

# Projet avec XCL

---

Bonjour,

V 1.2, jan 2025

Si vous lisez ce document c'est que vous démarrez un projet et que c'est moi - Xavier Carrel - qui porte la casquette de chef de projet.

Pour que notre collaboration soit le plus efficace possible, je tiens à vous communiquer à travers ce document certains points auxquels je tiens dans la manière de mener un projet.

## Méthodologie

Je me reconnais pleinement dans les valeurs énoncées par le manifeste agile, que je me permets donc de rappeler ici :

- Des personnes et leurs interactions plutôt que des outils et des processus
- Des logiciels qui fonctionnent plutôt qu'une documentation exhaustive
- La collaboration avec le client plutôt que la négociation contractuelle
- L'adaptation au changement plutôt que le suivi d'un plan

Par conséquent, je préconise l'utilisation de méthodes agiles (plus précisément Scrum) pour la gestion de projet.

## Pratiques

### Analyse fonctionnelle

Je demande à ce que l'analyse fonctionnelle se présente sous la forme d'une liste de User Stories (US), chacune étant détaillée au moyen d'un nombre significatif de tests d'acceptance.

La formulation des tests d'acceptance:

- Doit permettre leur exécution sans ambiguïté (SMAAAR)
- Est grandement simplifiée lorsqu'elle fait référence à des éléments graphiques (maquette, schéma, diagramme, ...).

Le sens et la clarté de ces éléments priment sur leur qualité graphique. Je préfère un dessin fait à la main et digitalisé à une maquette réalisée avec Figma si cette dernière prend plus de temps à être réalisée que le premier.

### Rapport

Conformément aux valeurs agiles, ce que vous réalisez est plus important que la documentation à mes yeux.

Il n'est cependant pas possible de s'affranchir de toute documentation.

Je souhaite que la documentation suive une organisation légèrement différente - mais pas en contradiction - du modèle ETML ou TPIVD. Je vous fournis un modèle Word ([.dotx](#)) qui contient ces différences. (Une

différence notable que je tiens à mentionner ici est la rédaction d'une section sur l'usage - ou pas - de l'IA dans votre projet.)

Il n'est pas obligatoire de maintenir la structure de votre rapport exactement comme ce canevas. Mais il est indispensable que les éléments identifiés par celui-ci soient traités dans votre rapport.

## Journal de travail

La tenue d'un journal de travail est indispensable.

Vous êtes libres de choisir l'outil et la forme que vous souhaitez. Mais votre journal de travail doit absolument satisfaire les critères suivants :

1. La structure et la présentation sont claires et soignées
2. Le résultat du travail (artefact) y est référencé précisément. Recommandation: mettre un lien sur le commit correspondant
3. La durée de chaque tâche est mentionnée. Si une tâche n'est pas achevée en fin de journée, son énoncé est suivi de **[WIP]**
4. Tout événement impactant le temps de travail prévu (activités ETML planifiées, absences, imprévus) y est reporté
5. Les heures supplémentaires y sont mentionnées
6. Les succès et les échecs sont mentionnés
7. Toute tâche de 45 minutes ou plus est commentée
8. Le temps de travail total par jour et sur l'ensemble du projet est calculé

## Daily Scrum

Votre projet est individuel ? Vous trouvez peut-être que le Daily Scrum n'a donc pas de sens (on ne fait pas une mêlée avec une seule personne!).

Je vous recommande pourtant vivement de conserver cette pratique malgré tout. Consacrez les premières minutes de chaque journée de travail à prendre un peu de recul et faire le point.

- Qu'est-ce que j'ai fait hier ? (l'occasion de vérifier que votre journal de travail est à jour)
- Qu'est-ce que je pense faire aujourd'hui ?
- Est-ce que j'ai un problème qui me freine ou me bloque ? (l'occasion de planifier un moment d'aide avec un tiers)

## Outils

Un outil est quelque chose qui nous aide à réaliser un travail plus efficacement. On creuse par exemple plus facilement un trou dans le sol avec une pelle qu'avec ses mains.

Je liste donc ici des outils informatiques qui - j'en suis convaincu - vous aident à réaliser votre projet.

Conformément aux valeurs agiles, les individus et leurs interactions priment sur l'outil. Je suis donc prêt à discuter et à m'adapter si vous souhaitez faire usage d'autres outils que ceux ci-dessous. Attention: il faudra savoir me convaincre que vous serez plus performants avec l'outil que vous proposez.

## Contrôle de version: GitHub

- Créez un repo privé dans votre compte et invitez-moi en lecture seule.

- Suivez la convention de nommage des commits [conventional commits](#).
- Appliquez le modèle [gitflow](#) pour la gestion des branches, avec les branches [main](#), [develop](#) et les branches de features.
- Lors des sprint reviews, les fonctionnalités sont évaluées dans leur branche de feature. Si la story est validée, la branche de feature est fermée.
- Tous les fichiers utiles au travail et autres que du code (documentation technique interne ou externe, rapports, exemples, snippets, etc...) se trouvent dans l'arborescence sous [doc](#), dont la structure est livrée à votre bon sens.
- Bien que cela soit une pratique controversée, je demande à ce que les fichiers concernant la documentation soient également intégrés dans le repository. Ils doivent se trouver également dans le dossier [doc](#).

## Gestion de projet: GitHub Project

Créez un projet, lié à votre repository, en vous basant sur le modèle [Kanban](#) proposé.

Ne conservez que 5 colonnes: [Sandbox](#), [Backlog](#), [In Progress](#), [In Review](#), [Done](#)

Invitez-moi dans le projet.

Rédigez vos US sous forme d'issues dans la SandBox et sollicitez-moi. Nous discutons, corrigeons, détaillons, précisons et finalement validons vos stories en les passant dans le backlog.

Vous estimez et planifiez vos stories vous-même.

## Journal de travail

Bien que vous soyez libre de gérer votre journal de travail manuellement avec l'outil qui vous convient, je recommande l'utilisation de [GitJournal](#), qui tire profit des efforts que vous avez fait dans la gestion de vos commits pour générer automatiquement un journal de travail.

## Aide à la réalisation : une IA ?

Le recours à une Intelligence Artificielle pour vous aider dans la réalisation de votre cahier des charges est autorisé - voire encouragé - pour autant que l'IA reste un **outil** qui vous **aide** et non un système qui fait le travail **à votre place**.

En d'autres termes, vous devez être capable d'expliquer vous-même tout ce qui est produit dans le cadre de votre projet.

## Livraison

Rappel: Le cahier des charges du projet définit les livrables.

Chaque livraison est effectuée au moyen d'une [release github](#)

Les livrables de type documentaire (rapport, journal de travail, ...) doivent être ajoutés en PDF aux assets de la release.

Dans le cas où un livrable ne peut pas être directement inclus dans le repository (taille trop grande par exemple), joignez à la release un fichier [pdf](#) qui contient un lien vers le livrable. Assurez-vous que la cible du lien soit accessible par votre destinataire.

Toute livraison doit m'être notifiée explicitement par un email contenant un lien sur la release.

## Evaluation

Dans le cadre des projets à l'approche - et durant - le TPI, je me sers de la [grille automatisée](#) de TPI Eval. Vous êtes invité-e à vous en servir également pour de l'auto-évaluation.

## Conclusion

Il ne serait pas cohérent de ma part de vous obliger à mettre en œuvre tout ce qui précède à la lettre, parce que ce serait contraire aux valeurs du manifeste agile citées au début de ce document.

Je reste par conséquent ouvert à discuter et mettre en œuvre des pratiques différentes que vous souhaiteriez me proposer.