

PROJET TATAMIS

ANGER BENOIT - BOURGINE BRUNO

Copyright © 2021-2022 ANGER Benoit - BOURGINE Bruno

**Copying prohibited**

All rights reserved. No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording, or by any information storage or retrieval system, without the prior written permission of the publisher.

Art. No xxxxx

ISBN xxx-xx-xxxx-xx-x

Edition 0.0

Cover design by Utagawa Hiroshige

Published by

Printed in

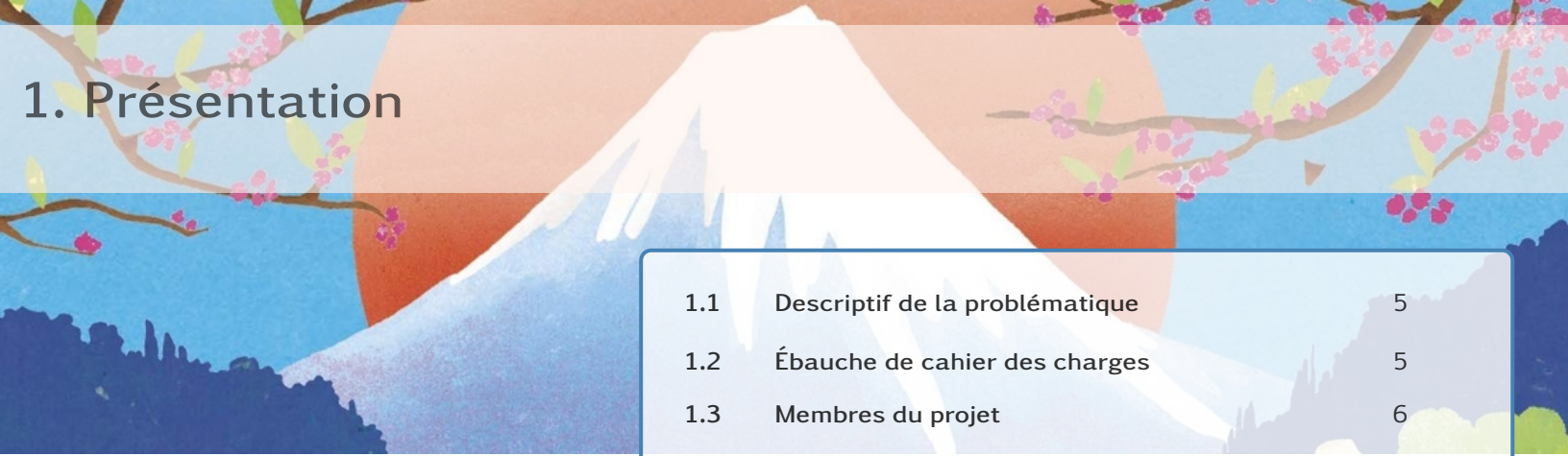




# Contents

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>Présentation</b>                      | <b>5</b>  |
| 1.1      | Descriptif de la problématique           | 5         |
| 1.2      | Ébauche de cahier des charges            | 5         |
| 1.3      | Membres du projet                        | 6         |
| <b>2</b> | <b>Méthodologie</b>                      | <b>7</b>  |
| 2.1      | Choix de méthodologie                    | 7         |
| 2.2      | Les événements                           | 7         |
| 2.2.1    | Les sprints et les phases du projet      | 7         |
| 2.2.2    | La session de planning du sprint         | 8         |
| 2.2.3    | "L'hebdo"                                | 8         |
| 2.2.4    | La revue du sprint                       | 9         |
| 2.2.5    | La retrospective                         | 9         |
| 2.3      | Les éléments de formalisation            | 9         |
| 2.3.1    | Cahier des charges global (et les Epics) | 9         |
| 2.3.2    | Le backlog global (et les User Stories)  | 9         |
| 2.3.3    | Roadmap globale                          | 9         |
| 2.3.4    | Le backlog d'un sprint (et les Tâches)   | 9         |
| 2.4      | Les rôles                                | 10        |
| 2.4.1    | Description des rôles                    | 10        |
| 2.4.2    | Assignation des rôles                    | 10        |
| 2.5      | L'approche de test                       | 10        |
| <b>3</b> | <b>Choix des outils et langages</b>      | <b>11</b> |

|     |   |    |
|-----|---|----|
| 4   | Cahier des charges .....                | 12 |
| 4.1 | Démarche .....                          | 12 |
| 4.2 | Fonctionnalités essentielles .....      | 12 |
| 4.3 | Fonctionnalités optionnelles .....      | 13 |
| 5   | Backlog global .....                    | 14 |
| 6   | Roadmap globale .....                   | 15 |
| 7   | Backlog de chaque sprint .....          | 16 |
| 8   | Explication de l'algorithme .....       | 17 |
| 9   | Choix de programmation .....            | 18 |
| 10  | Tests .....                             | 19 |
| 11  | Gestion des versions concurrentes ..... | 20 |



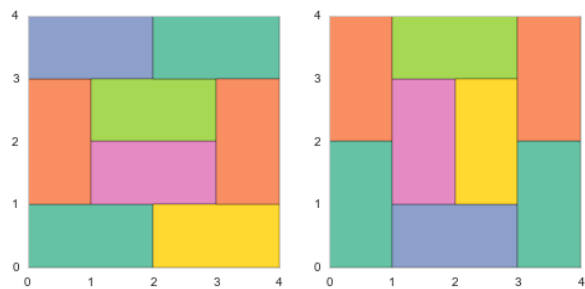
# 1. Présentation

|     |                                |   |
|-----|--------------------------------|---|
| 1.1 | Descriptif de la problématique | 5 |
| 1.2 | Ébauche de cahier des charges  | 5 |
| 1.3 | Membres du projet              | 6 |

*Ce premier chapitre sera l'occasion d'introduire le projet tel qu'il a été initié et de détailler les outils et les méthodes envisagées pour son développement.*

## 1.1 Descriptif de la problématique

Le pavage du plan avec des rectangle est un problème classique et déjà largement documenté, mais je souhaite l'aborder par un aspect très concret : étant donné un nombre de tatamis, quelles sont les configurations possibles.



C'est un problème que rencontre notamment toute personne qui se retrouve à devoir installer un dojo. Il existe une contrainte de base qui est que 4 tatamis ne rejoignent jamais en un même coin. Mais on peut en ajouter d'autres : possibilité de demi-tatamis (carré), répartition des couleurs, répartition de l'usure...

## 1.2 Ébauche de cahier des charges

Utilisateur final : gestionnaire de dojo

L'interface utilisateur devra comprendre un menu de paramétrage basique : nombre de tatamis, contraintes géométriques, contraintes d'aspect général; ainsi qu'un affichage des dispositions envisageables.

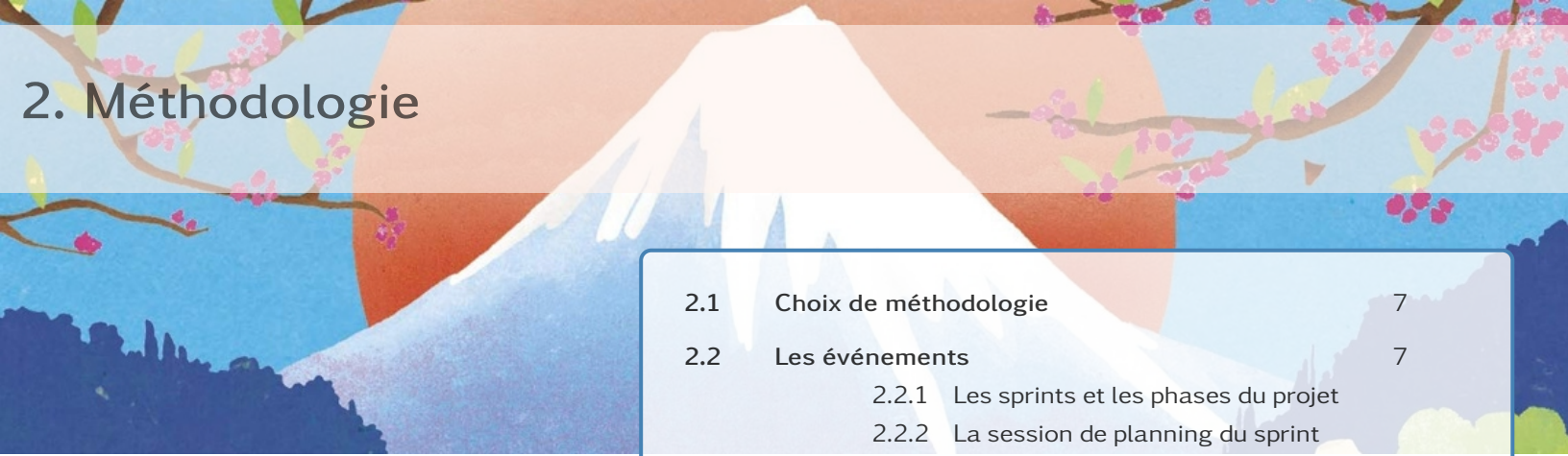
L'utilisateur devra pouvoir saisir :

- le nombre et le type (entier/demis) de tatamis à disposition
- leurs dimensions
- leurs couleurs
- éventuellement leur état (on dispose de préférence les plus usés en périphérie)
- les dimensions du dojo

L'affichage proposera différentes disposition selon les contraintes imposées.

## 1.3 Membres du projet

---



# 2. Méthodologie

|       |  |   |
|-------|--|---|
| 2.1   | Choix de méthodologie                    | 7 |
| 2.2   | Les événements                           | 7 |
| 2.2.1 | Les sprints et les phases du projet      |   |
| 2.2.2 | La session de planning du sprint         |   |
| 2.2.3 | "L'hebdo"                                |   |
| 2.2.4 | La revue du sprint                       |   |
| 2.2.5 | La retrospective                         |   |
| 2.3   | Les éléments de formalisation            | 9 |
| 2.3.1 | Cahier des charges global (et les Epics) |   |
| 2.3.2 | Le backlog global (et les User Stories)  |   |
| 2.3.3 | Roadmap globale                          |   |
| 2.3.4 | Le backlog d'un sprint (et les Tâches)   |   |

|     |                       |     |           |    |
|-----|-----------------------|-----|-----------|----|
| 2.1 | Choix de méthodologie | 2.4 | Les rôles | 10 |
|-----|-----------------------|-----|-----------|----|

La méthodologie agile nous paraît très adaptée au développement de notre programme. En effet, la méthodologie agile:

|     |                    |    |
|-----|--------------------|----|
| 2.5 | L'approche de test | 10 |
|-----|--------------------|----|

- Est particulièrement adaptée à la résolution de problème complexes et incertaines où l'on ne sait pas forcément avec précision l'objectif final ou la manière d'y arriver, ce qui est notre cas
- Est basée sur l'itération avec la production de livrables testables à la fin de chaque *sprint*, ce qui semble adapté pour produire nos différentes versions (alpha, beta...)
- Est basée sur une équipe multidisciplinaire qui couvre toutes les compétences pour produire un produit fini et qui s'auto organise, ce qui paraît également adapté à notre contexte

La méthodologie agile, étant particulièrement adaptée aux situations d'incertitude, préconise une planification au fur et à mesure de temps, plutôt que d'importantes et lourdes activités de planification en début de projet car les informations manquent pour cette planification totale en amont.

Pour nous donner un cadre, nous nous inspirons très fortement du schéma "Scrum" et du guide Scrum1, tout en l'adaptant à notre situation présente avec des ressources limitées et des contraintes particulières.

## 2.2 Les événements

### 2.2.1 Les sprints et les phases du projet

Les sprints sont des périodes de développement ayant un objectif précis et permettant d'arriver à une version du programme. Compte tenu du planning imposé par l'exercice, les sprints seront de durée variable et d'une durée légèrement supérieure à un mois (contrairement à ce

qui est suggéré par le schéma Scrum).

Les sprints commencent par la session de planning et se terminent par la revue et la rétrospective (événements détaillés ci-après). Elles comprennent également des activités de 'raffinement' ou de préparation du prochain sprint, pour qu'un nouveau sprint puisse commencer immédiatement après la clôture du précédent sprint.

Quatre sprints sont programmés pour le projet aboutissant aux versions Alpha, Beta, Release Candidate et Production.

*Nb: Une phase additionnelle de pré-développement aura lieu en amont pour la préparation du projet, mais est organisée de manière ad-hoc et ne peut être considérée comme un sprint. Cette phase a pour objectif d'analyser la demande (le cahier des charges), de déterminer la méthodologie et gouvernance et de préparer le développement pour aboutir sur une roadmap, un plan de développement global du programme qui sera bien sûr affiné au cours du temps.*

### 2.2.2 La session de planning du sprint

Pour chaque sprint la session de planning permet de déterminer:

- Le 'Quoi': quels éléments du backlog global peuvent être embarqué dans ce sprint pour créer le Sprint backlog
- Le 'Comment': comment chaque élément du Sprint backlog seront techniquement traités
- Le 'Pourquoi': quel objectif pour le sprint, sachant que chaque sprint doit délivrer un produit qui peut être limité en fonctionnalités mais qui fonctionne

Les décisions sont prises de la manière suivante:

- Quoi et pourquoi :  
Les propositions d'éléments à ajouter et d'objectif du sprint viennent du product owner. L'équipe de développement prend ensuite la décision de manière souveraine et autonome en session de planification.
- Comment :  
Chaque élément du Sprint backlog est discuté techniquement pour le décomposer en plus petites tâches qui feront l'objet de tickets.  
En cas de manque de compétences techniques, des tickets sont prévus pour la réalisation de recherches.  
Cette session couvre également les questions d'architecture du programme: choix du nombre de classes, de leurs interactions...

### 2.2.3 "L'hebdo"

Compte tenu de la situation particulière, un point de contact quotidien comme préconisé par le schéma Scrum n'est pas envisageable. Il sera remplacé par:



- Un point hebdomadaire facilité par le Scrum master pour un focus particulier sur l'échange, les challenges et solutions.
- La revue des tâches et le statut sont quand à eux discutés à travers:
  - Une communication continue sur Slack
  - Une mise à jour en direct des avancées sur Redmine, directement sur les tâches. Redmine apportant la visibilité nécessaire à tous les membres de l'équipe pour comprendre le statut rapidement grâce à la combinaison du diagramme Gantt, d'un tableau Kanban des tâches et d'un tableau de roadmap qui suit le pourcentage de complétude du backlog du sprint

#### 2.2.4 La revue du sprint

Chaque sprint se termine par une revue du produit livré à la fin du sprint. Les fonctionnalités développées sont discutées, ainsi que les challenges rencontrés. L'équipe commence à se projeter également sur le prochain sprint et ce qu'il faut faire ensuite.

#### 2.2.5 La retrospective

L'objectif de cette réunion est une introspection pour une amélioration continue notamment de la collaboration au sein de l'équipe, les processus et les outils. La session est facilitée par le Scrum master et se déroule selon les principes suivants:

- Discussions autour de 3 blocs successivement:
  - 'Garde': ce que l'on considère adapté et à conserver dans le futur
  - 'Start': ce que l'on souhaite commencer à faire pour améliorer la situation
  - 'Stop': ce que l'on souhaite arrêter sur un constat d'échec
- Étapes de chaque blocs:
  - Réflexion individuelle de points/idées à ajouter pour le bloc discute
  - Discussion de groupe pour regrouper les points mentionnés par thème
  - Résumé des actions/idées retenues
- La session se termine par la détermination du plan d'amélioration regroupant les idées retenues et en ajoutant une composante de temps et responsabilité des actions
- Outil utilisé pour la session: Miro (outil interactif et collaboratif permettant notamment de créer et arranger facilement des 'post it' représentant les points/idees)

### 2.3 Les éléments de formalisation

---

#### 2.3.1 Cahier des charges global (et les Epics)

#### 2.3.2 Le backlog global (et les User Stories)

#### 2.3.3 Roadmap globale

#### 2.3.4 Le backlog d'un sprint (et les Tâches)

## 2.4 Les rôles

---

### 2.4.1 Description des rôles

### 2.4.2 Assignment des rôles

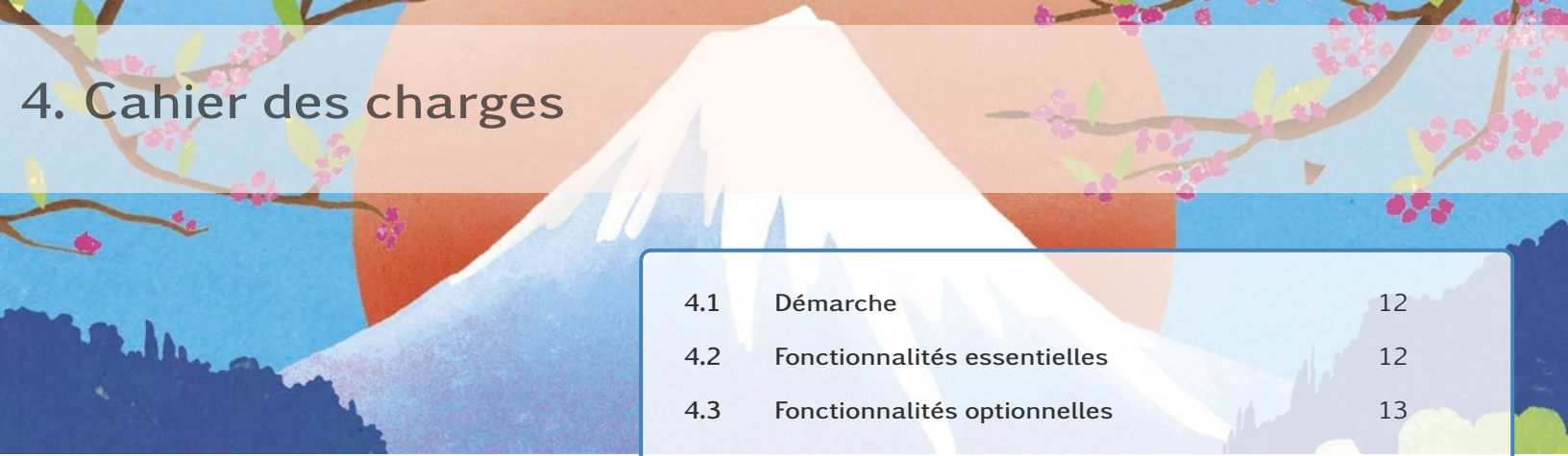
## 2.5 L'approche de test

---



### 3. Choix des outils et langages





# 4. Cahier des charges

|     |                              |    |
|-----|------------------------------|----|
| 4.1 | Démarche                     | 12 |
| 4.2 | Fonctionnalités essentielles | 12 |
| 4.3 | Fonctionnalités optionnelles | 13 |

## 4.1 Démarche

La problématique initiale telle qu'elle a été énoncée est la suivante : *étant donné un nombre de tatamis, quelles sont les configurations possibles ?*

Nous allons ici détaillé plus précisément le cahier des charges de l'application. Nous listerons les *epics* afin d'en déduire les *users stories* et leurs tâches afférentes, et ainsi établir la *roadmap* de notre projet. Le cahier des charges est formulé du point de vue de l'utilisateur final, les *epics* étant rédigées sous la forme : *en tant que ... ,je souhaite ... ,afin de ....*

Par ailleurs afin de prioriser les demandes, nous classerons les fonctionnalités en deux catégories :

- essentielles (must have)
- optionnelles (nice to have)

## 4.2 Fonctionnalités essentielles

- *En tant que* gestionnaire de dojo, *je souhaite* savoir s'il existe une solution utilisant l'ensemble de mes tatamis, *afin de* savoir si je pourrai tous les utiliser.
- *En tant que* gestionnaire de dojo, *je souhaite* connaître le nombre de tatamis utilisables *afin de* n'en déployer que le nombre nécessaire.
- *En tant que* gestionnaire de dojo, *je souhaite* visualiser l'ensemble des dispositions possibles, modulo une rotation ou une symétrie *afin de* ne voir sur l'écran que les solutions réellement différentes.
- *En tant que* gestionnaire de dojo, *je souhaite* pouvoir renseigner le nombre de tatamis dont je dispose, *afin de* d'obtenir une solution adaptée à mon matériel.



- *En tant que* gestionnaire de dojo, *je souhaite* pouvoir renseigner les dimensions de mon dojo *afin d'* obtenir une solution adaptée à l'espace dont je dispose.
- *En tant que* gestionnaire de dojo, *je souhaite* voir afficher les dimensions (longueur, largeur, surface) des dispositions proposées *afin d'* exploiter aux mieux l'espace disponible à l'intérieur et à l'extérieur du tatamis.

### 4.3 Fonctionnalités optionnelles

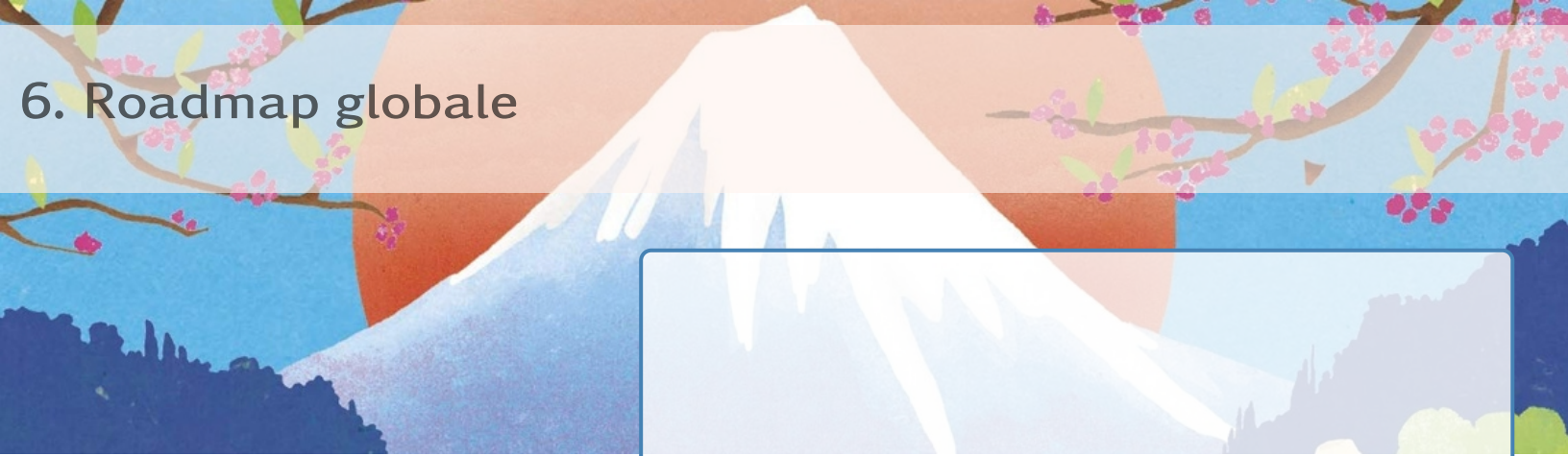
---

- *En tant que* gestionnaire de dojo, *je souhaite* intégrer la possibilité d'utiliser des demi-tatamis *afin de* remplir au mieux l'espace disponible.
- *En tant que* gestionnaire de dojo, *je souhaite* utiliser un nombre donné de de mi-tatamis *afin de* pouvoir exploiter tous les demi-tatamis dont je dispose.
- *En tant que* gestionnaire de dojo, *je souhaite* pouvoir modifier les dimensions d'un tatamis *afin de* obtenir des propositions correspondant à mon matériel.
- *En tant que* gestionnaire de dojo, *je souhaite* pouvoir créer des catégories de couleurs de tatamis *afin de* visualiser des propositions de placement avec les couleurs réelles.
- *En tant que* gestionnaire de dojo, *je souhaite* pouvoir renseigner des critères de placement *afin de* disposer de la meilleure solution selon moi.



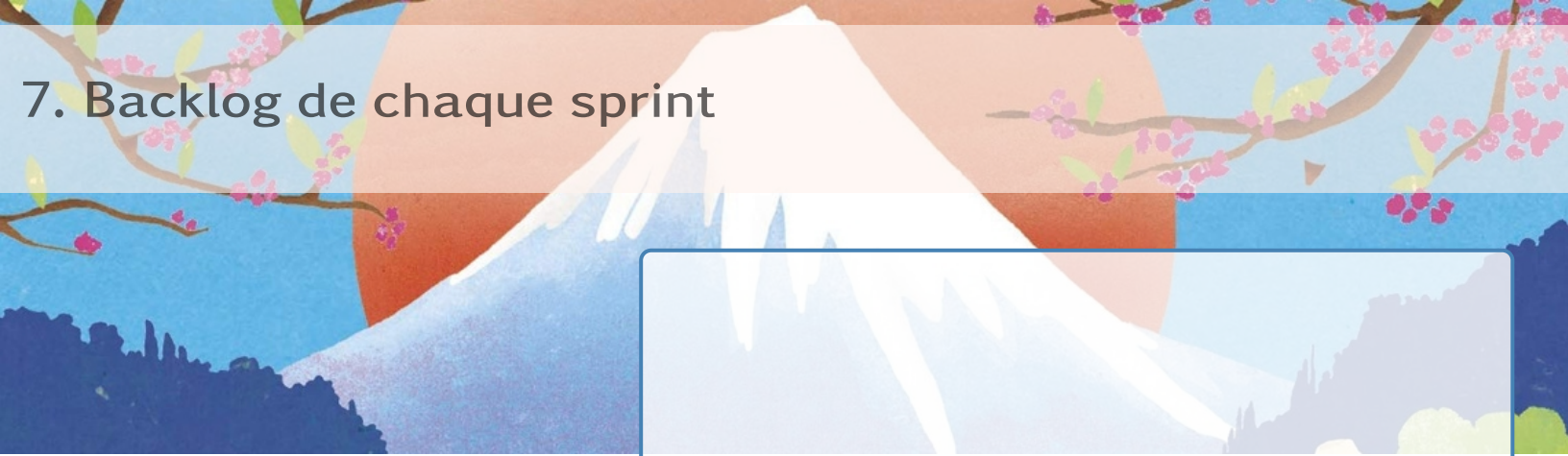
## 5. Backlog global





## 6. Roadmap globale





## 7. Backlog de chaque sprint

A large empty rectangular box with a blue border, intended for the sprint backlog.





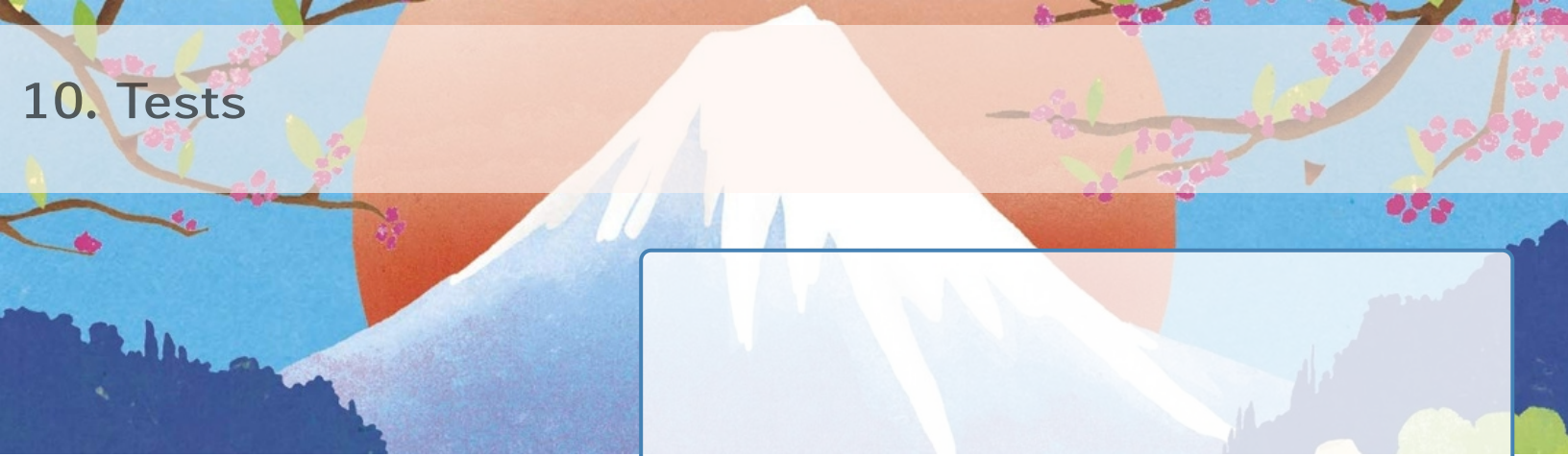
## 8. Explication de l'algorithme





# 9. Choix de programmation





# 10. Tests



## 11. Gestion des versions concurrentes





