



X



# Gestion des risques et de rendus

**Auteurs :**

Mehdi El Idrissi El Fatmi

Khadija El Adnani

Alaa Jennine

Badr El Khoundafi

Yassine El Kouri

22/01/2025

# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Identification et Classification des Risques</b>	<b>2</b>
2.1	Risques Techniques . . . . .	2
2.2	Risques Humains . . . . .	2
2.3	Risques Organisationnels . . . . .	3
<b>3</b>	<b>Plan d’Atténuation des Risques</b>	<b>3</b>
3.1	Stratégies pour Réduire les Risques Techniques . . . . .	3
3.2	Stratégies pour Améliorer la Collaboration et Réduire les Risques Humains	3
3.3	Stratégies pour Optimiser l’Organisation et Minimiser les Retards . . . . .	4
<b>4</b>	<b>Gestion des Rendus</b>	<b>4</b>
4.1	Calendrier des Livrables . . . . .	4
4.2	Utilisation d’Outils de Gestion . . . . .	4
4.3	Vérifications Avant Soumission . . . . .	5
4.4	Gestion des Retards et Contingences . . . . .	5
<b>5</b>	<b>Conclusion</b>	<b>5</b>

# 1 Introduction

Dans tout projet de développement logiciel, la gestion des risques est une activité clé qui permet d'anticiper les obstacles potentiels et de définir des plans d'actions pour en minimiser l'impact. Ce document présente les principaux risques identifiés dans notre projet de Génie Logiciel (GL), leur classification par ordre d'importance ainsi que les stratégies que nous avons adoptées pour les gérer.

## 2 Identification et Classification des Risques

Les risques rencontrés dans notre projet peuvent être regroupés en trois grandes catégories : techniques, humains et organisationnels. Chaque risque est analysé selon son impact potentiel et sa probabilité d'occurrence.

### 2.1 Risques Techniques

Ces risques concernent les aspects techniques liés au développement et à l'intégration des différentes parties du projet.

- **Bugs critiques dans le code** : La présence de bugs dans des parties critiques du projet pourrait retarder les phases de validation. (Impact : *élevé*)
- **Performance insuffisante** : Une mauvaise optimisation du code peut entraîner une performance inacceptable de l'application. (Impact : *moyen*)
- **Complexité de l'extension ARM** : L'implémentation de l'extension ARM peut poser des difficultés supplémentaires en raison des spécificités de cette architecture. (Impact : *élevé*)
- **Non-conformité avec les spécifications** : Le compilateur doit respecter strictement les spécifications de la machine cible. Toute non-conformité pourrait entraîner une note basse lors de l'évaluation finale. (Impact : *élevé*)

### 2.2 Risques Humains

Les risques humains découlent de la gestion de l'équipe et des interactions entre ses membres.

- **Mauvaise communication au sein de l'équipe** : Une communication inefficace peut engendrer des malentendus et ralentir le projet. L'utilisation d'outils collaboratifs comme **Jira** et Discord permettra de suivre les tâches et de faciliter les échanges. (Impact : *élevé*)
- **Répartition inégale des tâches** : Une répartition non équilibrée peut démotiver certains membres ou surcharger d'autres. (Impact : *moyen*)
- **Disponibilité limitée de certains membres** : Des contraintes personnelles peuvent limiter l'engagement de certains membres à certaines périodes critiques. (Impact : *moyen*)

## 2.3 Risques Organisationnels

Ces risques concernent la planification et la coordination globale du projet.

- **Non-respect des échéances** : Tout retard dans les jalons intermédiaires pourrait compromettre la livraison finale. (Impact : *élevé*)
- **Manque de suivi des tâches** : Une absence de suivi rigoureux peut entraîner des oublis et des retards. (Impact : *moyen*)
- **Documentation insuffisante** : Une documentation incomplète rendra difficile la maintenance et l'évolution du projet. (Impact : *moyen*)

## 3 Plan d'Atténuation des Risques

Afin de garantir la réussite de notre projet, nous avons mis en place des stratégies concrètes pour chaque type de risque identifié. Ces stratégies visent à réduire à la fois la probabilité d'occurrence et l'impact de chaque risque.

### 3.1 Stratégies pour Réduire les Risques Techniques

- **Automatisation des tests** : Développement d'un grand nombre de tests unitaires, couvrant tous les cas d'usage et les scénarios possibles, y compris les cas limites et les erreurs attendues. Des scripts d'exécution automatisés seront conçus pour tester systématiquement chaque partie du projet après toute modification. Cela garantit une détection rapide des régressions et une validation rigoureuse des nouvelles fonctionnalités.
- **Revue de code en binôme (Pair programming)** : Chaque contribution doit être validée par un autre membre de l'équipe afin d'identifier les erreurs potentielles en amont.
- **Documentation technique précise** : Rédaction systématique de guides et d'exemples d'utilisation pour chaque composant, facilitant ainsi la maintenance et l'intégration.

### 3.2 Stratégies pour Améliorer la Collaboration et Réduire les Risques Humains

- **Daily Stand-ups** : Réunions courtes chaque matin (15 minutes maximum) pour synchroniser l'équipe, partager les progrès, et identifier les blocages.
- **Répartition dynamique des tâches** : Utilisation d'un tableau Trello avec des colonnes claires (*To Do*, *In Progress*, *Done*) pour visualiser la charge de travail et assurer une répartition équitable.
- **Système d'alerte précoce** : Chaque membre signale immédiatement les obstacles qu'il rencontre pour éviter un effet domino sur les autres tâches.

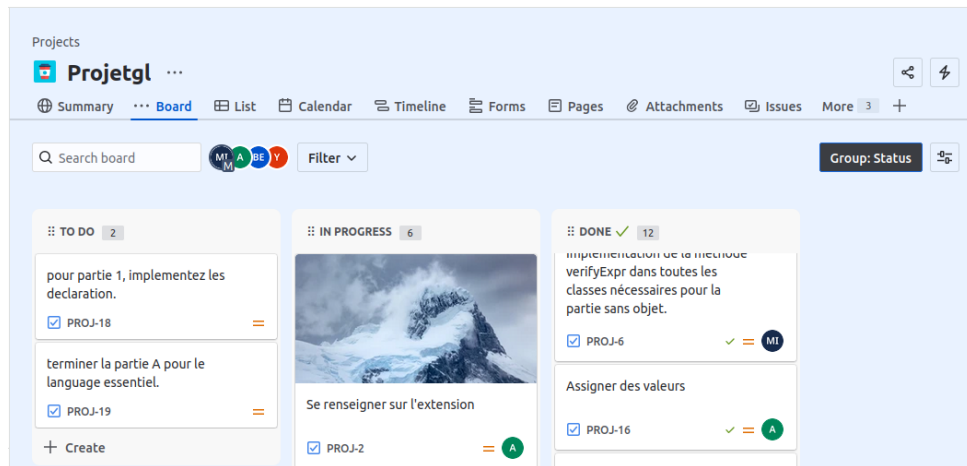


Figure 1: Exemple de tableau de bord Jira utilisé pour suivre les tâches et identifier les blocages.

### 3.3 Stratégies pour Optimiser l'Organisation et Minimiser les Retards

- **Suivi par jalons** : Fractionnement du projet en jalons hebdomadaires clairs avec des livrables définis. Chaque jalon est revu collectivement.
- **Planning interactif avec Trello** : Mise à jour régulière des échéances et réajustements en cas de retard.
- **Plan de secours pour chaque jalon** : Préparation de solutions alternatives pour les tâches critiques afin de garantir l'achèvement des livrables.

## 4 Gestion des Rendus

Une gestion efficace des rendus est cruciale pour assurer le bon déroulement du projet et le respect des délais fixés. Pour cela, nous avons mis en place une méthodologie rigoureuse qui repose sur les éléments suivants :

### 4.1 Calendrier des Livrables

Un calendrier précis des rendus a été établi dès le début du projet. Chaque livrable est associé à :

- Une date limite clairement définie.
- Un responsable principal chargé de coordonner la finalisation du livrable.
- Une période tampon pour permettre des ajustements avant la soumission officielle.

### 4.2 Utilisation d'Outils de Gestion

Pour garantir un suivi optimal des rendus, nous utilisons les outils suivants :

- **GitHub** : Chaque livrable doit être intégré au dépôt GitHub principal avant la date limite. Les branches dédiées sont utilisées pour le développement et les tests.
- **Jira** : Les tâches liées à chaque livrable sont assignées et suivies sur le tableau de bord, assurant ainsi une transparence totale.
- **Notifications automatisées** : Des rappels automatiques sont envoyés via des intégrations comme Discord pour alerter les membres des échéances imminentes.

### 4.3 Vérifications Avant Soumission

Avant chaque rendu, des vérifications finales sont effectuées :

- **Validation technique** : Tests automatisés pour vérifier la conformité du code et des fonctionnalités.
- **Relecture et corrections** : Chaque document est relu par au moins deux membres de l'équipe pour garantir la qualité et l'absence d'erreurs.
- **Check-list de soumission** : Une liste de contrôle est utilisée pour s'assurer que tous les éléments requis sont inclus.

### 4.4 Gestion des Retards et Contingences

En cas de retard potentiel, des plans de contingence sont activés :

- **Réattribution des tâches** : Si un membre rencontre des difficultés, les tâches critiques peuvent être redistribuées.
- **Extensions d'échéances** : Des extensions limitées peuvent être envisagées si elles n'affectent pas le calendrier global.
- **Communication proactive** : Les membres de l'équipe signalent rapidement tout problème pour éviter un effet domino.

## 5 Conclusion

La gestion proactive des risques est essentielle pour garantir la réussite de notre projet. En identifiant les risques majeurs et en mettant en place des stratégies adaptées, nous maximisons nos chances d'atteindre nos objectifs dans les délais impartis tout en assurant une qualité optimale.