DES SPRINTS

À FAIRE

SCRUM-50

créer/modifier/supprimer les user stories

GESTION DU BACKLOG

À FAIRE

SCRUM-61

consulter le burndown chart

GESTION DES SPRINTS

À FAIRE

SCRUM-52

consulter la liste des événements Sprint

GESTION DES SPRINTS

À FAIRE

SCRUM-53

consulter/modifier/supprimer une tâche

GESTION DU BACKLOG

À FAIRE

SCRUM-54

consulter le backlog du sprint et les indicateurs

GESTION DES SPRINTS

À FAIRE

SCRUM-55

consulter le backlog du projet

GESTION DU BACKLOG

À FAIRE

SCRUM-56

consulter une fiche de bug

GESTION DU BACKLOG

À FAIRE

SCRUM-57

attacher une fiche de bug à une user story

GESTION DU BACKLOG

À FAIRE

SCRUM-58

consulter le tableau des tâches

GESTION DES SPRINTS

À FAIRE

SCRUM-59

lister les bugs

GESTION DU BACKLOG

À FAIRE

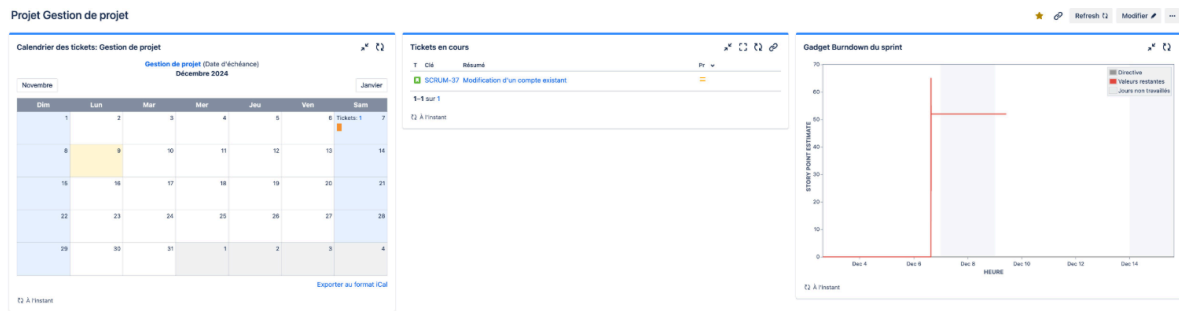
6.2 Suivi et Reporting

Tableaux de bord créés

Les tableaux de bord JIRA permettent de centraliser certaines informations du projet au même endroit pour pouvoir y accéder facilement à un moment donné. L'avantage des tableaux de bord est qu'ils peuvent être personnels ou partagés avec tout ou une partie de l'équipe. Ainsi on peut facilement créer des vues personnalisées en fonction de notre besoin ou des besoins de l'équipe.

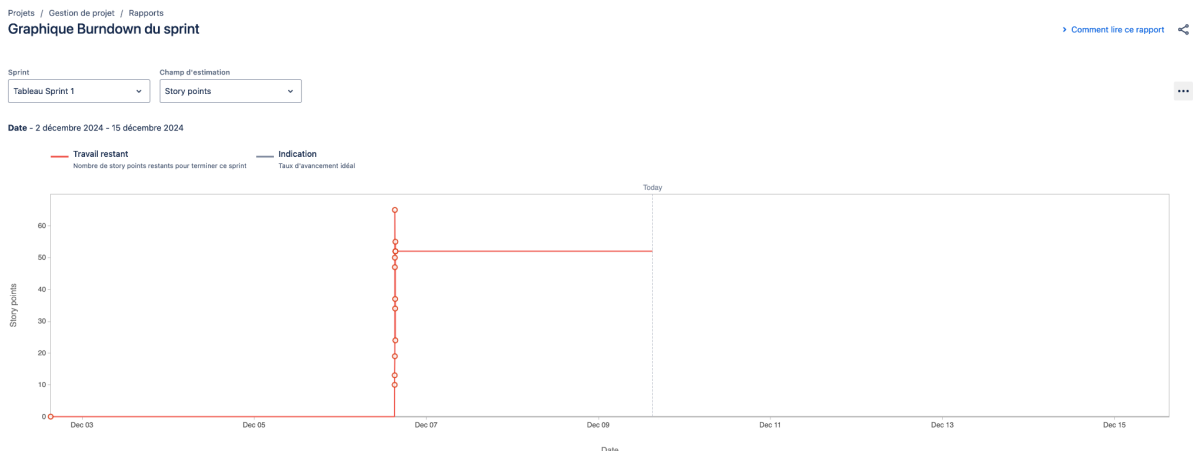
Nous avons créé un seul tableau de bord simple qui rassemble un calendrier qui présente les tickets planifiés/en cours, les tickets en cours et le burndown chart du projet.

Le voici :



Rapports

Nous n'avons pas configuré de rapports ou utilisé ceux pré-générés par JIRA. Par exemple le burndown chart est générée automatiquement par JIRA, seulement les tests ont été réalisés sur une journée et malgré la planification des ticket ou les changement des "actual start date" et "actual end date", JIRA considérait toujours le ticket ajouter et réaliser le jour des tests. Ce qui nous donne un graph peu représentatif :

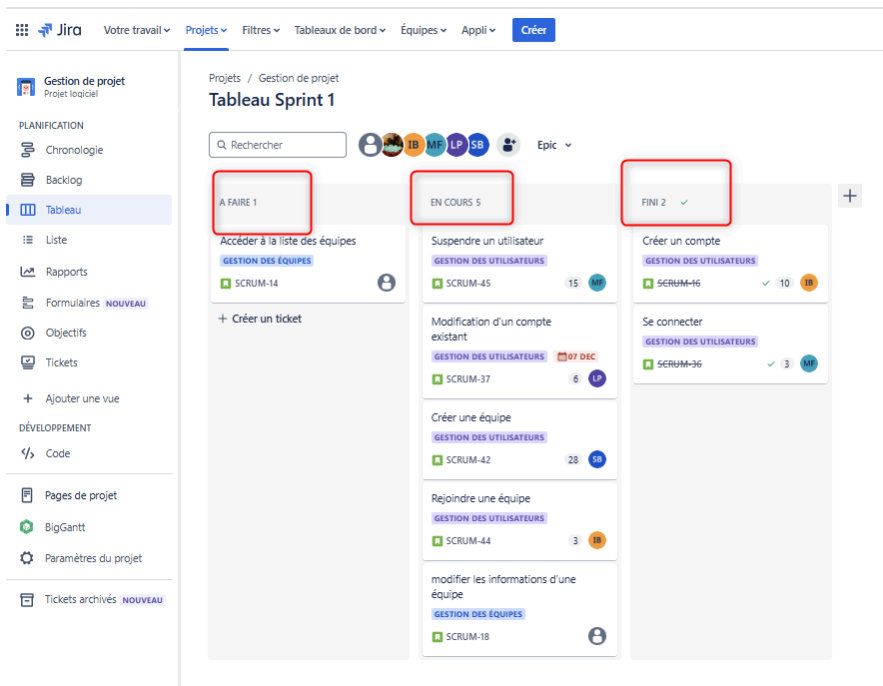


7. Bonnes Pratiques et Recommandations

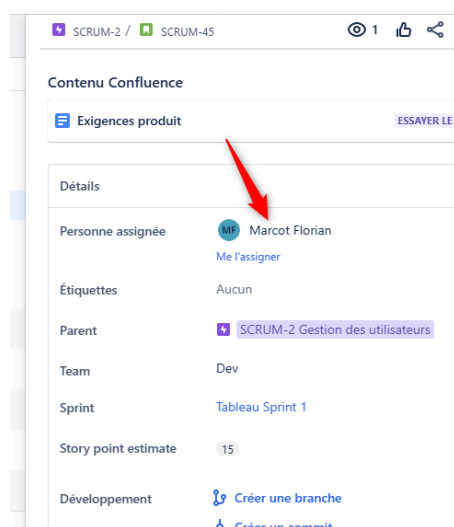
7.1 Bonnes pratiques identifiées

Parmi les bonnes pratique identifiées nous avons :

- la création d'un tableau Kanban adapté à notre méthodologie de travail avec des colonnes claires représentant les étapes du workflow(à faire, en cours, fini)

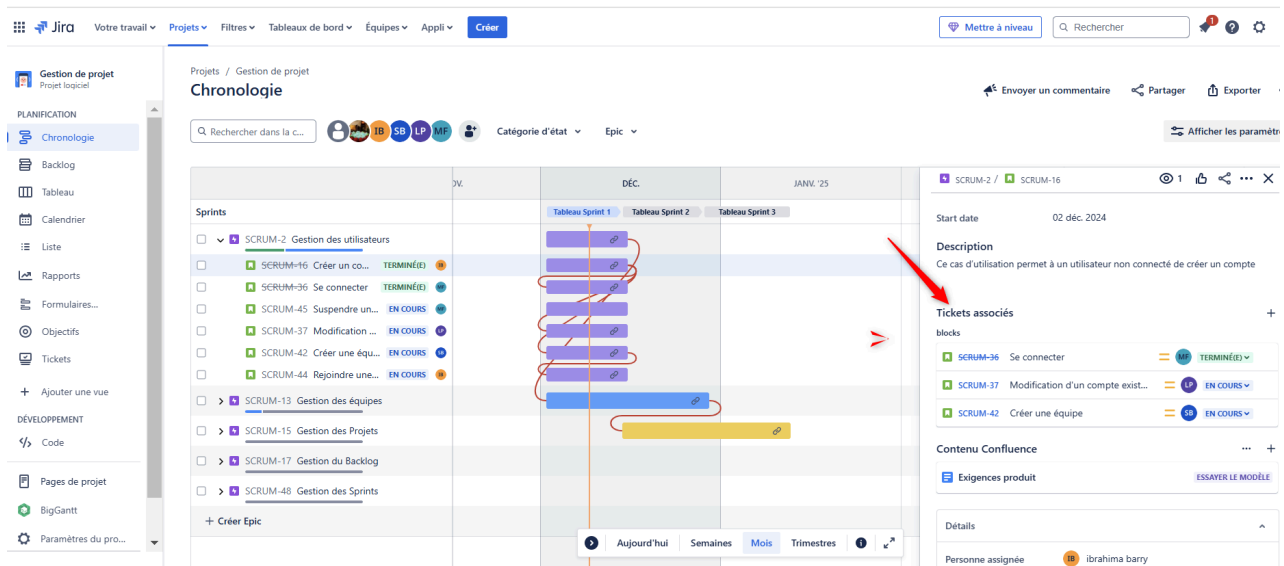


- Établir des catégories de tickets bien définies:
 - Epics
 - User stories
 - Tâches
 - Sous-tâche
 - Bugs
- Création et gestion des user stories
 - Découper les user stories en tâches concrètes et réalisables
 - Assignment et Estimation des Tâches
- Attribuer clairement chaque ticket à un membre de l'équipe

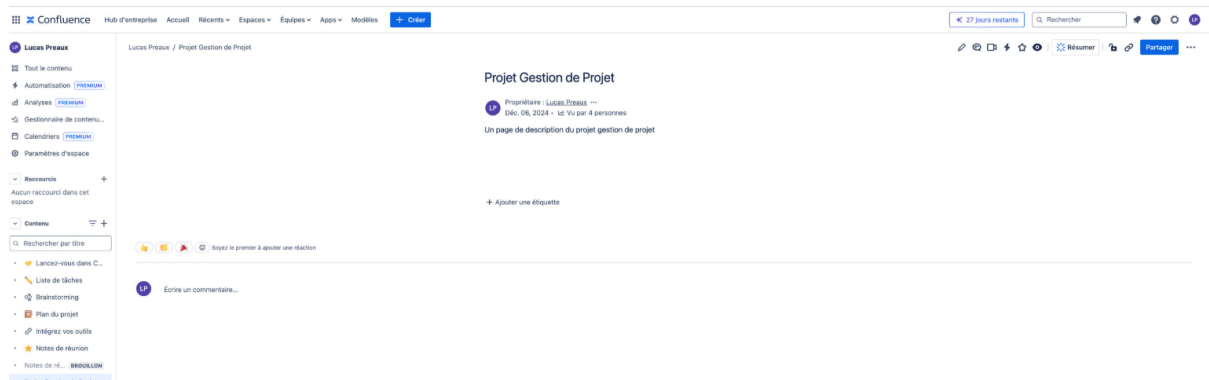


- Utiliser le système de points ou d'estimation de charge (points d'effort)
- Respecter les capacités et compétences de chaque membre
- Équilibrer la charge de travail entre les développeurs (membres)
- Communication et suivi

- Mettre à jour régulièrement les tickets (statut, progression)
- Utiliser les commentaires pour :
 - Partager des détails techniques
 - Mentionner des blocages
 - Demander des clarifications
- Ajouter des étiquettes (tags) pour faciliter le filtrage et la recherche
- Gestion des Sprints
 - Définir des sprints de durée constante (2 semaines recommandées)
 - Organiser des réunions régulières : sprint planning, daily sprint, sprint review, sprint rétrospective
- Qualité et Suivi
 - Intégrer des critères de "Definition of Done" clairs
 - Utiliser des workflows pour garantir la qualité
 - Ajouter des liens vers : documentation, ressources techniques et codes de références
 - Gérer les dépendances entre tickets :



- Gestion des priorités
 - Utiliser le champ de priorité (haute, moyenne basse)
 - Prioriser les tickets en fonction : des besoins métiers, de la valeur ajoutée et des contraintes techniques
- Réviser régulièrement la priorisation
- Bonnes Pratiques Collaboratives
 - Encourager la transparence
 - Favoriser la collaboration par des commentaires constructifs
 - Utiliser les @mentions pour impliquer directement un collègue
 - Partager les connaissances via les tickets
- Outils complémentaires:
 - Intégrer des outils externes si nécessaires: Git, Confluence, Slack :



- Améliorations continu
 - Analyser les métriques de vélocité
 - Identifier les goulots d'étranglement
 - Ajuster les processus régulièrement
 - Encourager les retours de l'équipe

Ces pratiques permettront une gestion de projet efficace, collaborative et transparente.

7.2 Pièges à éviter

- Pièges Organisationnels
 - Surcharge de tickets : Ne pas créer trop de tickets qui deviennent illisibles
 - Tickets trop vagues ou incomplets
 - Manque de clarté dans les descriptions
 - Absence de critères d'acceptance précis
 - Tickets sans définition claire de "Done"
- Erreurs de Gestion des Tâches
 - Surdécoupage ou sous-découpage des user stories
 - Estimation irréaliste des charges de travail
 - Non-respect des capacités individuelles
 - Assignment de trop nombreuses tâches simultanées
 - Ignorance des compétences spécifiques des membres
- Problèmes de Communication
 - Communication uniquement via les tickets
 - Manque de mise à jour régulière
 - Communication trop formelle ou trop technique
 - Absence de contexte dans les commentaires
 - Non-utilisation des @mentions
- Difficultés Méthodologiques
 - Rigidité excessive du workflow
 - Sauts d'étapes dans le processus
 - Non-respect des rituels agiles
 - Planning des sprints trop ambitieux
 - Absence de rétrospective

- Problèmes Techniques
 - Mauvaise utilisation des étiquettes
 - Configurations de tableau inadaptées
 - Non-utilisation des filtres et vues
 - Paramètres de notifications inappropriés
 - Intégrations mal configurées
- Erreurs de Suivi
 - Pas de vision globale du projet
 - Absence d'indicateurs de performance
 - Non-tracking des dépendances
 - Négligence des blocages
 - Manque de priorisation claire
- Problèmes Collaboratifs
 - Manque de transparence
 - Cultures du blâme
 - Cloisonnement des informations
- Gestion des Priorités
 - Changements constants de priorités
 - Dispersion des efforts
 - Sur-priorisation de tâches mineures
 - Négligence des tâches techniques
- Aspects Humains
 - Surcharge de travail
 - Démotivation par micromanagement
 - Communication négative
 - Absence de reconnaissance
 - Non-prise en compte du bien-être
- Anti-Patterns Agiles
 - Réunions improductives
 - Daily stand-up trop longs
 - Rétrospectives formelles
 - Absence d'amélioration

8. Conclusion

Ce projet de mise en pratique sur les outils Jira et Confluence a été une expérience enrichissante qui nous a permis de comprendre les fondamentaux de la gestion de projet agile dans un contexte technique.

8.1 Synthèse des Réalisations

Nous avons configuré un environnement projet complet, en nous appuyant sur un cas d'usage concret : le développement d'une API pour une application de gestion des utilisateurs. Nous avons structuré le backlog autour des besoins métiers, organisé des sprints en suivant la méthodologie SCRUM, et lié efficacement Jira et Confluence pour

centraliser les informations. L'utilisation des workflows, des user stories, et des outils de reporting a permis une gestion claire et organisée des tâches.

8.2 Compétences Acquises

- Maîtrise des fonctionnalités principales de Jira (création de projets, workflows, tickets).
- Structuration et gestion de backlog, sprints et tableaux Kanban pour une approche agile.
- Utilisation de Confluence pour la documentation et son intégration avec Jira.
- Collaboration en équipe à travers une gestion efficace des permissions, des rôles, et des priorités.

8.3 Points d'Amélioration Identifiés

- Bien pensé ses workflows pour mieux s'adapter à la taille du projet.
- Optimiser l'estimation des tâches et mieux équilibrer la charge de travail entre les membres de l'équipe.
- Accroître l'utilisation des rapports et des tableaux de bord pour améliorer le suivi des performances.

8.4 Perspectives d'Évolution

À l'avenir, cette expérience peut être étendue en explorant d'autres outils complémentaires comme des intégrations CI/CD avec Jenkins et en appliquant les méthodologies agiles à des projets plus complexes. Elle a également renforcé notre capacité à travailler dans un environnement collaboratif, une compétence clé pour nos futurs projets professionnels.

En conclusion, cette expérience nous a permis de **mieux appréhender les enjeux pratiques et méthodologiques de la gestion de projet moderne**, tout en **consolidant nos compétences techniques et organisationnelles**. Ce TP constitue une base solide pour appliquer ces apprentissages à des projets réels et évoluer dans nos rôles de Lead Developers.

9. Annexes

A. Captures d'Écran Complémentaires

1.exemple d'epic(Gestion des utilisateurs)

Projets / Gestion de projet / SCRUM-2

Gestion des utilisateurs

+ Ajouter @ Apps

Description

Modifier la description

Tickets enfant

Progression : 33%

Organiser par

SCRUM-16

Créer un compte

10

TERMINÉ(E)

SCRUM-36

Se connecter

3

TERMINÉ(E)

SCRUM-45

Suspendre un utilisateur

15

EN COURS

SCRUM-37

Modification d'un compte existant

6

EN COURS

SCRUM-42

Créer une équipe

28

EN COURS

SCRUM-44

Rejoindre une équipe

3

EN COURS

Tickets associés

blocks

SCRUM-13

Gestion des équipes

À FAIRE

Contenu Confluence

Plan de projet

ESSAYER LE MODÈLE

Activité

Afficher : Tout Commentaires Historique

Les plus récents d'abord

LP

Ajouter un commentaire...

Joli !

Besoin d'aide ?

C'est bloqué...

Pourriez-vous clarifier... ?

C'est sur la bonne voie

Conseil de pro : appuyez sur M pour commenter

À faire Actions

Détails

Personne assignée

Non assigné

Me l'assigner

Étiquettes

Aucun

Team

Aucun

Start date

Aucun

Date d'échéance

Aucun

Développement

Créer une branche

Créer un commit

Rapporteur

Lucas Preaux

Automation

Rule executions

Project overview

Link to share status and updates

Création il y a 3 jours

Mis à jour il y a 3 jours

Configurer

2.exemple de définition d'une user story (Créer un compte)

Projets / Gestion de projet / SCRUM-2 / SCRUM-16

Créer un compte

+ Ajouter @ Apps

Actual start

Aucun

Actual end

Aucun

Date d'échéance

Aucun

Start date

02 déc. 2024

Description

Ce cas d'utilisation permet à un utilisateur non connecté de créer un compte

Tickets associés

blocks

SCRUM-36

Se connecter

TERMINÉ(E)

SCRUM-37

Modification d'un compte existant

EN COURS

SCRUM-42

Créer une équipe

EN COURS

Contenu Confluence

Exigences produit

ESSAYER LE MODÈLE

Activité

Afficher : Tout Commentaires Historique

Les plus récents d'abord

LP

Ajouter un commentaire...

Joli ! Besoin d'aide ? C'est bloqué... Pourriez-vous clarifier... ? C'est sur la bonne voie

Conseil de pro : appuyez sur M pour commenter

Terminé(e)

Terminé

Actions

Détails

Personne assignée

ibrahima barry

Me l'assigner

Étiquettes

Aucun

Parent

SCRUM-2 Gestion des utilisateurs

Team

Dev

Sprint

Tableau Sprint 1

Story point estimate

10

Développement

Créer une branche

Créer un commit

Rapporteur

Lucas Preaux

Automation

Rule executions

Configurer

Création il y a 3 jours

Mis à jour il y a 3 jours

Résolu il y a 3 jours

3.Tableau de bord du projet

Jira

Votre travail

Projets

Filtres

Tableaux de bord

Équipes

Plans

Actifs

Apps

Créer

Rechercher

Refresh

Modifier

Projet Gestion de projet

Calendrier des tickets: Gestion de projet

Gestion de projet (Date d'échéance)

Novembre

Decembre 2024

Janvier

Dim	Lun	Mar	Mer	Jeu	Ven	Sam
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	1	2	3	4

Exporter au format Ical

Tickets en cours

1 On Réinitial

SCRUM-37 Modification d'un compte existant

3-1 Sur 1

A l'instant

Gadget Burndown du sprint

10

80

60

40

20

0

Dec 4

Dec 6

Dec 8

Dec 10

Dec 12

Dec 14

HEURE

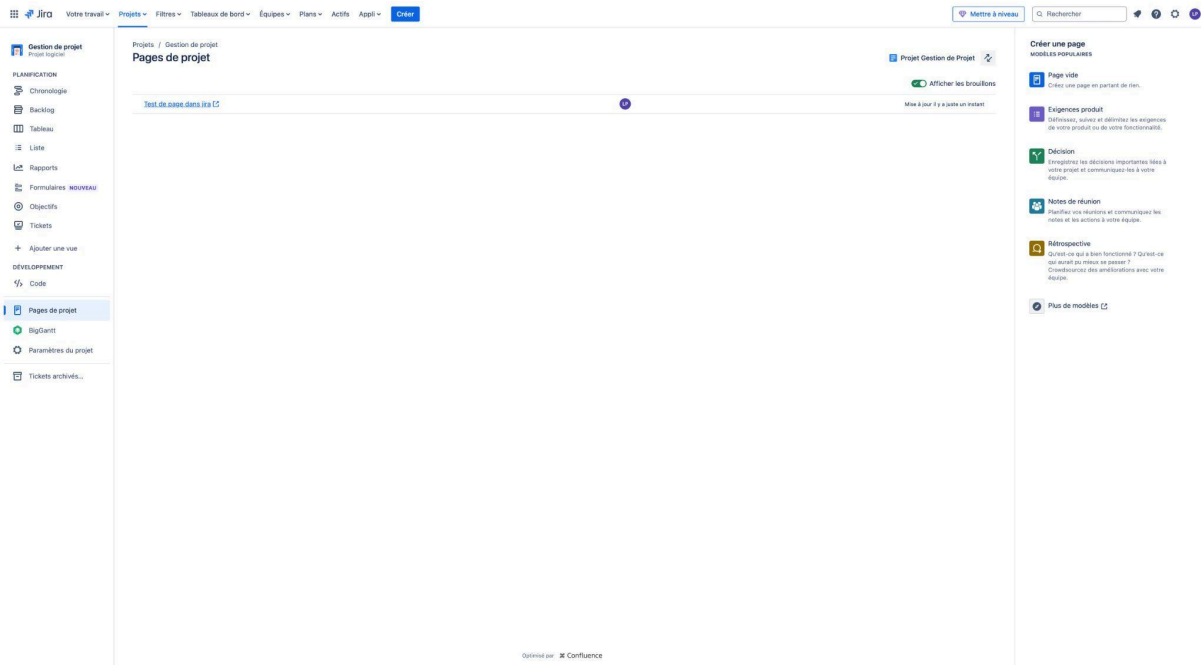
Directrice

valeurs restantes

jours non travaillés

A l'instant

4. Page associée au projet



B. Ressources et Références

- [Essentiel de JIRA](#)
-

C. Glossaire

Termes JIRA

- Backlog : Liste complète des tâches à réaliser dans le cadre d'un projet
- Epic : Ensemble de fonctionnalités regroupées qui forme une feature importante du projet
- Story (User Story) : Description d'une fonctionnalité du point de vue de l'utilisateur
- Sprint : Période de temps définie (généralement 2 semaines) pendant laquelle l'équipe s'engage à réaliser un ensemble de tâches
- Kanban : Méthode de visualisation du flux de travail sous forme de tableau avec des colonnes représentant les différentes étapes
- Workflow : Processus définissant les étapes par lesquelles passe un ticket (ex: À faire > En cours > Fait)
- Burndown Chart : Graphique montrant la quantité de travail restant à faire par rapport au temps

Termes Confluence

- Espace : Zone dédiée à un projet ou un département contenant un ensemble de pages
- Macro : Fonctionnalité permettant d'ajouter du contenu dynamique dans une page
- Page parent : Page principale contenant des sous-pages
- Template : Modèle de page pré-formaté pour uniformiser la documentation

Termes Méthodologie

- Scrum : Framework agile définissant des rôles, événements et artefacts pour la gestion de projet
- Daily Stand-up : Réunion quotidienne courte (15 min max) où chaque membre de l'équipe partage son avancement
- Sprint Planning : Réunion de planification au début de chaque sprint
- Sprint Review : Présentation des réalisations à la fin du sprint
- Sprint Retrospective : Réunion d'analyse et d'amélioration continue à la fin du sprint
- Vélocité : Mesure de la capacité de travail d'une équipe basée sur les sprints précédents

Termes Techniques

- API (Application Programming Interface) : Interface permettant à des applications de communiquer entre elles
- JWT (JSON Web Token) : Standard utilisé pour l'authentification et l'échange sécurisé d'informations
- REST : Style d'architecture pour les systèmes distribués, utilisé dans le développement web
- Endpoint : Point d'accès spécifique d'une API

Termes Gestion de Projet

- Definition of Done (DoD) : Critères définissant quand une tâche est considérée comme terminée
- Points d'effort : Unité relative utilisée pour estimer la complexité d'une tâche
- Product Backlog : Liste priorisée de toutes les fonctionnalités souhaitées pour le produit
- Sprint Backlog : Sous-ensemble du Product Backlog sélectionné pour un sprint spécifique