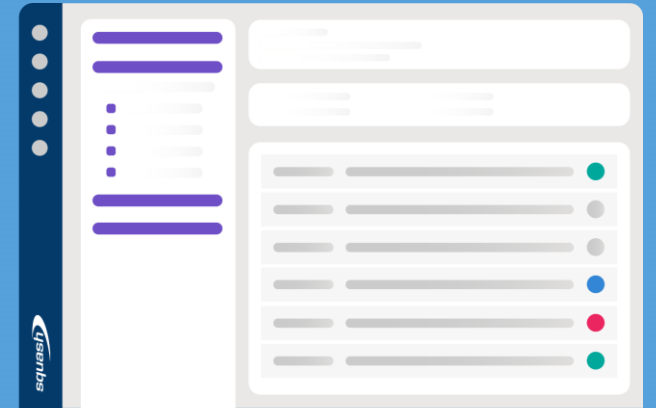


JIRA en contexte agile



Introduction

1. Les principes directeurs de l'Agile
2. Les méthodes Agiles : XP, Scrum, Kanban
3. Les membres d'une équipe Agile
4. Activités de test et rôles associés

I. JIRA en contexte Agile

1. Comprendre les fonctionnalités et concepts clés de JIRA Agile
 2. Recherches et filtres
- Cas pratique n°1 : Créer un projet JIRA et des tableaux, filtres et demandes pour ce projet.

II. Configuration des utilisateurs

1. Gérer les utilisateurs
2. Gérer les groupes d'utilisateurs

III. Paramétrage avancé

1. Gérer les systèmes d'autorisation
 2. Gérer les types de demande
 3. Gérer les flux de travaux
 4. Gérer les écrans
 5. Gérer les priorités
 6. Liens directs entre projet et entités paramétrées
- Cas pratique n°2 : Paramétrer un projet JIRA

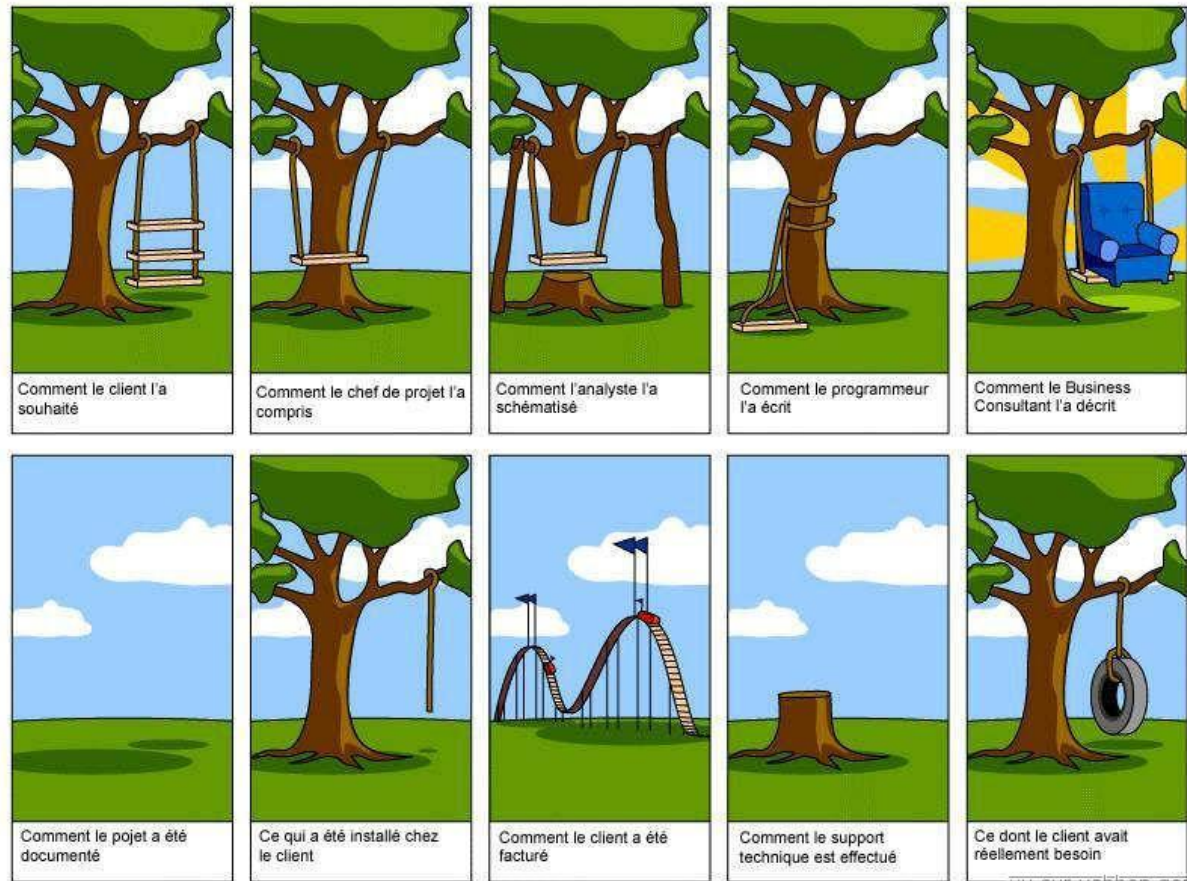
Introduction

- I. Les principes directeurs de l'Agile
- II. Les méthodes agiles : XP, Kanban, Scrum
- III. Activités de test et rôles associés

Les principes directeurs de l'Agile

Les principes directeurs de l'Agile

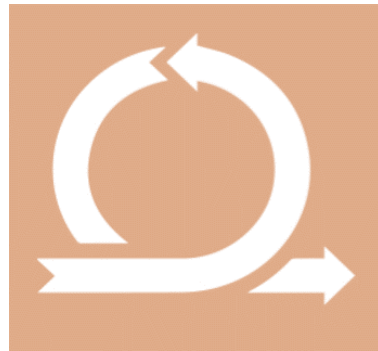
Echecs d'un projet ?



Qu'est ce que l'Agile ?

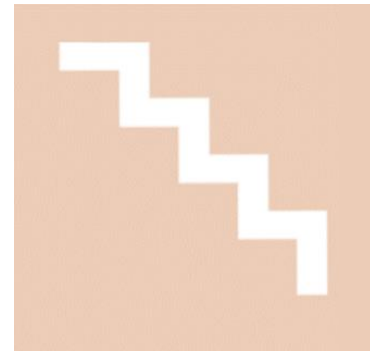
L'Agile est une approche de gestion de projet qui adopte une démarche **adaptative et itérative**.

APPROCHE AGILE



L'approche agile prend le contre-pied des approches traditionnelles **prédictives et séquentielles** de type cycle en V.

APPROCHES
TRADITIONNELLES

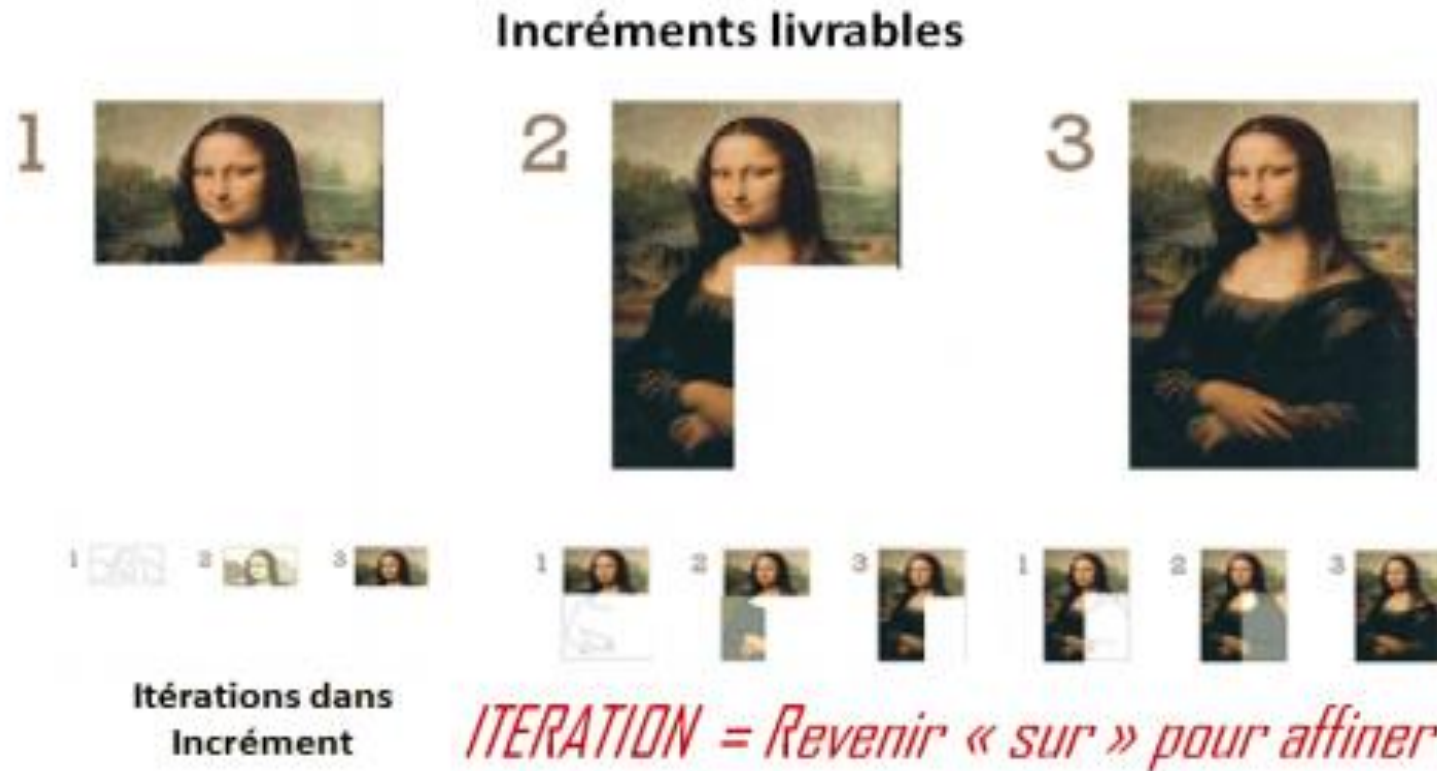


Historique :

- 1986 : premier modèle de développement itératif et incrémental de Barry W. Boehm.
- 2001 : le « Manifeste Agile », rédigé par 17 figures du développement logiciel qui se sont réunies afin de dégager de leurs méthodes respectives un ensemble de bonnes pratiques et bâtir un socle commun.

« *Nous découvrons comment mieux développer des logiciels par la pratique et en aidant les autres à le faire.* »

Manifeste Agile, 2001



« *Ces expériences nous ont amenés à valoriser :*



Nous reconnaissons la valeur des seconds éléments, mais privilégions les premiers. »

« **Plus que** » ne veut pas dire « **au lieu de** » ! Les valeurs énoncées ne sont pas une remise en cause totale des approches « traditionnelles ». Le risque est sinon de tomber dans les idées reçues selon lesquelles dans un projet agile il n'y a pas de spécifications, d'outils ou même de contrat...

Le Manifeste Agile – 12 principes

1

Notre plus haute priorité est de satisfaire le client en livrant rapidement et régulièrement des **fonctionnalités à grande valeur ajoutée**.

2

Accueillir positivement les changements de besoins, même tard dans le projet. Les processus agiles exploitent le changement pour donner un avantage compétitif au client.

3

Livrer fréquemment un logiciel opérationnel avec des cycles de quelques semaines à quelques mois et une préférence pour les plus courts.

4

Les utilisateurs ou leurs représentants et les développeurs doivent **travailler ensemble quotidiennement** tout au long du projet.

5

Réaliser les projets avec des **personnes motivées**. Fournissez-leur l'**environnement et le soutien** dont ils ont besoin et faites-leur confiance pour atteindre les objectifs fixés.

6

La méthode la plus simple et la plus efficace pour **transmettre de l'information** à l'équipe de développement et à l'intérieur de celle-ci est le **dialogue en face à face**.

7

Un logiciel opérationnel est la principale mesure d'avancement.

8

Les processus agiles encouragent un **rythme de développement soutenable**. Ensemble, les commanditaires, les développeurs et les utilisateurs devraient être capables de maintenir indéfiniment un rythme **constant**.

9

10

Une attention continue à l'**excellence technique** et à une bonne conception renforce l'agilité.

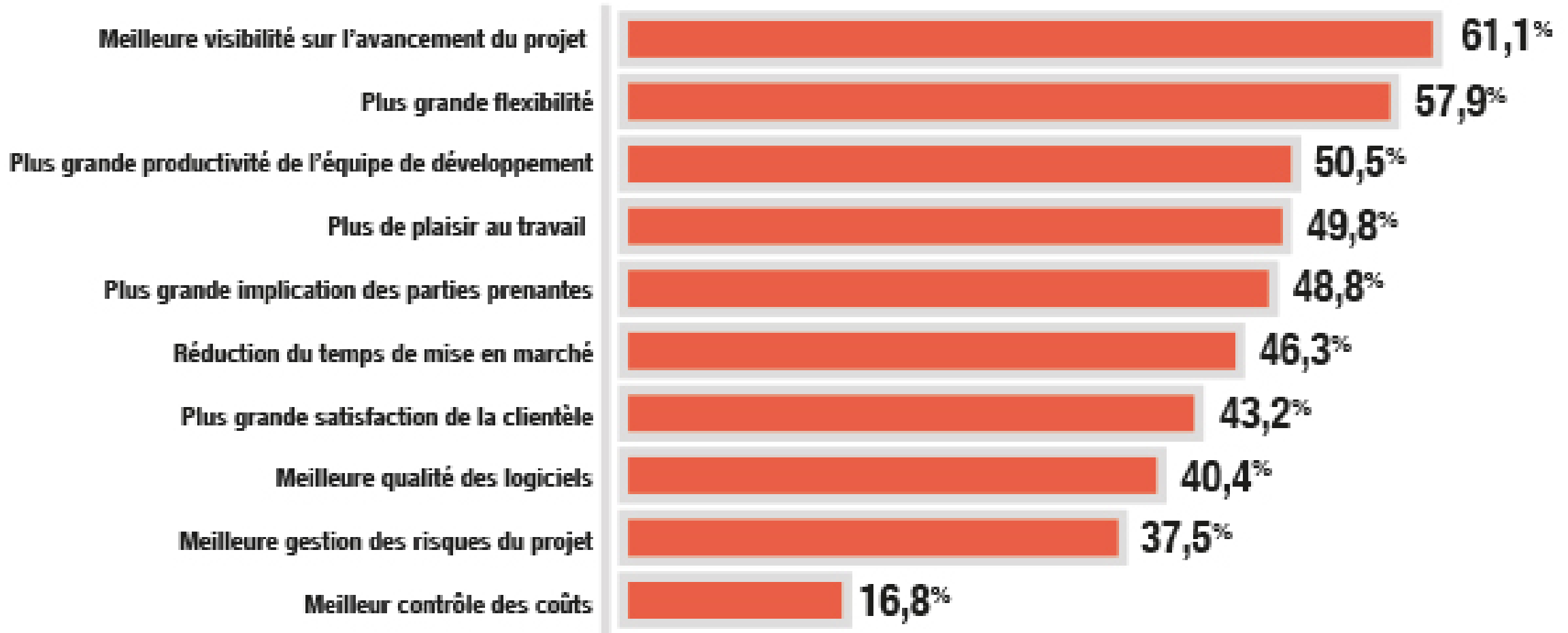
11

La simplicité – c'est-à-dire l'art de minimiser la quantité de travail inutile – est essentielle.

12

Les meilleures architectures, spécifications et conceptions émergent d'**équipes auto-organisées**.

À intervalles réguliers, l'équipe réfléchit aux **moyens de devenir plus efficace**, puis règle et modifie son comportement en conséquence.



76,9% | Culture organisationnelle difficile à changer

65,8% | Résistance au changement

53,0% | Manque d'expérience

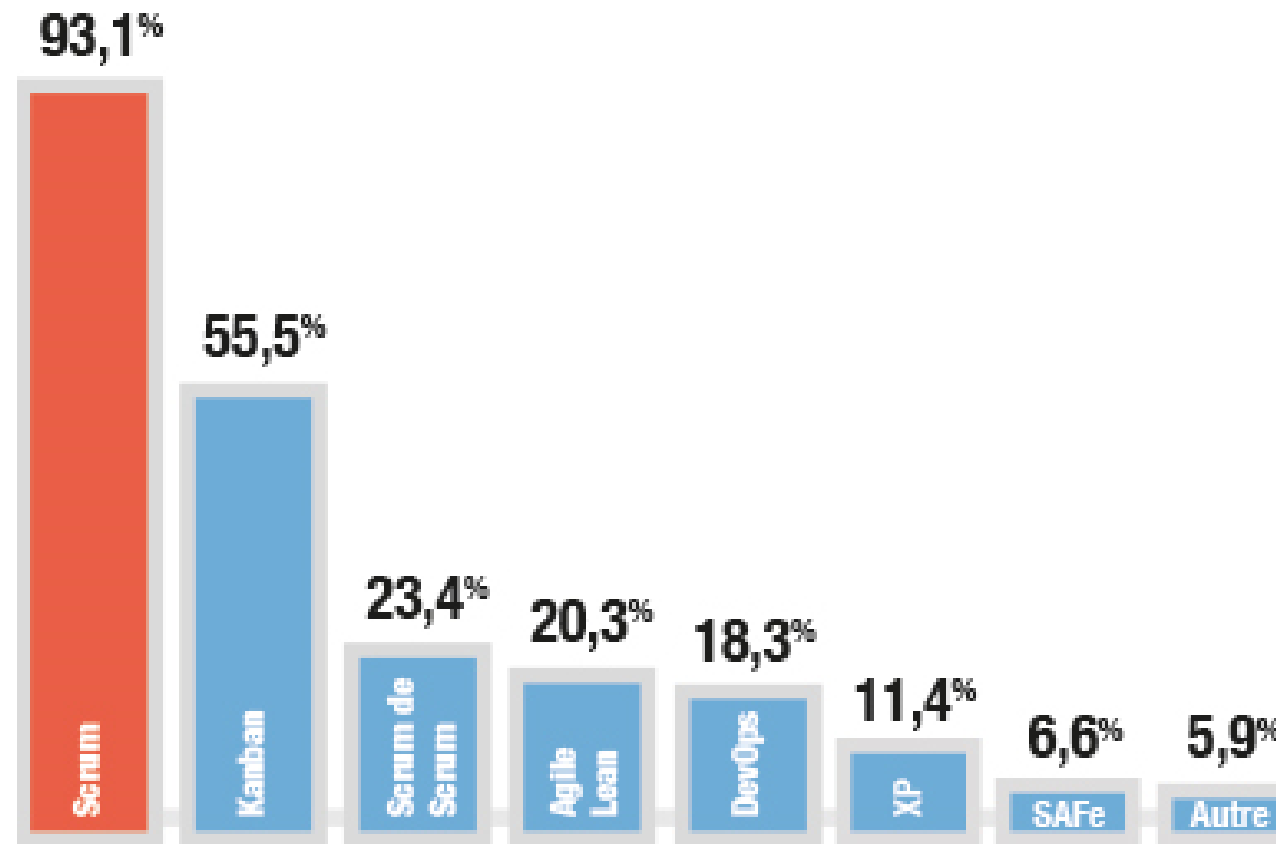
39,9% | Manque d'implication de la part des parties prenantes

34,5% | Manque d'engagement de la part de la direction

Ex aequo
28,1% | Complexité du projet

28,1% | Difficultés de communication

Les méthodes agiles



Les méthodes et pratiques agiles les plus utilisées

C'est une méthode de développement logiciel, **orientée programmation**.

XP préconise un certain nombre de pratiques :

- **Pair programming** : pratique fondamentale d'XP. Deux développeurs travaillent ensemble, l'un écrit le code, l'autre regarde, propose, corrige et commente. Et les rôles s'inversent. Les connaissances sont ainsi partagées, et les idées émanent plus facilement de ce type d'organisation.
- **Test Driven Development** : développement dirigé par les tests. C'est une technique de développement qui suppose de rédiger les cas de test, puis que l'on écrive le code, teste et enfin revisite le code produit. C'est une pratique assez classique au niveau des tests unitaires, qui peut s'appliquer également au niveau des tests d'acceptance (test fonctionnel).

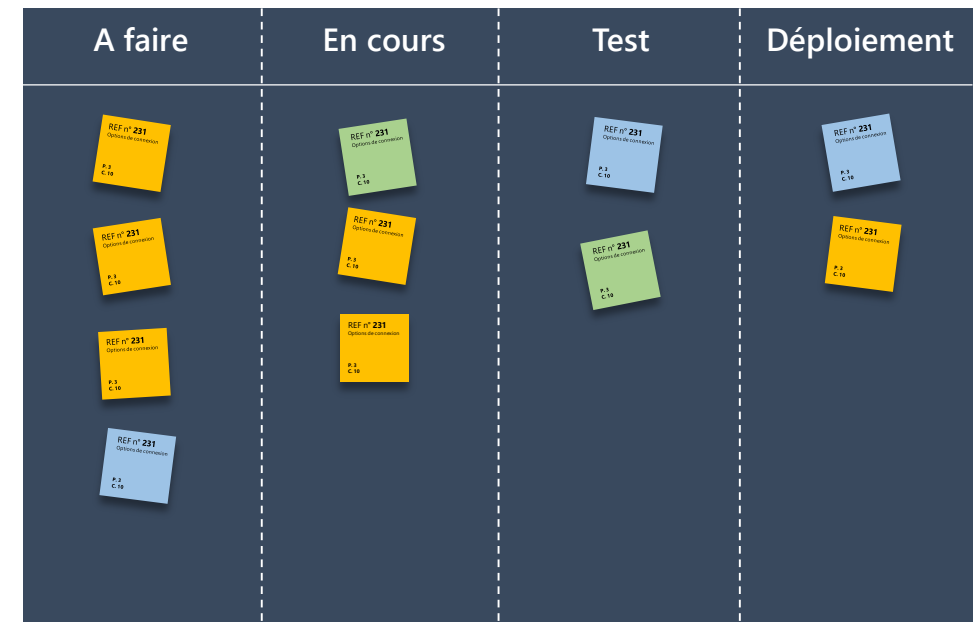
L'écriture du code se fait suivant 6 itérations rapides :

1. Conception rapide (optionnelle)
2. Ecriture d'un petit nombre de test unitaires automatisés
3. Lancement des tests pour vérifier qu'ils échouent (car pas encore de code correspondant)
4. Implantation du code à tester
5. Lancement des tests jusqu'à ce qu'ils passent tous
6. Restructuration du code et des tests si nécessaire pour améliorer les performances ou le design.

La méthode Kanban est tournée vers l'**amélioration des processus et la fluidité des flux de production**.

La visualisation du workflow : la méthode Kanban préconise de mettre en œuvre des moyens visuels et explicites pour visualiser les processus qui interviennent dans le projet.

Tableau Kanban : chaque carte regroupe les informations utiles comme la description de la fonctionnalité, un identifiant, la date d'entrée dans le processus et sa date de sortie.



Contrairement à d'autres méthodes agiles comme Scrum ou XP, Kanban n'est pas centrée sur la notion d'itération. Le travail se fait **au fil de l'eau**.

Limiter le travail en cours : en partant du constat que plus on fait de choses en même temps, plus on a tendance à perdre en productivité, la méthode Kanban recommande de limiter la quantité de travail en cours, appelé **WIP** pour work in progress. On instaure donc des limites dans le système.



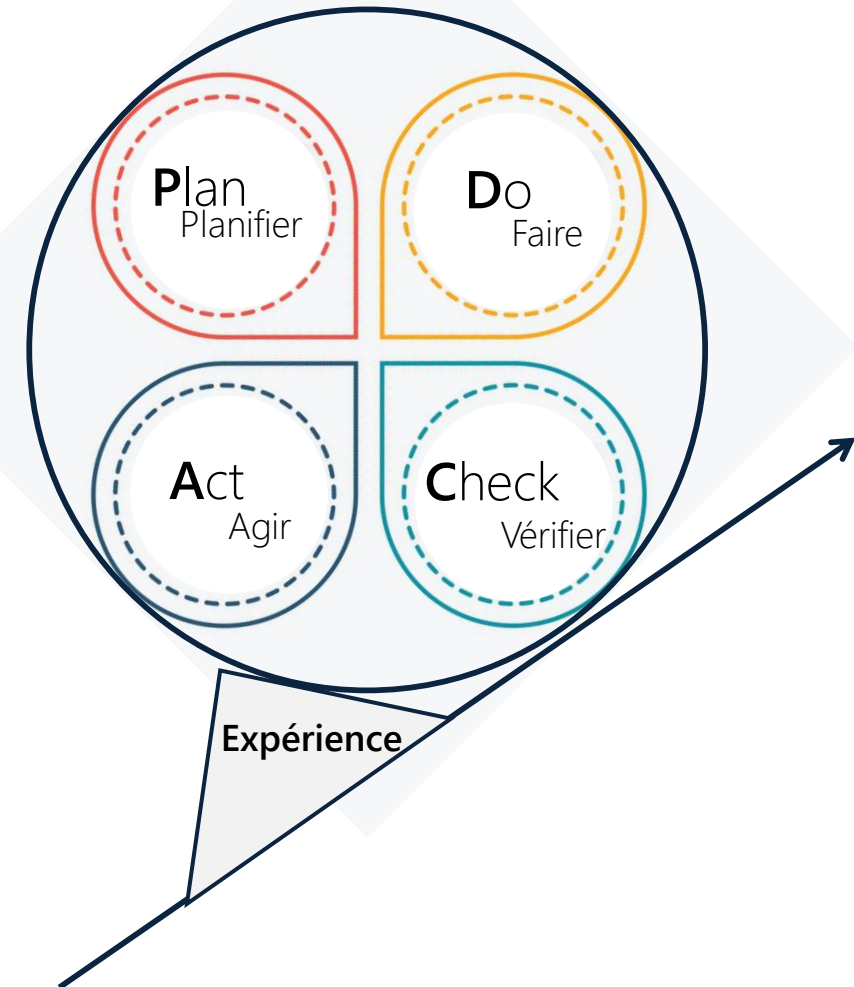
Scrum est un **cadre méthodologique** : il définit des rôles, un rythme itératif avec des réunions précises et limitées dans le temps, des artefacts (product backlog, sprint backlog).

Scrum en soi ne comporte aucune pratique ni outil de développement. Généralement, on complète Scrum par les pratiques de développement issues de la méthode XP telles que la programmation en binôme, le développement piloté par les test (TDD), la description des besoins/fonctionnalités en User Stories, etc. On va aussi ajouter un outillage tel que l'intégration continue.

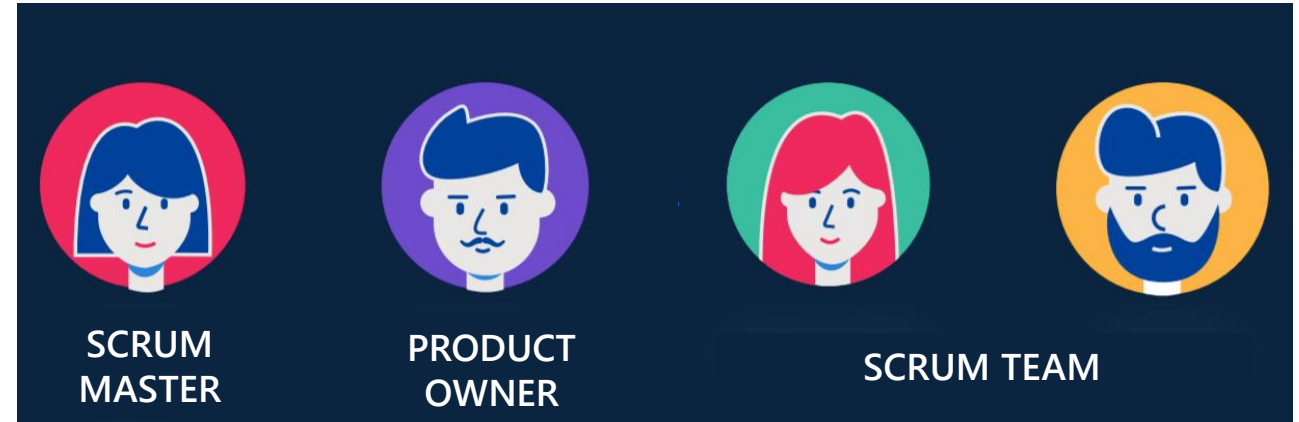
Scrum est basé sur l'idée de **processus empirique**, en considérant que **la connaissance vient de l'expérience**. Le principe est de focaliser le travail de l'équipe sur une partie limitée et maîtrisable des fonctionnalités à réaliser.

Chaque incrément du produit est réalisé successivement, pendant des périodes de durée fixe d'une à quatre semaines, appelées sprints. Chaque sprint possède un objectif, défini par le propriétaire du produit, à partir duquel sont choisies les fonctionnalités à implémenter.

Roue de Deming Amélioration continue



- **Le product owner** : il porte la **vision du produit** à réaliser (représentant généralement le client). Il doit faire en sorte qu'un maximum de valeur soit produite par l'équipe, en comprenant parfaitement les enjeux et en faisant en sorte que les fonctionnalités importantes soient priorisées comme telles. Il est responsable du product backlog et des activités associées : description des user-stories, ordonnancement du backlog, s'assurer de la bonne compréhension de ses éléments par l'équipe...



- **Le scrum master** : il est le **facilitateur**, celui qui s'assure que Scrum est compris et accepté par toute l'équipe (il insuffle et facilite, il est à l'écoute de l'équipe et prend toutes les dispositions possibles pour faciliter la progression de l'équipe).
- **L'équipe** : elle est composée des personnes qui travaillent pour livrer un incrément « Fini » et potentiellement livrable en production. elle **s'auto-organise** pour créer des incréments de produit fonctionnels et de qualité.

- **Le product backlog** : il représente la liste des fonctionnalités à développer, priorisée et maintenue par le product owner. Cet artefact est sous la responsabilité du Product owner. Il est voué à évoluer pour correspondre au mieux aux besoins.
- **Le sprint backlog** : il est constitué d'un sous ensemble du product backlog représentant des fonctionnalités qui ont été choisies pour être implémentées pendant le sprint.
- **L'incrément produit** : c'est l'ensemble des items du backlog complétés pendant le sprint. A la fin du sprint, le nouvel incrément doit être « Fini » selon le « definition of done » défini par l'équipe.
- **Definition of done** : il s'agit de définir une liste de critères qui doivent être remplis pour pouvoir considérer que le produit est livrable à la fin du sprint. Elle doit être comprise de la même manière par tous les membres de l'équipe.

Les cérémonies sont organisées dans une logique d'amélioration continue, et sont toutes limitées dans le temps.

Le sprint

Le sprint correspond à l'itération pendant laquelle est fabriqué un produit fini, exploitable. Chaque sprint possède donc un but, et une liste de fonctionnalités à développer.

Il n'est pas mis en production à chaque fois, car il peut être insuffisant pour être utilisable et rendre un service, mais permet de se rendre compte réellement de l'avancement du produit.

Il est d'une durée moyenne de **2 semaines à 1 mois maximum**. La limite d'un mois permet de réduire la complexité et donc les risques liés au sprint. Une fois la durée déterminée, elle reste constante pendant toute la durée du développement.

Avant le sprint

La réunion de planification de Sprint

Réunion pendant laquelle l'équipe planifie le travail qui va être réalisé durant le sprint à venir (qu'est ce qui va être fait, comment on va le faire, et quel est le but global du sprint).

Tous les acteurs doivent être présents à cette réunion qui ne doit pas durer plus de **8 heures** pour un sprint d'un mois. Pour un sprint plus court, la durée est réduite proportionnellement.

Pendant le sprint

Le daily scrum

Il s'agit d'une réunion de synchronisation de l'équipe qui se fait debout (elle est aussi appelée "stand up meeting").

Elle permet à l'équipe, chaque jour en **15 minutes**, d'inspecter son travail et de s'adapter (chaque membre répond à ces 3 questions : qu'ai-je fait hier, que vais-je faire aujourd'hui, quels sont les problèmes que je rencontre).

Cette réunion a pour but de faire un point de coordination, mais ne doit pas être vécue comme un rapport d'activité.

C'est le niveau quotidien du principe « Inspect and Adapt » de Scrum.

Après le sprint

La revue de sprint (parfois appelée « sprint demo »)

C'est une réunion de **4 heures maximum** avec les parties prenantes, pendant laquelle l'équipe présente ce qui a été fait et obtient des retours constructifs.

L'objectif de la revue de sprint est d'inspecter l'incrément produit réalisé au cours du sprint écoulé, de faire un point sur l'avancement de la Release et d'adapter au besoin le Product Backlog.

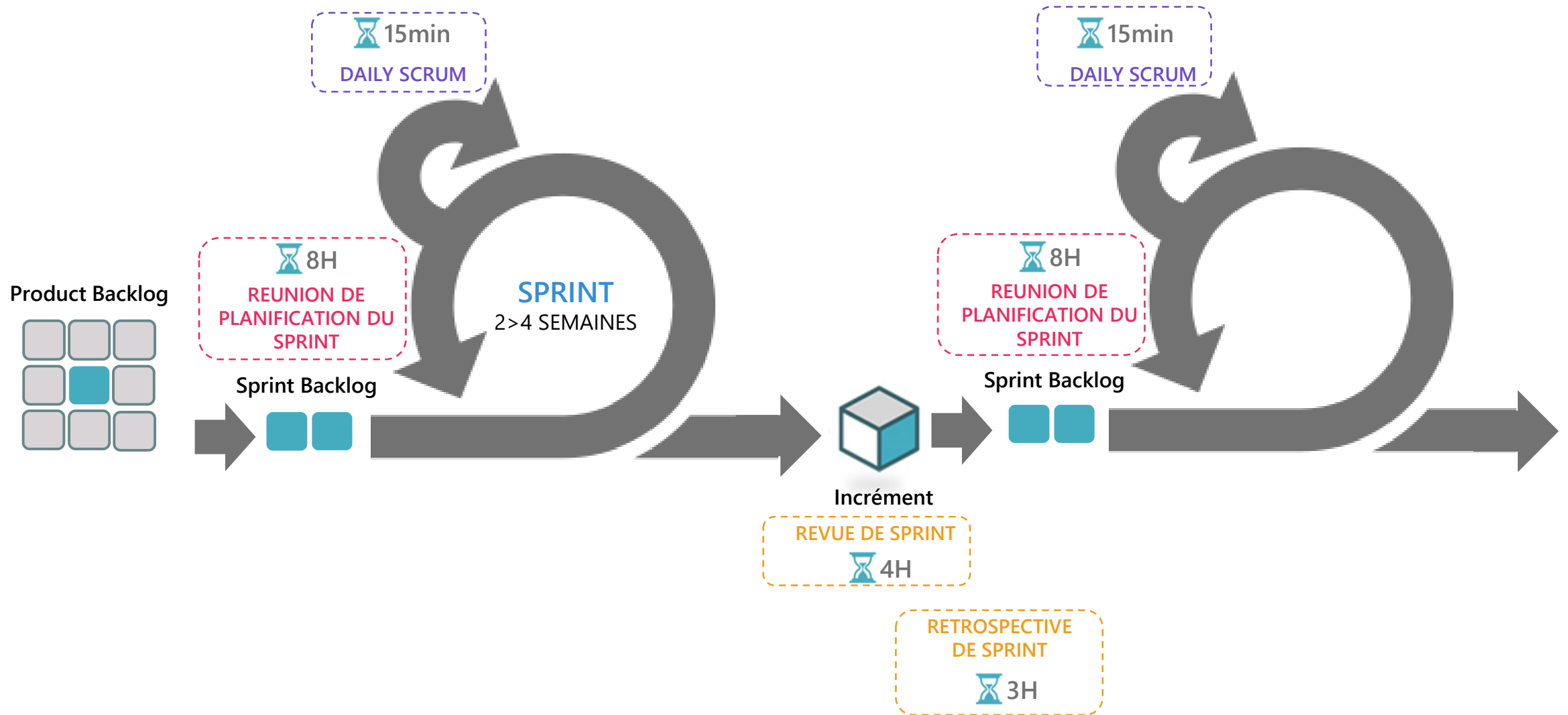
L'équipe de développement présente à tout acteur projet intéressé (a minima le Product Owner idéalement accompagné d'utilisateurs finaux) les nouvelles fonctionnalités développées au cours du sprint. Le Product Owner donne un feedback à l'équipe de développement, il accepte ou refuse les fonctionnalités présentées.

C'est sur la base de cette démonstration que le product owner valide les fonctionnalités développées pendant ce sprint.

La rétrospective de sprint

Elle a généralement lieu après la revue de sprint et doit permettre à l'équipe de s'améliorer (productivité, qualité, efficacité, conditions de travail, etc.) au regard du "vécu" sur le sprint écoulé (principe d'amélioration continue). Elle dure **3 heures** pour un sprint d'un mois.

Cette réunion est généralement animée par le ScrumMaster qui s'adresse à son équipe. Elle a pour but d'améliorer continuellement le processus de développement de l'équipe en mettant les idées de chacun à contribution.

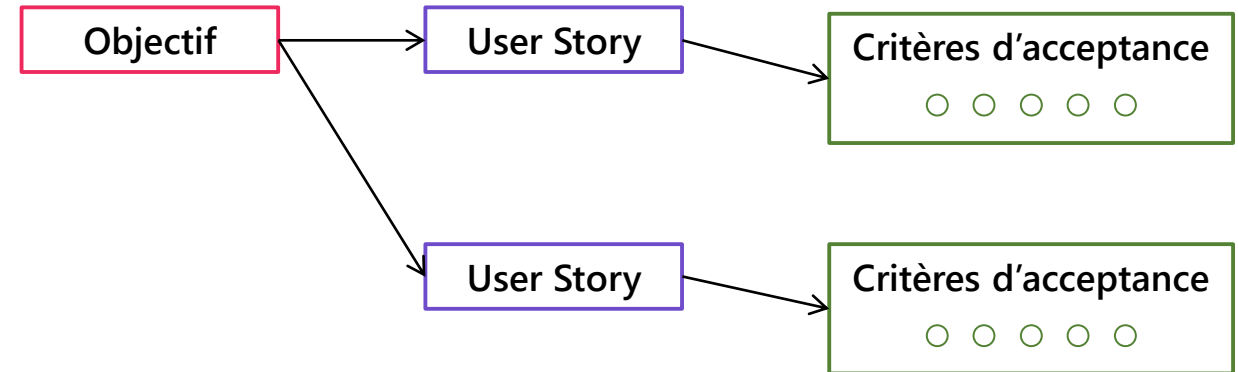


La rédaction d'une User Story se fait en trois **étapes incrémentales** appelées les 3C (Card, Conversation, Confirmation) :

- Une description du besoin

En tant que <rôle>, je veux pouvoir <action>, afin de <objectif>.

- Une négociation en vue d'une définition du besoin
- Une définition des critères d'acceptance



Une bonne User Story doit aussi respecter les **caractéristiques** réunies sous le sigle INVEST :

- **Indépendante** : assure l'indépendance d'une User Story vis-à-vis des autres user stories du backlog.
- **Négociable** : une User Story doit être un support de discussion en vue d'une amélioration du besoin initial.
- **Valorisable** : la réalisation d'une User Story doit rendre un service à l'utilisateur, elle n'a de sens que si elle apporte une valeur métier.
- **Estimable** : elle doit être bien définie pour être facilement chiffrable.
- **Suffisamment petit** : une User Story doit être réalisable sur un sprint.
- **Testable** : une User Story doit être accompagnée de critères d'acceptance pour permettre sa validation.

Priorisation :

La priorisation se fait en fonction de la valeur ajoutée métier, par le product owner. Il peut à tout moment revoir la priorité des user stories qui n'ont pas encore été planifiées dans le sprint en cours. Le changement est autorisé et encouragé pour pouvoir éliminer les idées de départ qui s'avèreraient mauvaises ou permettre de prendre en compte les nouvelles idées.

La méthode de MoSCoW permet de prioriser les user stories selon les critères suivants :

Mo - Must : doit être réalisée (vitale).

S - Should : devrait être réalisée si possible (essentielle).

Co - Could : pourrait être réalisée s'il n'y a pas d'impact sur les autres tâches en cours (confort).

W - Won't : ne sera pas réalisée tout de suite mais serait souhaitable pour une version ultérieure (luxe).

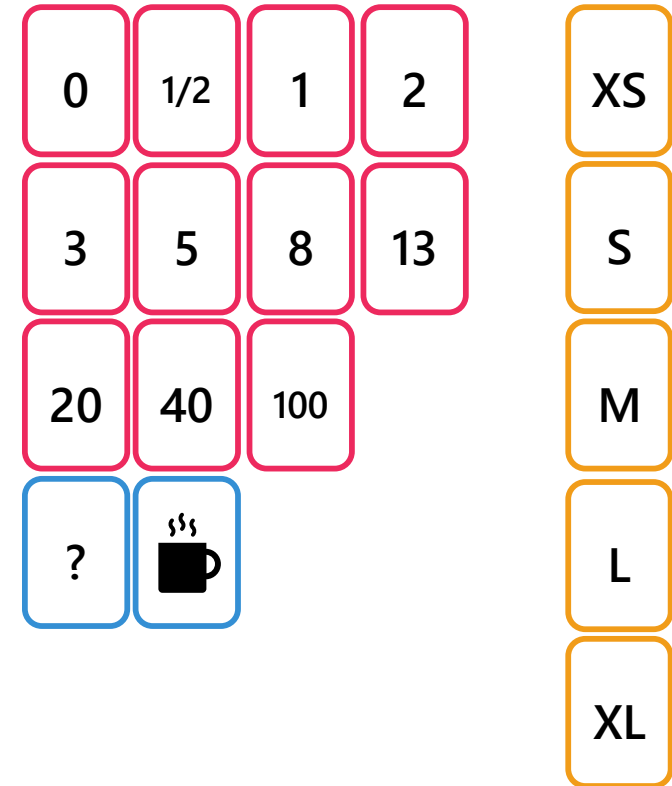
La priorisation et l'estimation des user stories font partie de l'activité de construction du backlog, qui implique le product owner et l'ensemble de l'équipe.

Estimation :

L'estimation est généralement faite par relativité en définissant un étalon de base : on part d'une user-story que l'on sait évaluer, qui servira de mesure de référence.

Il existe différentes techniques d'estimation : avec le planning poker basé sur une suite d'intervalles de plus en plus larges (pour témoigner de l'incertitude sur la maîtrise du sujet), ou la taille de t-shirt.

Ces estimations sont retranscrites en story points. En fixant le nombre de story points qui peuvent être réalisés par l'équipe au cours d'un sprint, il sera ensuite possible de sélectionner le nombre de user-stories qui pourront être intégrées à un sprint.



Activités de test et rôles associés

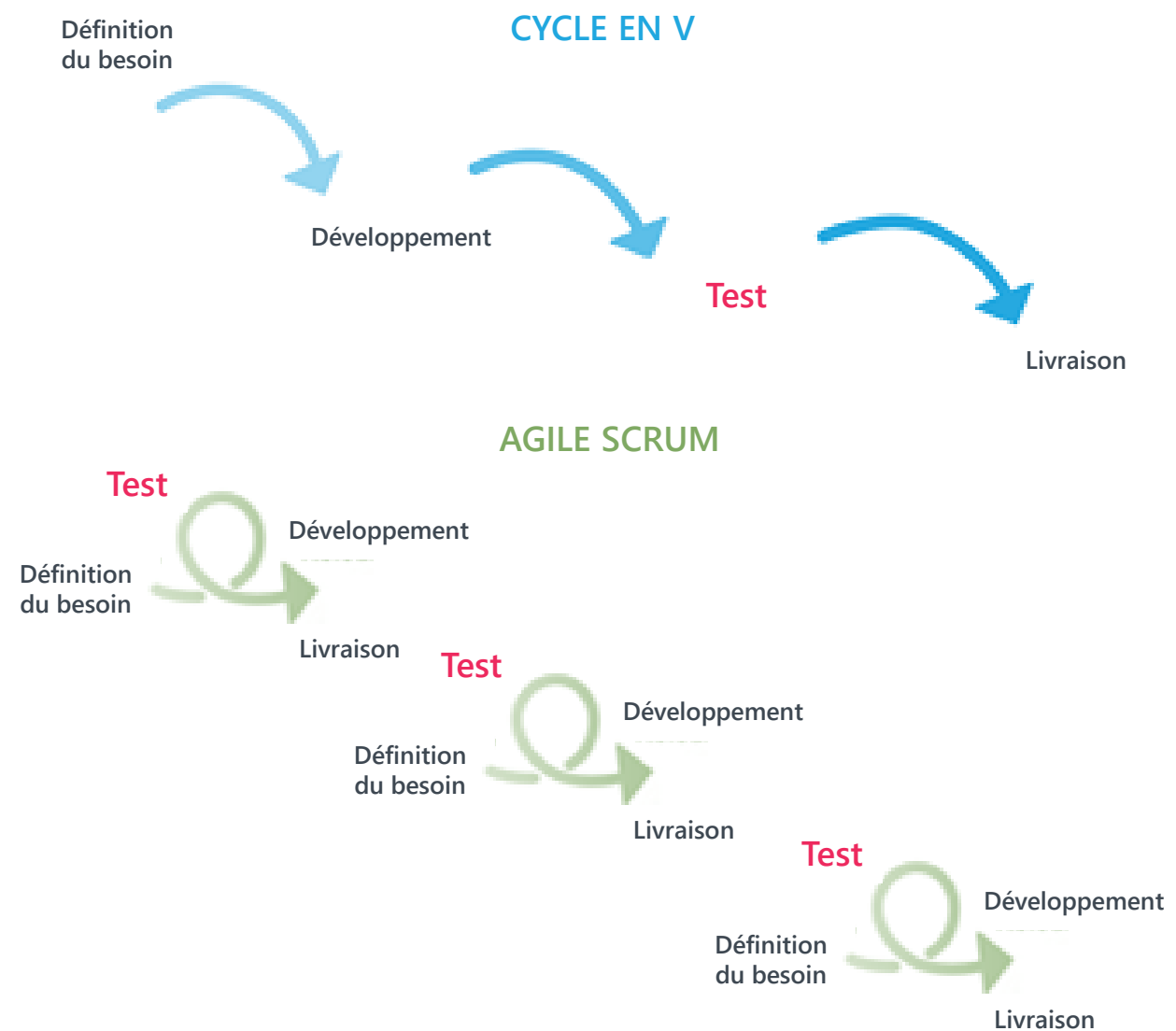
Le test dans la démarche agile

Dans une approche agile, la nécessité de tester les développements conserve toute son importance. La mise en place d'une réelle stratégie de test est nécessaire pour assurer une qualité satisfaisante du logiciel et renforcer l'image de marque et la légitimité du produit sur le marché.

- Les testeurs peuvent faire directement partie de l'équipe agile. Celle-ci rassemble alors les activités de test et de développement, ce qui facilite l'échange entre les membres de l'équipe.
- Ces activités peuvent également être différenciées, avec une équipe exclusivement dédiée aux tests, en fonction de la taille et de la complexité du projet.

Comparativement aux approches traditionnelles, la démarche Agile intègre les tests et l'assurance qualité très en amont dans le cycle de vie du projet et la place du test est encore plus importante dans une démarche agile :

- Les tests commencent plus tôt
- Les tests sont réalisées très régulièrement



On utilise l'acronyme **PURIFF** comme concept pour regrouper l'ensemble des activités de tests à mener durant le Sprint.

Chaque élément de l'acronyme correspond à une catégorie de tests :

- P : pour les tests de Performances
- U : représente les tests Unitaires
- R : désigne les tests de non Régression
- I : couvre les tests d'Intégration
- F : pour les tests Fonctionnels
- F : représente les tests (Non) Fonctionnels

Les tests de Performance :

Ils consistent à **mesurer les temps de réaction de l'application, soumise à une charge d'utilisation donnée.**

Rôle des testeurs :

- Aider les développeurs à concevoir les tests,
- Aider à déterminer les plages à tester
- Exécuter les tests de performance, avec ou sans l'aide des développeurs

Les tests Unitaires :

Ils sont écrits par les développeurs pour tester les classes de façon unitaire. Ils s'inscrivent en général dans une stratégie de développement dirigé par les tests (TDD).

Rôle des testeurs :

- Aider les développeurs à exécuter les tests unitaires
- Automatiser les tests unitaires

Les tests de non régression :

L'objectif des tests de non-régression est de garantir que des défauts n'ont pas été introduits suite à la correction ou la modification de l'application.

Les tests de non régression constituent une activité fastidieuse notamment dans un contexte Agile, caractérisé par des modifications et des livraisons fréquentes. Exécutés manuellement, ils sont très consommateurs de ressources. Il est fortement recommandé d'automatiser ces tests.

Les tests d'intégration :

Les tests d'intégration représentent le niveau qui se situe juste au-dessus des tests unitaires. Ils consistent à vérifier que les différents modules fonctionnent correctement une fois assemblés.

Ils sont en général exécutés sur un environnement d'intégration continue de façon automatisée.

Les tests fonctionnels :

Les tests fonctionnels sont écrits et mis en place afin de vérifier que les fonctionnalités de l'incrément répondent aux spécifications définies dans le Backlog du produit. Ils correspondent aux tests des critères d'acceptance des éléments du Backlog (user stories). Ces tests peuvent être déroulés manuellement ou de façon automatisée.

Rôle des testeurs :

- Concevoir les TNR
- Automatiser les TNR
- Adapter les TNR en continu

Rôle des testeurs :

- Vérifier que les TI ont été exécutés
- Participer au bilan des tests

Rôle des testeurs :

- Concevoir les tests fonctionnels
- Les exécuter Participer au bilan des tests
- Automatiser les tests fonctionnels

Les tests non fonctionnels :

Les tests non fonctionnels permettent de vérifier les aspects non fonctionnels de l'application.

Ces aspects sont souvent définis comme étant des exigences non fonctionnelles (user stories techniques). Par exemple : les tests de montée en charge, les tests de robustesse, les tests de sécurité, les tests de portabilité, etc.

Rôle des testeurs :

- Participer à la création et à l'exécution
- Participer au bilan des tests

JIRA en contexte agile

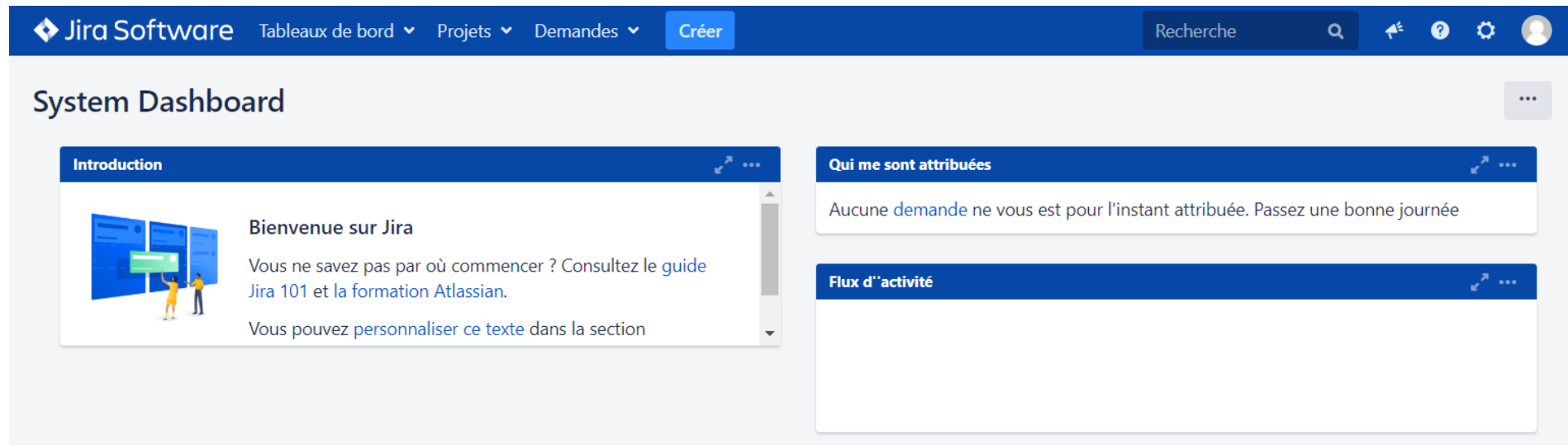
- I. Fonctionnalités et concepts clés de JIRA
- II. Recherches et filtres

Fonctionnalités et concepts clés de JIRA

Vue d'ensemble

JIRA est un gestionnaire de demandes.

Le terme « demande » utilisé dans JIRA est un terme générique qui peut correspondre à des anomalies, tâches, user-stories...



Les projets software

Un projet Jira Software comprend plusieurs demandes et des outils qui permettent à l'équipe Agile de coordonner le développement du logiciel.

JIRA permet de choisir parmi 3 options :

- **Le développement de logiciel Scrum** : avec un backlog, des sprints et des rapports d'avancement, un flux de travail agile et l'édition de demandes.
- **Le développement logiciel Kanban** : avec un tableau Kanban, un flux de travail basique et la création de demandes. Le mode Kanban permet une répartition des tâches en colonnes entre ce qui est « fait », « en cours », « à faire ». Ce mode convient au monitoring des équipes et au suivi d'avancement d'un projet. Il est possible d'instaurer les limites en n'autorisant par exemple pas plus de 10 tâches dans la colonne « to do ».

Il est possible de mixer les deux approches et d'organiser le travail en colonnes sur un sprint donné. Ou au contraire d'utiliser le tableau Kanban en amont du backlog.

- **Le développement logiciel basique** : un flux de travail classique et la création de demandes.

Créer un projet

[Afficher les flux de travaux Marketplace](#)

Software



Développement logiciel Scrum

Développement d'Agile avec un tableau, des sprints et des récits. Connecté aux outils de...



Développement logiciel Kanban

Optimiser le flux de développement avec un tableau. Connecté aux outils de contrôle de la...



Développement logiciel basique

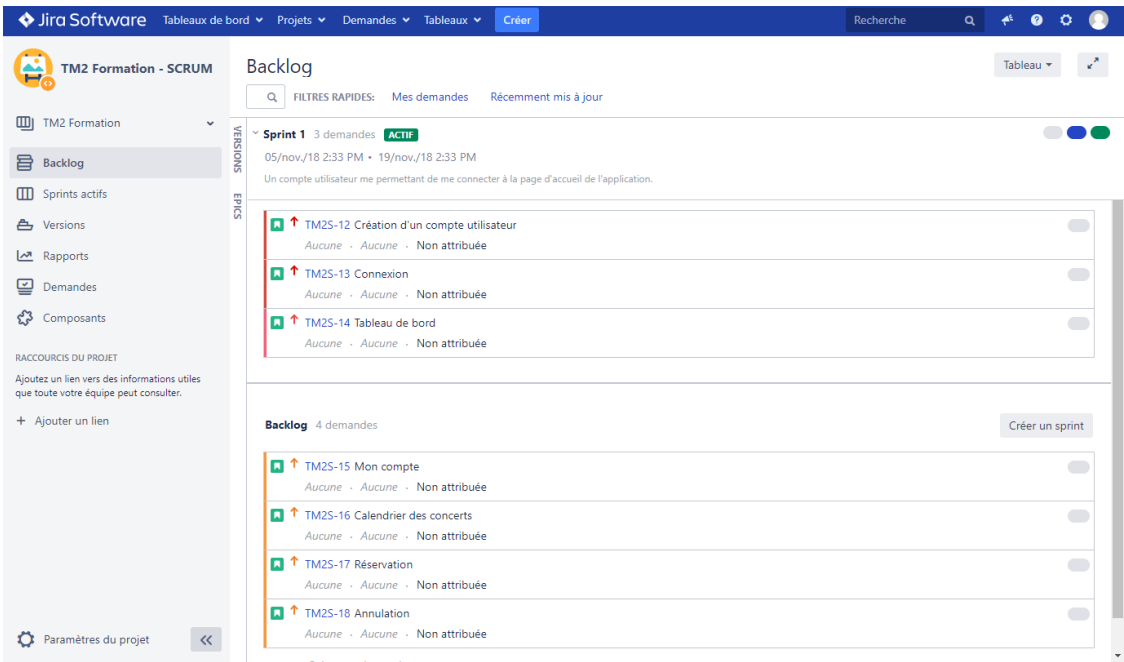
Suivre les tâches de développement et les bogues. Connecté aux outils de contrôle de la...

Les fonctionnalités et concepts clés de JIRA Software

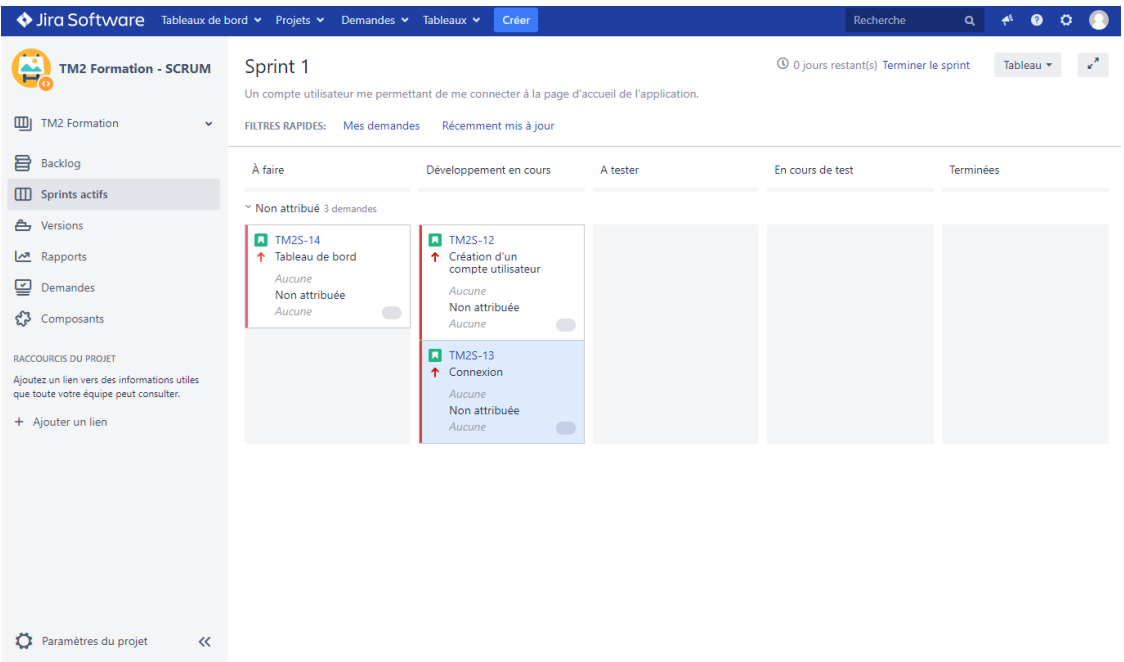
Projet Scrum

Il est possible de mixer les deux approches et d'organiser le travail en colonnes sur un sprint donné. Ou au contraire d'utiliser le tableau Kanban en amont du backlog.

Une fois le projet créé on accède à son backlog (liste des demandes priorisées).




Backlog : mode Planifier



Backlog : mode Travail

Un workflow est un ensemble de statuts et de transitions par lesquels une demande peut passer durant le cycle de vie de développement. Dans JIRA il s'agit de l'ensemble des étapes par lesquelles la demande peut passer durant son cycle de vie. Le workflow dans JIRA est fortement personnalisable, pour l'adapter au mieux aux besoins de l'équipe.

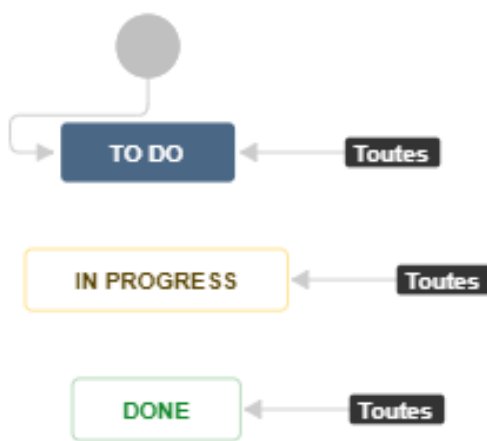



Flux de travaux

Un flux de travaux détermine les états et les transitions qu'un type de demande peut utiliser dans un projet.

Système:
TM2: Software Simplified Workflow Scheme

Software Simplified Workflow for Project TM2



**HalimPierre**

HALTableau de

Backlog

Sprints actifs

Versions


Rapports

Demandes

Composants

RACCOURCIS DU PROJET

Ajoutez un lien vers des informations utiles que toute votre équipe peut consulter.

**HalimPierre** / HAL-4

Ma demande numéro 2

Modifier

Commentaire

Attribuer

Suite

À faire

En Cours

Terminé

Gérer

Informations

Type: Récit

Priorité: Medium

Affecte la/les version(s): Aucune

Etiquettes: Aucune

Lien d'épopée: Epic 1

Sprint: HALTableau Sprint 2

Description

Cliquer pour ajouter une description

Pièces jointes

Etat: A FAIRE (Afficher le flux de travaux)

Résolution: Non résolu

Version(s) corrigée(s): Aucune

37

Les demandes

La demande (« issue ») est l'élément fondamental de JIRA.

Chaque demande JIRA est porteuse d'informations. Cette information est structurée à l'aide de champs.

Par défaut la demande comprend différents champs, dont certains sont obligatoires :

- Projet de la demande
- Type de la demande
- Résumé de la demande
- Rapporteur

Créer une demande

[Configurer les champs](#)

Projet* HENIX Formations TM2 (TM2)

Type de demande* Récit

Résumé* Création d'un compte utilisateur

Rapporteur* Formateur

Commencer à taper pour obtenir une liste de correspondances possibles.

Composants Aucune

Descriptif

Style B I U A +

En tant qu'utilisateur qui accède pour la première fois à l'application, **je veux pouvoir** créer un compte utilisateur, **afin de** pouvoir accéder à l'application.

☐ Créer une autre demande [Créer](#) [Annuler](#)

Les demandes

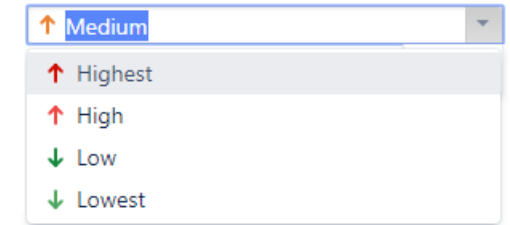
Chaque demande permet de définir sa priorité, sa durée, d'ajouter des pièces-jointes, ou toute autre information qui pourra être portée dans des champs personnalisés.

- **Un niveau de priorité** (lowest, low, medium, high, highest)

Ordonnancement du backlog : par défaut les demandes du backlog sont rangées dans l'ordre dans lequel elles ont été créées. Il est possible de modifier cet ordre pour l'ajuster à la priorité de la demande. Cela permet d'organiser le backlog de manière plus efficiente : il est par exemple possible d'avoir deux demandes de priorité 'High', en ordonnant ces demandes il est possible d'attribuer un rang plus haut et donc prioritaire à l'une ou l'autre.

- **Estimer sa durée**
- Ajouter des **pièces-jointes** par un glisser-déposer
- **Créer une sous-tâche** (Consultation de la demande > Suite > Créer une sous-tâche)
 - Pour diviser une demande en plus petits morceaux.
 - Pour permettre d'assigner à des personnes différentes des aspects de la demande.
 - Pour créer une to-do list.
- **Attribuer la demande** à un utilisateur


Priorité:



A screenshot of the JIRA priority dropdown menu. The menu is open, showing five options: 'Medium' (selected), 'Highest', 'High', 'Low', and 'Lowest'. Each option is preceded by a small arrow icon indicating its relative position in the hierarchy.

Personnes

Responsable:

 Non attribuée

[Me l'affecter](#)

Rapporteur:

 Formateur

Votes:

0

Gérer les

observateurs:

0 [Commencer l'observation de cette demande](#)

Implémenter les user stories

Le backlog est une liste de demandes qui représentent les fonctionnalités à développer pour un projet. Les demandes qui existeront dans le backlog pourront ensuite être déplacées vers un sprint pour être intégrées à celui-ci. Il sera possible d'organiser le backlog, de réunir les demandes en epics et de démarrer un sprint.

1. Dans le menu de navigation haut, cliquez sur **Créer**

2. Pour la demande, renseignez :

- **Projet** : sélectionnez votre projet
- **Type de demande** : Récit

Les types de demandes de la configuration par défaut :

- Récit (user story) : il s'agit de la description en une à deux phrases et dans le langage utilisateur d'une fonctionnalité à développer.
- Épopée (Epic) : les récits peuvent être regroupés en différentes épopées, qui décrivent une fonctionnalité entière recouvrant plusieurs user stories.
- Une tâche
- Un bug

- **Résumé**
- **Description** : Rédigez la description de la user-story au format
En tant que [rôle], je veux pouvoir [action], afin de [but].

3. Cliquez sur **Créer**

Créer une demande

Configurer les champs

Projet* HENIX Formations TM2 (TM2)

Type de demande* Récit ?

Résumé* Création d'un compte utilisateur

Rapporteur* Formateur

Commencer à taper pour obtenir une liste de correspondances possibles.

Composants Aucune

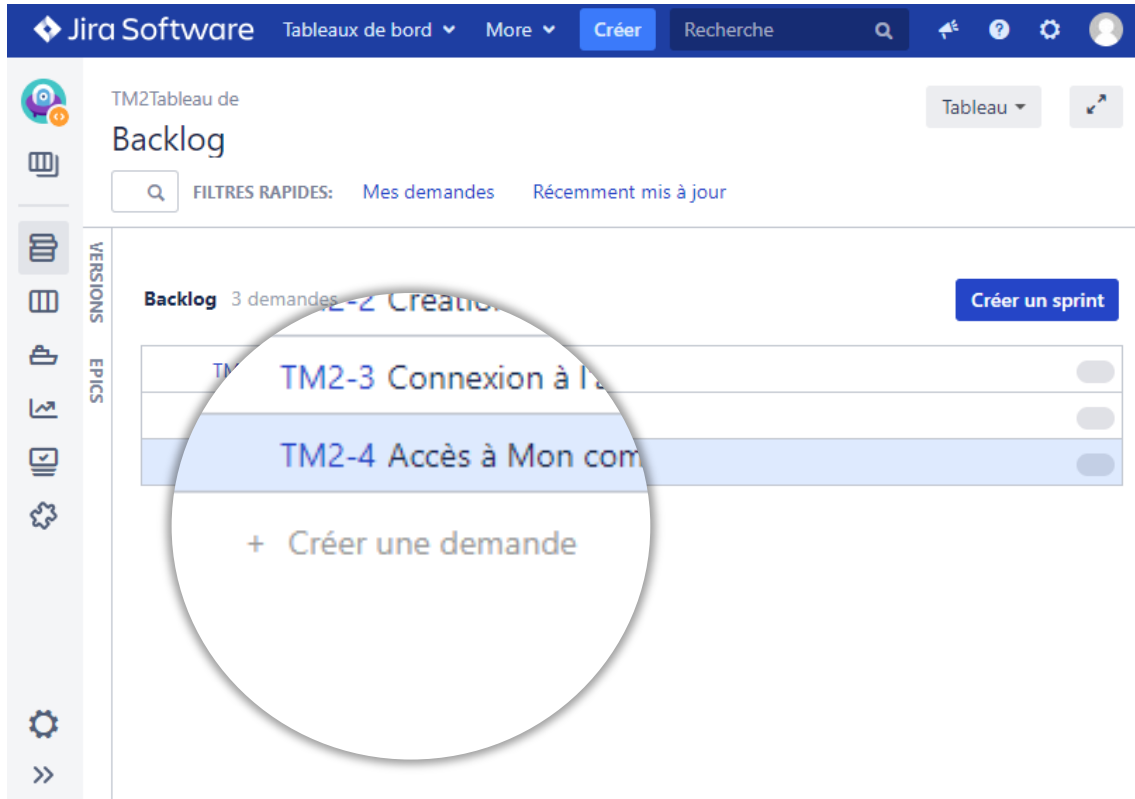
Descriptif

Style B I U A v A° v P v :≡ ≡ ☹ v + v ^

En tant qu'utilisateur qui accède pour la première fois à l'application, je veux pouvoir créer un compte utilisateur, afin de pouvoir accéder à l'application.

Créer une autre demande Créer Annuler

Implémenter les user stories



Vous pouvez aussi créer une demande directement à partir du backlog.

1. A partir de la vue backlog, cliquez sur **+ Créer une demande**
2. Pour la demande, renseignez le résumé : « Connexion à l'application ».
3. Cliquez sur **...** pour renseigner les données complémentaires de la demande (type, projet, description).
4. Cliquez sur **Créer**
5. Liez les demandes de votre backlog entre elles (page de consultation d'une demande > Suite > Lien)



Créez toutes les user-stories dans votre backlog



Créez votre projet

1. Créer un projet Développement Projet Scrum

- Choisissez le flux de travaux par défaut
- Nom du projet « QL35_AuditeurXX_SVEL

2. Dans le backlog, créer deux épics (Epopée) :

- « Navigation »
- « Vente en ligne »

3. Créer pour l'épic « Navigation » les 3 US suivants permettant de :

- S'authentifier
- Consulter le catalogue
 - Sous tâche : Afficher les images du produit
 - Sous tâche : Gérer la pagination des produits présentés
- Sélectionner un produit

4. Créer pour l'épic « Vente en ligne » les 3 US suivants permettant de :

- Valider le panier
- Indiquer une adresse de livraison
- Effectuer un paiement

5. Créer deux sprints :

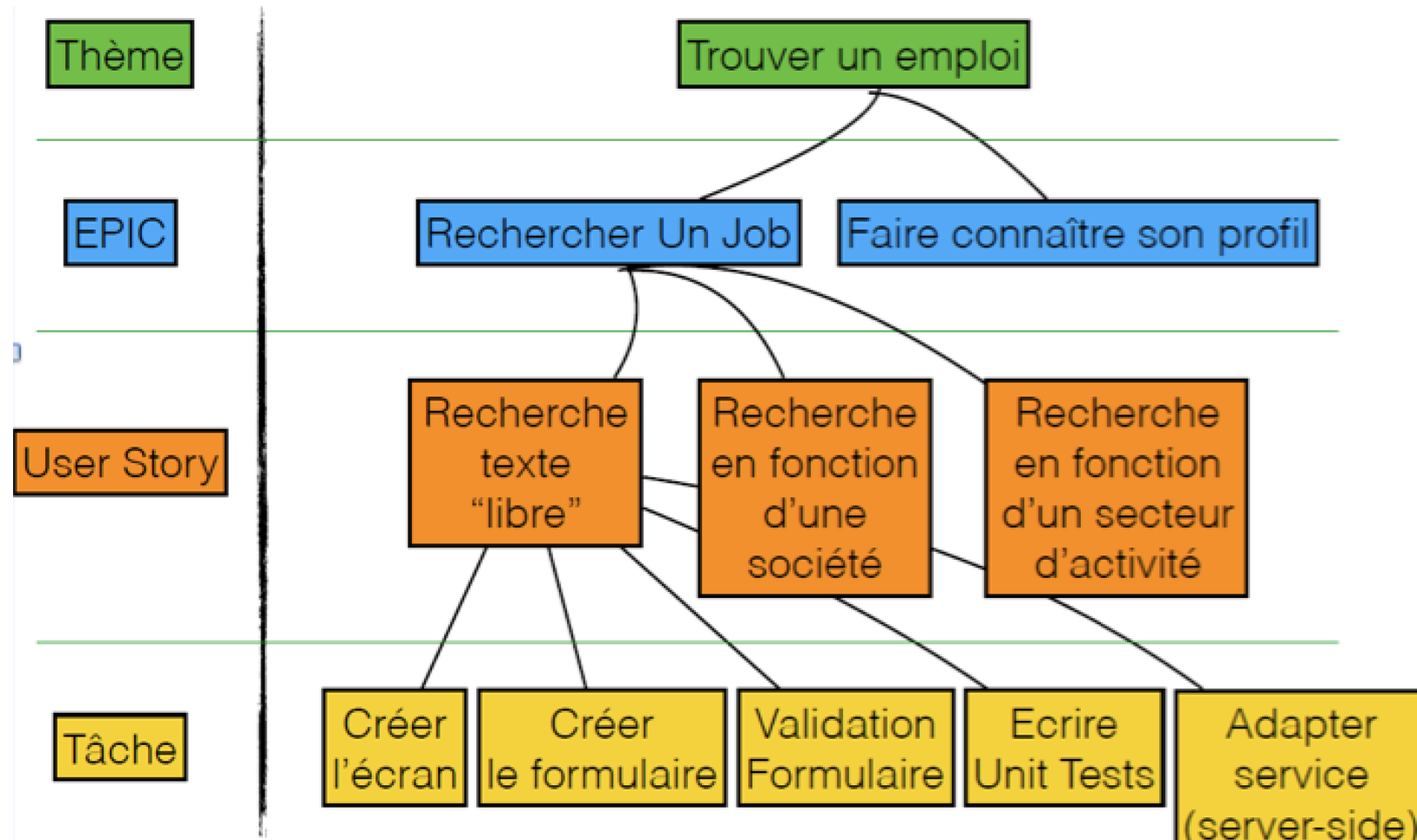
- « Sprint 1 » Dans lequel vous affecterez les US liés à l'épic « Navigation »
- « Sprint 2 » Dans lequel vous affecterez les US liés à l'épic « Vente en ligne »

6. Démarrez le premier sprint

Connectez-vous à l'instance JIRA



Créez votre projet

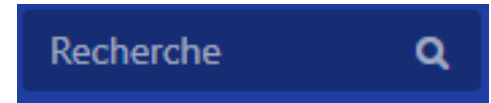


Recherches et filtres

Recherche rapide

Accessible directement depuis la barre de menu de l'outil et permet de rechercher des tickets.

- En tapant n'importe quel mot, JIRA recherchera automatiquement dans les champs *Résumé*, *Description* et *Commentaires* des tickets. Vous serez alors redirigé vers le Navigateur de demandes qui affichera la liste des demandes qui correspondent à la recherche.
- En tapant la clef d'une demande, on sera redirigé directement vers la demande.
- Recherche intelligente : On peut rechercher des demandes spécifiques en tapant de courtes requêtes JQL.



Recherche basique

Demandes > Chercher des demandes

En sélectionnant les différentes options dans les menus déroulants, il est possible d'affiner la recherche. Il est également possible d'ajouter plus de champs à la recherche en utilisant le menu déroulant Suite à côté du bouton Rechercher (la petite loupe).

Projet: Tous ▼ Type: Tous ▼ Etat: Tous ▼ Attribution: Tous ▼ Contient du texte Suite ▼ Recherche Avancé

Recherche avancée

Demandes > Chercher des demandes > Avancé

La recherche avancée est un champ auto-complété qui permet d'élaborer une requête complexe en utilisant JQL.

Toute requête basique peut être traduite en requête JQL en basculant en mode Avancé depuis le mode Basique.

Pourquoi basculer en mode avancé ? Notamment pour :

- Utiliser l'opérateur OU,
- Quand un champ de la requête n'est pas supporté par le mode basique.

✓ type = Story AND description is not EMPTY Recherche Basique

Exemples de requêtes JQL fréquemment utilisée :

Demandes liées

Obtenir la liste de toutes les demandes liées à une demande en particulier :

Fonction JQL `linkedIssues(issueKey)`

Ex. `issue in linkedIssues(« TP-345 »)`

Demandes liées par type de lien

Obtenir la liste des demandes liées à une demande spécifique avec un type de lien donné :

Fonction JQL `linkedIssues(issueKey, linkType)` .

Ex. `Issue in linkedIssues("TP-345", blocks)`

Retrouver les stories associées à une epic

`« Epic Ling » in (SCUR-24, SCUR-25)`

Il est nécessaire de spécifier les clefs des epics parentes.

Retrouver toutes les demandes créées depuis une demande particulière jusqu'à aujourd'hui

`Key >= VSD-30`

Retourne toutes les demandes créées depuis la VSD-30 jusqu'à aujourd'hui

NB. Pour retrouver toutes les demandes créées ou mises à jour depuis une certaine date ou pendant une période donnée, on peut utiliser les champs « Created » et « Updated ».



Créez des filtres

1. Créez un filtre pour toutes les demandes de votre projet de type « Récit ».
 - Effectuez votre recherche
 - Cliquez sur Enregistrez sous
 - Nom du filtre : Demandes Récit
2. Créer un second filtre pour toutes les demandes de votre projet de type « Bogue », à l'état « In progress ».
 - Effectuez votre recherche
 - Cliquez sur Enregistrez sous
 - Nom du filtre : Bug ouvert
3. Vous pouvez désormais accéder à vos recherches à partir de :
 - Demandes > Filtres > Demandes Récit / Bug ouvert
 - Demandes > Gérer les filtres > « Mes »

Demandes ▾Tableaux ▾Créer

Recherche actuelle

Chercher des demandes

DEMANDES RÉCENTES

TM2S-12 Création d'un compte u...

TM2S-13 Connexion

TM2S-14 Tableau de bord

TM2S-18 Annulation

TM2K-1 Tableau de bord

plus...

Importer des demandes à partir d'un fichier...

FILTRES

Mes demandes ouvertes

Que j'ai signalées

Bug ouvert

Demandes Récit

Gérer les filtres

CONFIGURATION DES UTILISATEURS

- I. Gérer les utilisateurs
- II. Gérer les groupes d'utilisateurs
- III. Gérer les systèmes d'autorisation

Gérer les utilisateurs

IL est possible d'ajouter, de supprimer ou de modifier des utilisateurs.

Chaque utilisateur est rattaché à un groupe. À sa création un utilisateur sera rattaché par défaut au groupe [jira-software-users](#).

Administration 🔍 Rechercher dans l'administration de Jira

Applications Projets Demandes Composants additionnels **Gestion des utilisateurs** Dernier rapport de mise à niveau Système

GESTION DES UTILISATEURS

Utilisateurs

Groupes

Serveur de l'utilisateur Jira

RÉPERTOIRES UTILISATEUR

Répertoires utilisateur





Utilisateurs

Inviter les utilisateurs Créer un utilisateur

Filtrer les utilisateurs Dans le groupe Accès à l'application État Utilisateurs par page

Le nom, nom d'utilisateur ou l'adresse 🔍 N'importe lequel Tous les utilisateurs Tous les utilisateurs 20 Filtre Réinitialiser

Affichage des utilisateurs 1 à 10 sur 10.

Nom complet	Nom d'utilisateur	Informations de connexion	Nom du groupe	Applications	Répertoire	Actions
 Auditeur1	Auditeur1 hhassine@henix.fr	Nombre: 20 Dernier: Il y a 3 heures	jira-software-users	Jira Software	Jira Internal Directory	Modifier 
 Auditeur2	Auditeur2 hhassine@henix.fr	Nombre: 3 Dernier: Il y a 4 heures	jira-software-users	Jira Software	Jira Internal Directo	Modifier
 Auditeur3	Auditeur3 hhassine@henix.fr	Nombre: 9 Dernier: 10/12/2024 10:00 AM	jira-software-users	Jira Software	Jira Internal Directory	Modifier ...

- Modifier les groupes d'utilisateurs
- Afficher les rôles du projet
- Supprimer l'utilisateur

Les utilisateurs peuvent être regroupés dans une même entité : le groupe

Administration

Rechercher dans l'administration de Jira

ApplicationsProjetsDemandesComposants additionnelsGestion des utilisateursDernier rapport de mise à niveauSystème

GESTION DES UTILISATEURS

Utilisateurs

Groupe

Serveur de l'utilisateur Jira

RÉPERTOIRES UTILISATEUR

Répertoires utilisateur

Groupe

Modifier les membres du groupe par lots

Filtrer le groupe

Le nom contient

Groupe par page

20

Ajouter un groupe

Nom

Ajouter un groupe

Affichage des groupes 1 vers 2 de 2.

Nom du groupe	Utilisateurs	Systèmes d'autorisation	Actions
jira-administrators	ADMINISTRATIONJIRA SOFTWARE	1	Modifier les membres Supprimer
jira-software-users	JIRA SOFTWARE	9	Modifier les membres Supprimer

Gérer les types de demandes

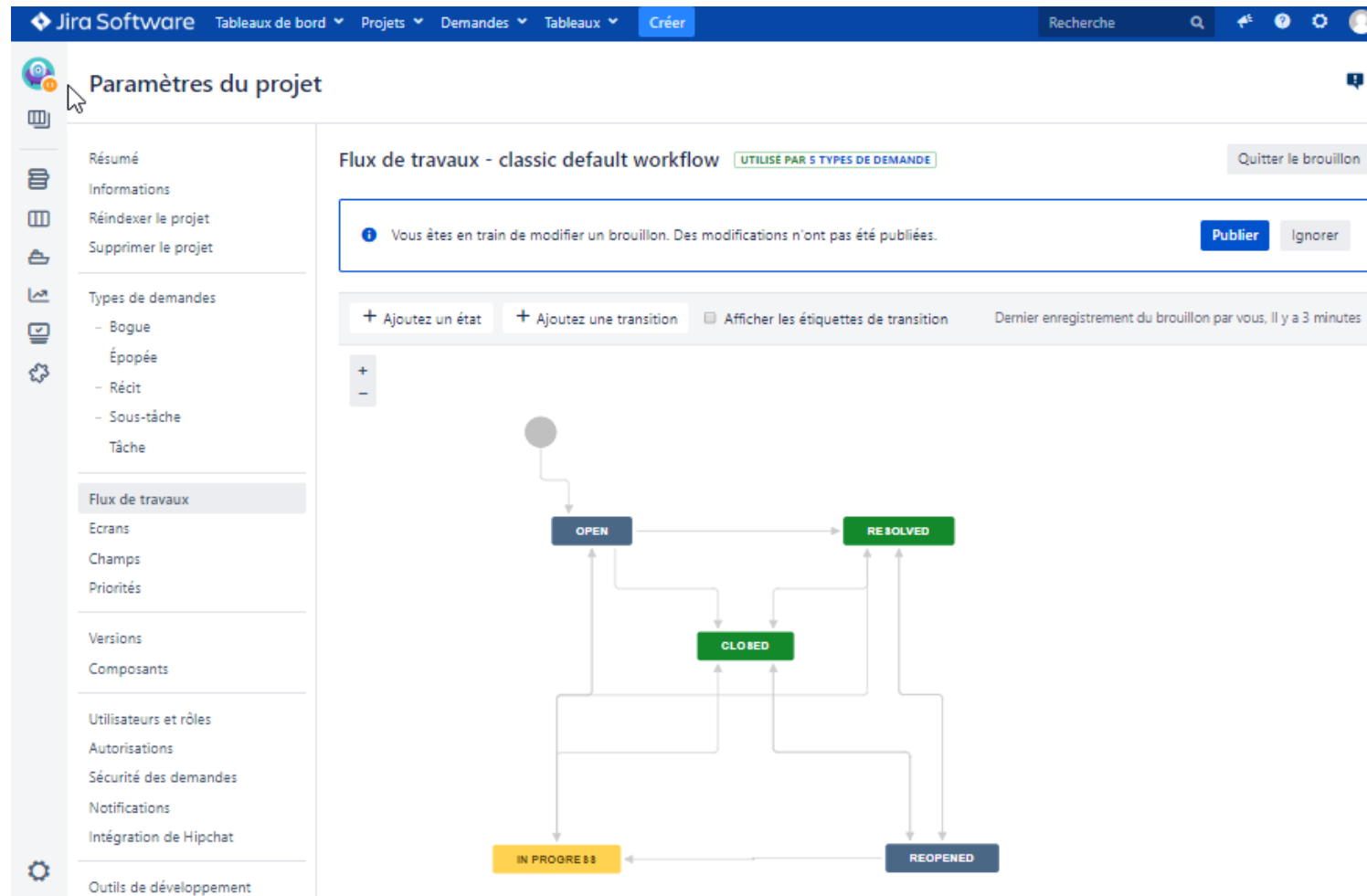
- Si on veut **ajouter un type** qui n'existe pas, aller dans configuration/Demandes/Type de demandes/Ajouter un type de demande
- Si on veut ajouter un ensemble de types de demande (« **Système de type de demande** ») que l'on pourrait proposer pour un projet, aller dans : configuration/Demandes/Systeme de type de demande/Ajouter Système de type de demande. **On peut directement l'associer à un projet**

The screenshot shows the Jira Software Administration interface. The top navigation bar includes 'Jira Software', 'Tableaux de bord', 'Projets', 'Demandes', 'Tableaux', and a 'Créer' button. A search bar is on the right. The main header is 'Administration' with a search prompt. Below this is a breadcrumb trail: 'Applications > Projets > Demandes > Composants additionnels > Gestion des utilisateurs > Dernier rapport de mise à niveau > Système'. The left sidebar lists various administration categories, with 'Types de demandes' selected. The main content area is titled 'Types de demandes' and contains a table of request types. A button 'Ajouter un type de demande' is visible in the top right of the table area. The table has columns for 'Nom', 'Type', 'Systèmes liés', and 'Actions'.

Nom	Type	Systèmes liés	Actions
Amélioration Amélioration ou perfectionnement apporté à une fonction ou tâche existante.	Standard	<ul style="list-style-type: none">TEST: Système de type de demande de développement logiciel	Modifier Supprimer Traduire
Bogue	Standard	<ul style="list-style-type: none">TEST: Système de type de demande ScrumTEST: Système de type de demande de développement logicielPK: Système de type de demande de KanbanFLUX: Système de type de demande de KanbanMAL: Système de type de demande de KanbanPW: Système de type de demande de KanbanCK: Système de type de demande de KanbanPP: Système de type de demande de Kanban	Modifier Supprimer Traduire

Gérer les flux de travaux

Un flux de travaux permet de déterminer les différentes étapes par la quelle une demande doit passer. Exemple : A faire; En cours d'analyse, En cours de recette etc...



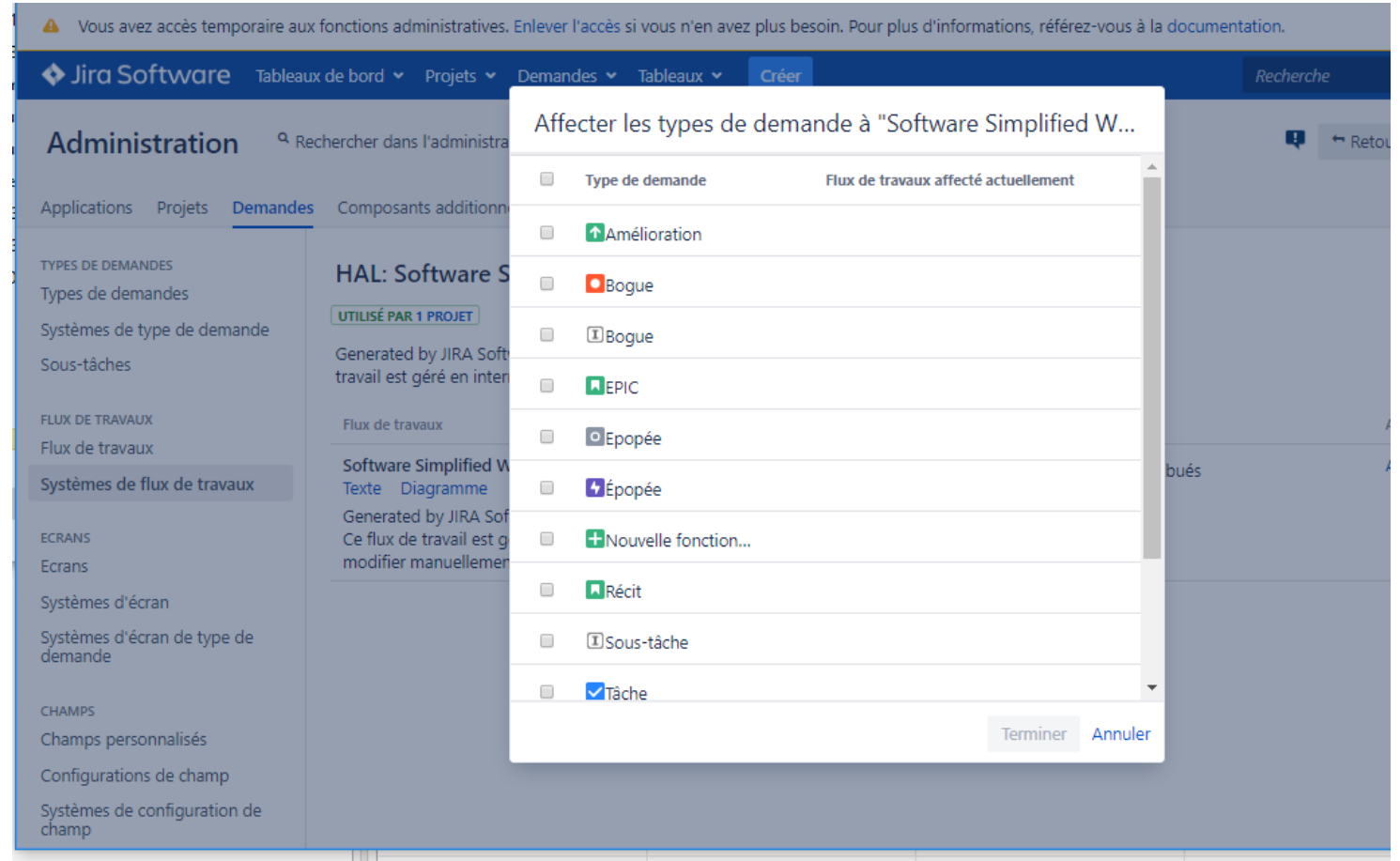
Un état est une étape par laquelle passe la demande.

Une transition permet de lier deux états.

Un système de flux de travaux peut être composé de de plusieurs flux de travaux .

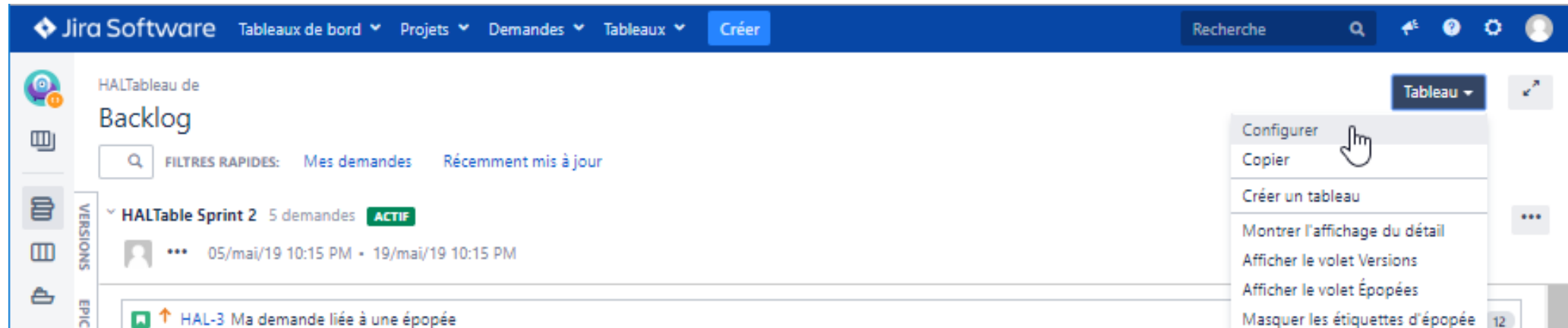
Chaque flux va être rattaché à des types de demandes.

Chaque type de demande pourra avoir son propre flux.



Une fois le système de flux rattaché au projet, afin de le voir implémenté, il est nécessaire de retourner sur le projet et d'effectuer une manipulation :

1. Sélectionner la configuration du tableau



2. Insérer les nouveaux états dans les colonnes du tableau

Jira Software Tableaux de bord Projets Demandes Tableaux Créer Recherche

Configurer HALTableau de

Retour dans le tableau Tableau

CONFIGURATION

- Général
- Colonnes**
- Couloirs
- Filtres rapides
- Couleurs de carte
- Mise en page de la carte
- Estimation
- Jours travaillés
- Affichage des détails de la demande

Gestion de colonne

Les colonnes peuvent être ajoutées, retirées, ordonnées. Elles sont basées sur les statuts de demandes, ceux-ci étant configurables par colonne. Des contraintes définissant le nombre minimal ou maximal de demandes peuvent être définies pour chacune de ces colonnes.

Contrainte de colonne: Aucune

Des contraintes peuvent être ajoutées aux colonnes sur le tableau, pour une statistique.

Jours dans la colonne: ☐

Affiche les points représentant le temps passé par un ticket dans une colonne spécifique d'un tableau Jira.

Flux de travaux simplifié: Simplifiez le flux de travaux

Simplifiez le flux de travail du projet **HalimPierre** afin qu'il soit géré par Jira Software. Cela permettra aux administrateurs de projet d'ajouter et de supprimer des états ci-dessous.

Ajoutez une colonne

Ouverte	A Analyser	Nouvelle colonne	Nouvelle colonne	Nouvelle colonne	États non mapp...
OUVERTE 13 Demandes	A ANALYSER ⚠ 0 Aucune demande	EN COURS ⚠ 0 Aucune demande	EN TEST ⚠ 0 Aucune demande	TERMINÉ ⚠ 0 Aucune demande	États ne contenant pas de demandes FERMÉE 0 Aucune demande ROUVERTE 0 Aucune demande

A blue arrow points from the 'EN COURS' state in the 'Nouvelle colonne' column to the 'ROUVERTE' state in the 'États non mapp...' column.

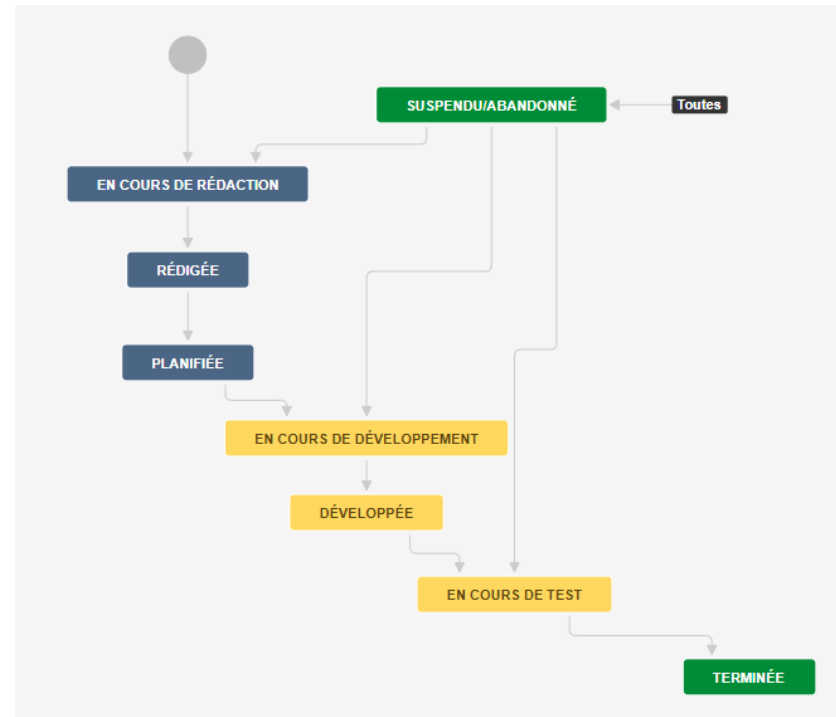
Créez un workflow personnalisé pour les demandes de type Récit/Épopée/Tâche et Sous-tâche

1. Administration > Projets > Sélectionnez votre projet
2. Flux de travaux
3. En haut à droite de la page, cliquez sur **Modifier**
4. Personnalisez le workflow en ajoutant des états et des transitions d'états. Les User Stories doivent passer par les étapes suivantes : Nouvelles, En cours d'analyse, En cours de développement, En cours de recette et enfin Terminée

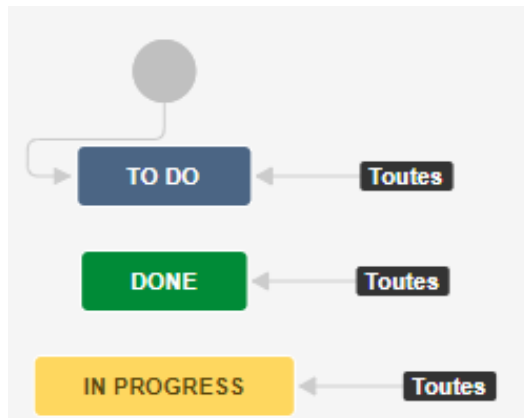


Mettez en place les flux de travaux pour votre projet

Exemple de flux de travail personnalisé



Flux de travail par défaut



Ajoutez ensuite un nouveau workflow pour les demandes de type « Bogue »

1. Ajoutez un nouveau flux de travaux > Ajouter existant
2. Flux de travaux Jira > Suivant
3. Affectez le type de demande « Bogue »

Gérer les écrans

Gérer les écrans

Gérer les écrans eux mêmes

Un écran reprend tout ce que l'on veut rattacher à un type de demande (Bug, US, Epic...)

- On peut ajouter des champs existants, des nouveaux onglets ou changer l'ordre d'apparition.
- Aller dans configuration/Demandes/Ecrans/Ajouter un écran

Les champs que l'on peut ajouter sont ceux déjà configurées dans Jira

Administration

Rechercher dans l'administration de Jira

Applications Projets **Demandes** Composants additionnels Gestion des utilisateurs Dernier rapport de mise à niveau Système

TYPES DE DEMANDES

Types de demandes

Systèmes de type de demande

Sous-tâches

FLUX DE TRAVAUX

Flux de travaux

Systèmes de flux de travaux

ECRANS

Ecrans

Systèmes d'écran

Systèmes d'écran de type de demande

CHAMPS

Champs personnalisés

Configurations de champ

Systèmes de configuration de champ

PRIORITÉS

Priorités

Systèmes de priorité

CARACTÉRISTIQUES DES DEMANDES

Suivi temporel

Création de liens entre des demandes

ATTRIBUTS DES DEMANDES

Etats

Résolutions

Configurer l'écran

Cette page montre la façon dont les champs sont organisés sur l'écran **CHAB: Écran de bogue Kanban**.

Remarque : lorsque l'écran s'affiche pour un utilisateur, seuls les champs non cachés que l'utilisateur a l'autorisation de modifier apparaîtront.

||| Onglet de champ

Field	Type
Résumé	Champ de système
Type de demande	Champ de système
Rapporteur	Champ de système
Composants	Champ de système
Descriptif	Champ de système
Version(s) corrigée(s)	Champ de système
Priorité	Champ de système
Étiquettes	Champ de système
Niveau de sécurité	Champ de système
Environnement	Champ de système
Pièce jointe	Champ de système
Affecte la/les version(s)	Champ de système
Demandes liées	Champ de système
Responsable	Champ de système
Lien d'épopée	Relation du lien d'épopée

Nom du champ

Ajouter

Les systèmes d'écran permettent de choisir les **écrans** à afficher pour **chaque opération de demande**. Les systèmes d'écran sont mappés aux demandes à l'aide de systèmes d'écran de type de demande, qu'il est possible d'associer à un ou à plusieurs projets.

The screenshot shows the 'Gérer les systèmes d'écran' (Manage screen systems) interface. On the left, a sidebar contains a tree view with the following items: 'FLUX DE TRAVAUX' (containing 'Flux de travaux' and 'Systèmes de flux de travaux'), 'ECRANS' (containing 'Ecrans' and 'Systèmes d'écran'), and 'CHAMPS' (containing 'Champs personnalisés'). The 'Systèmes d'écran' item is selected. The main content area has a header 'mappage précis dans ce système.' followed by the text 'Pour activer ce système d'écran, ma plusieurs projets.' and a note 'Remarque : Un système d'écran ne'. Below this is a bullet point '• Afficher tous les systèmes d'. The main area is divided into two sections: 'Opération de demande' and 'Ecran'. The 'Opération de demande' section has a 'Par défaut' label and the text 'Utilisé pour toutes les opérations non mappées.' The 'Ecran' section is currently empty. A modal dialog titled 'Associer une opération de demande à un écran' is open in the foreground. It contains a table with two columns: 'Opération de demande' and 'Ecran'. The first row has a dropdown menu in the 'Opération de demande' column with the value 'Créer une demande' selected. The 'Ecran' column for this row is empty. Below the table, the text 'Ecran à afficher pour l'opération de demande sélectionnée.' is displayed. At the bottom right of the modal, there are two buttons: 'Ajouter' and 'Annuler'.

FLUX DE TRAVAUX
Flux de travaux
Systèmes de flux de travaux

ECRANS
Ecrans
Systèmes d'écran

Systèmes d'écran de type de demande

CHAMPS
Champs personnalisés

mappage précis dans ce système.
Pour activer ce système d'écran, ma plusieurs projets.
Remarque : Un système d'écran ne
• **Afficher tous les systèmes d**

Opération de demande

Par défaut
Utilisé pour toutes les opérations non mappées.

Ecran

Associer une opération de demande à un écran

Opération de demande	Ecran
Créer une demande	

Ecran à afficher pour l'opération de demande sélectionnée.

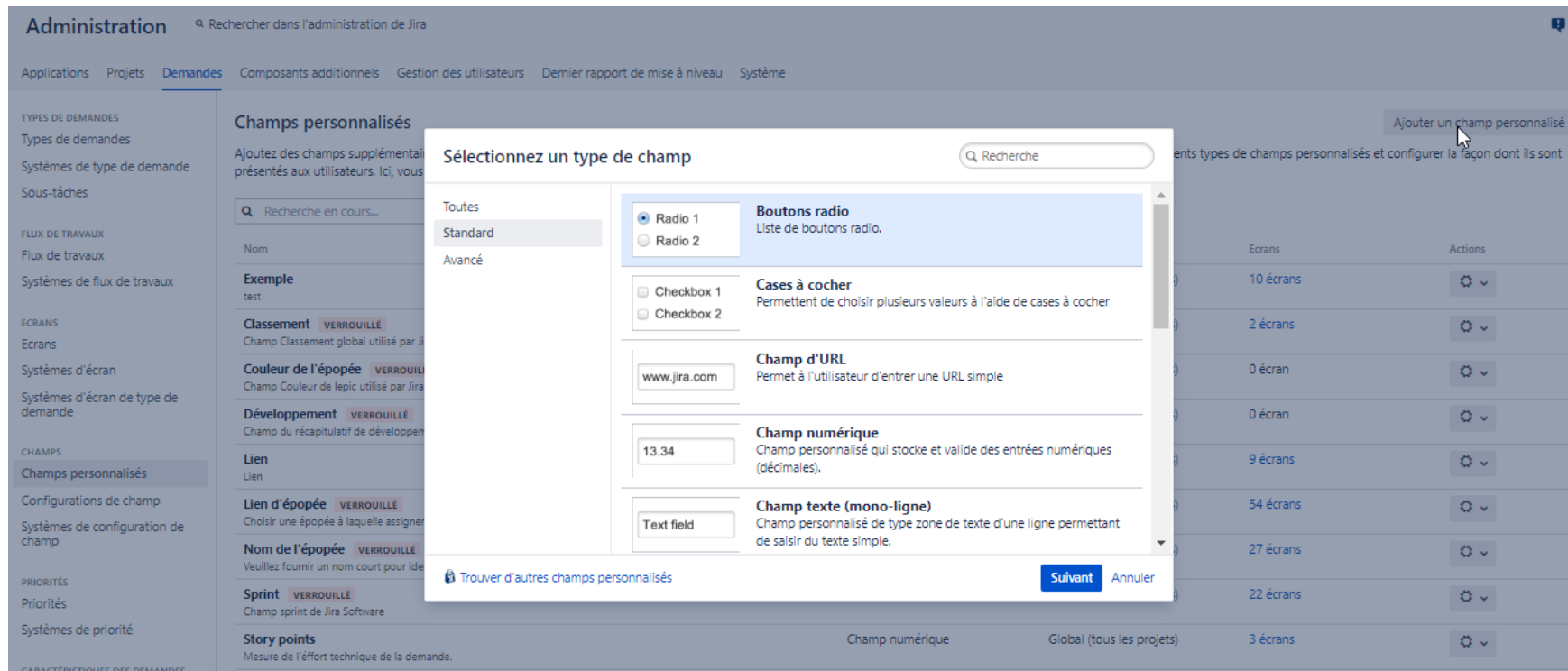
Ajouter Annuler

Un système d'écran de type de demande permet de choisir le système d'écran à utiliser pour **chaque type de demande**.
Allez dans Configuration/Demande/Système d'écran de type de demande/Configurer/Associer un type de demande à un systèmes d'écran

Associer un type de demande à un système d'écran

Gérer les champs

Une fois les écrans créés, il est possible d'affiner la configuration des champs sur chaque écran.
En plus des champs proposés par défaut dans JIRA, il est possible d'en ajouter de nouveaux types.



Remarque : En créant le champ personnalisé, on l'affecte directement aux écrans où on veut le voir.

Une configuration de champ permet de modifier le fonctionnement d'un champ. Elle permet à Jira de gérer un champ particulier.

66

Une fois une configuration de champ sélectionnée, chaque champ ainsi que tous les écrans auxquels ils sont rattachés sont présentés.

On peut ainsi ajuster notre configuration de champs en rendant **obligatoire**, **facultatif** ou **même masquer** chacun des ces champs existant.

Le fait de masquer un champ par exemple, dans le cadre de notre configuration de champ, va le masquer de tous les écrans auxquels il est rattaché.

Ainsi tous les projets rattachés à cette configuration de champs masquerons ce champ. Cela pourra être tout de même affiné grâce au systèmes de configuration de champs...

Administration Rechercher dans l'administration de Jira

Applications Projets **Demandes** Composants additionnels Gestion des utilisateurs Dernier rapport de mise à niveau Système

TYPES DE DEMANDES

- Types de demandes
- Systèmes de type de demande
- Sous-tâches

FLUX DE TRAVAUX

- Flux de travaux
- Systèmes de flux de travaux

ECRANS

- Ecrans
- Systèmes d'écran
- Systèmes d'écran de type de demande

CHAMPS

- Champs personnalisés
- Configurations de champ**
- Systèmes de configuration de champ

Afficher la configuration de champ

Le tableau ci-dessous montre tous les champs configurés dans Jira ainsi que leurs propriétés pour **Configuration de champ Halim**.

Cette page permet de rendre obligatoire certains champs, de les masquer ou des les afficher et d'indiquer leur description. Il est également possible de modifier les écrans sur lesquels apparaît un champ en utilisant le lien "Ecrans" en regard de chaque champ.

- Afficher toutes les configurations de champ

Nom	Ecrans	Actions
Affecte la/les version(s) [Générateur de rendu à saisie semi-automatique]	<ul style="list-style-type: none">CHAB: Écran de bogue KanbanCK: Écran de bogue KanbanCS: Écran Scrum BugDefault ScreenFLUX: Écran de bogue KanbanHAL: Écran Scrum Bug (Onglet de champ)HIND: Écran de bogue KanbanMAL: Écran de bogue KanbanMSA: Écran Scrum BugNON: Écran Scrum BugPC: Écran de bogue KanbanPH: Écran de bogue KanbanPK: Écran de bogue KanbanPO: Écran de bogue KanbanPOP2: Écran Scrum BugPOP: Écran Scrum BugPP: Écran de bogue KanbanPPC: Écran Scrum Bug	Modifier Masquer Obligatoire Ecrans Générateurs de rendu

Plusieurs configurations de champ peuvent être assemblées pour constituer un système de configuration de champ. Il est possible d'affecter pour chaque type de demande une configuration de champ particulière.

The screenshot displays the Jira Administration interface. The left sidebar contains a navigation menu with categories: TYPES DE DEMANDES, FLUX DE TRAVAUX, ECRANS, and CHAMPS. The 'Demandes' tab is selected under 'TYPES DE DEMANDES'. The main content area is titled 'Configurer le système de configuration de champ : Exemple de système de configuration de champs'. A modal dialog is open in the center, titled 'Associer un type de demande à une configuration de ch...'. It features two dropdown menus: 'Type de demande' with 'Boque' selected, and 'Configuration de champ' with 'test' selected. Below the dropdowns are 'Ajouter' and 'Annuler' buttons. The background interface shows a table with columns 'Type de demande', 'Configuration de champ', and 'Actions'. The first row is labeled 'Par défaut' and 'Default Field Configuration'.

Administration Rechercher dans l'administration de Jira Retour au projet: HalimPierre

Applications Projets **Demandes** Composants additionnels Gestion des utilisateurs Dernier rapport de mise à niveau Système

TYPES DE DEMANDES
Types de demandes
Systèmes de type de demande
Sous-tâches

FLUX DE TRAVAUX
Flux de travaux
Systèmes de flux de travaux

ECRANS
Ecrans
Systèmes d'écran
Systèmes d'écran de type de demande

CHAMPS
Champs personnalisés
Configurations de champ
Systèmes de configuration de champ

Configurer le système de configuration de champ : Exemple de système de configuration de champs

Associer un type de demande à une configuration de champ

Associer un type de demande à une configuration de ch...

Type de demande Boque

Configuration de champ Default Field Configuration
Default Field Configuration
Configuration de champ Halim
test

Ajouter Annuler

Type de demande	Configuration de champ	Actions
Par défaut Utilisé pour tous les types de demandes non mappés.	Default Field Configuration	Modifier

Applications

Projets

Demandes

Composants additionnels

Gestion des utilisateurs

Dernier rapport de mise à niveau

Système

Type de demandes

Type de demandes

Systèmes de type de demande

Sous-tâches

Flux de travaux

Flux de travaux

Systèmes de flux de travaux

Ecrans

Ecrans

Systèmes d'écran

Systèmes d'écran de type de demande

Champs personnalisés

Configurations de champ




Systèmes de configuration de champ

Configurer le système de configuration de champ : Système de configuration de champs Halim

Associer un type de demande à une configuration de champ ?

i

Ce système peut être utilisé par un ou plusieurs projets. La configuration de champ indiquée pour chaque type de demande sera appliquée aux demandes de ces projets.
L'entrée *Par défaut* indique quelle configuration de champ sera utilisée pour tout type de demande qui n'a pas été explicitement mappé avec une configuration de champ.
[Afficher tous les systèmes de configuration de champ.](#)

Type de demande	Configuration de champ	Actions	
Par défaut Utilisé pour tous les types de demandes non mappés.	Default Field Configuration	Modifier	
<input checked="" type="checkbox"/> Tâche	Configuration de champ Halim	Modifier	Supprimer
 EPIC	Configuration de champ Halim	Modifier	Supprimer
 USER STORY	Default Field Configuration	Modifier	Supprimer
 Bogue	Configuration de champ Halim	Modifier	Supprimer

69

Gérer les priorités

Gérer les priorités

Gérer les priorités elles-mêmes

Permet d'ajouter un niveau de priorité / Modifier un niveau de priorité / Supprimer un niveau de priorité

Ces priorités seront celles présentées par le système de priorité par défaut.

Administration

Rechercher dans l'administration de Jira

Applications Projets **Demandes** Composants additionnels Gestion des utilisateurs Dernier rapport de mise à niveau Système

TYPES DE DEMANDES

Types de demandes

Systèmes de type de demande

Sous-tâches

FLUX DE TRAVAUX

Flux de travaux

Systèmes de flux de travaux

ECRANS

Ecrans

Systèmes d'écran

Systèmes d'écran de type de demande

CHAMPS

Champs personnalisés

Configurations de champ

Systèmes de configuration de champ

PRIORITÉS

Priorités

Systèmes de priorité

Afficher les priorités

Ajouter une priorité



Le tableau ci-dessous affiche les priorités utilisées dans cette version de Jira, des plus élevées aux plus faibles.

Icône et nom	Description	Couleur	Ordre	Associé à	Actions
↑ Highest	This problem will block progress.		↓	2 systèmes	Modifier Supprimer
↑ High	Serious problem that could block progress.		↑ ↓	2 systèmes	Modifier Supprimer
↑ Medium	Has the potential to affect progress.		↑ ↓	2 systèmes	Modifier Supprimer
↓ Low	Minor problem or easily worked around.		↑ ↓	2 systèmes	Modifier Supprimer
↓ Lowest	Trivial problem with little or no impact on progress.		↑ ↓	1 système	Modifier Supprimer
⛔ Undefined			↑	2 systèmes	Modifier Supprimer

Permet de mettre en place un système de priorité qui sera ensuite affecté à un projet

Administration

Rechercher dans l'administration de Jira

Applications

Projets

Demandes

Composants additionnels

Gestion des utilisateurs

Dernier rapport de mise à niveau

Système

TYPES DE DEMANDES

Types de demandes

Systèmes de type de demande

Sous-tâches

FLUX DE TRAVAUX

Flux de travaux

Systèmes de flux de travaux

ECRANS

Ecrans

Systèmes d'écran

Systèmes d'écran de type de demande

CHAMPS

Champs personnalisés

Configurations de champ

Systèmes de configuration de champ

PRIORITÉS

Priorités

Systèmes de priorité

Ajouter un système de priorité

Nom du système*

Système de priorité Halim

Description

Sélectionner priorités

Modifiez l'ordre des priorités, en utilisant glisser-déposer pour les organiser dans l'ordre voulu.
Vous pouvez aussi glisser-déposer les priorités d'une liste à une autre pour les ajouter ou les supprimer.

Selectionne priorités

Retirer tout

↑ Highest

↑ High

↑ Medium

↓ Low

⊘ Undefined

Disponible priorités

Ajouter tout

↓ Lowest

Priorité par défaut

Aucune

Ajouter

Annuler