

# Gestion des risques et rendus

## Projet Génie Logiciel 2020

### Suivi 2

Groupe n° 13

MORIN Lucas

NAVARRO Jérémy

ODEH Majd

OUDOUMANESSAH Geoffroy

POUGET Sylvain

# Conscience des risques

Lors d'un projet aussi conséquent que celui de génie logiciel, il est nécessaire d'instaurer des règles ainsi que des méthodologies précises. Cela vient du fait que le compilateur à implémenter est un programme impossible à tester de manière exhaustive.

Sur un programme informatique les erreurs sont variées, imprévisibles, contre intuitives et pas forcément évidentes à repérer. Ainsi la manière d'éviter les erreurs reste encore de ne pas en faire. Cela peut paraître triviale comme solution mais cela a le mérite d'être efficace.

## Les méthodes mise en place

Lors du développement nous avons choisis de ne pas faire de code au kilomètre. Chaque étapes, aussi simple qu'elle peut paraître doit être réfléchi, sans précipitation.

Une part importante de notre travail s'est fondé sur de la discussion autour de schémas, de diagrammes, ... de supports visuels. Cela permet d'avoir une vue globale d'une tâche à effectuer et donc de ne pas se risquer à créer des problèmes sur des couches plus éloignées sans le savoir.

Nous avons réalisé beaucoup de vérification de code. Pour cela nous avons travaillé avec deux méthodes.

La première est la review de code. Autrement dit la relecture de notre travail par une autre personne du groupe. En plus de permettre l'identification de problèmes au plus tôt, cela a de nombreux avantages. L'équipe entière par ce système peut avoir une vue d'ensemble du code implémenté. De plus, cela permet de s'améliorer en découvrant de nouvelles visions et d'approches sur des problèmes.

La deuxième solution de coder à deux. Cela permet d'identifier rapidement des erreurs mais surtout d'accélérer grandement la phase normale de réflexion.

Ces différentes parties représentent la correction d'erreurs humaines avant même sa propagation. Cependant comme il n'est pas possible d'être infallible, il a fallu utiliser d'autres outils prévenir des erreurs.

## Les outils à notre disposition

Généralement, même si on ne peut pas identifier clairement du premier coup une erreur, on peut isoler un sous ensemble dans lequel on sait qu'elle va apparaître. Le fait de pouvoir pointer un échantillon spécifique posant problème permet de mettre en place des processus autonomes de vérification et d'identification, en vu d'une correction.

Cela constitue nos tests de codes automatiques. Ces tests sont généralement très courts mais très spécifiques. Chacun se focalise uniquement sur une fonctionnalité très précise en vue d'identifier une erreur au plus proche.

Un test efficace est un test peu commun, presque inimaginable. Il faut réussir à trouver un cas si particulier qu'il rend possible l'émergence d'un bogue.

C'est pour cela que ces tests sont créés de manière séquentielle. Chacun, dès lors qu'il perçoit une faille possible va mettre en place un petit test.

Cet ensemble de tests est disponible pour tout le monde et surtout est utilisé par tout le monde. De ce fait, chacun teste le travail des autres.

Ensuite un outil que nous utilisons au quotidien est Trello. Il permet de s'organiser très facilement et de manière très lisible en permettant une gestion des tâches sous la forme d'action à réaliser.

Il y a ainsi une section Bogue qui nous permet d'épingler une erreur qui a été détectée. Ensuite une personne peut s'approprier la tâche pour la résoudre.

Cette communication permet de ne pas travailler simultanément sur une tâche identique sans le savoir. Nous avons mis en place une section des opérations en cours qui permet à tout moment de connaître le travail de chacun. Cela évite la redondance inutile et la multiplication possible des erreurs.

## Les méthodologies pour réduire les erreurs

Un élément critique de notre projet est le fait que nous utilisons un gestionnaire de version partagée. Tout notre code est hébergé et stocké sur gitlab. Cela permet de travailler sur une même version qui est la plus aboutie. Cependant cela pose un problème énorme, à savoir qu'un fichier déposé sur ce répertoire est considéré comme correct et opérationnel.

Ainsi il est nécessaire de contrôler un maximum les fichiers déposés dans ce répertoire pour éviter une contamination.

Pour valider le dépôt d'un nouveau fichier, voici les étapes que nous cherchons à valider systématiquement.

Tout d'abord le projet global doit se comporter de manière correct. C'est à dire qu'il doit compiler et s'exécuter de manière correcte dans la version actuelle. Ensuite, le programme doit réussir tous les tests qui sont validés dans la version courante de travail.

Ensuite en ce qui concerne les fichiers déposés, il faut être sûr de déposer uniquement les fichiers modifiés. Il serait dangereux de modifier un fichier sain inutilement.

Enfin, pour s'assurer une future correction du code, il est nécessaire que celui-ci soit compréhensible facilement et qu'il soit convenablement commenté.

Cela nous permet d'avoir une architecture toujours opérationnelle et valide. De plus, cela permet livrer à tout moment un programme qui sera certes limité mais fonctionnel.

# Préparation des rendus

Les rendus est une étape importante dans le processus de ce projet. En effet, il permet de s'assurer que l'on se dirige dans la bonne direction et impose des points de contrôles.

C'est pour cela que l'on se prépare au maximum en avance. On se force à rester honnête dans notre travail et notre avancement. Pour cela on conserve tout état de notre avancement et progression. Cela permet de nous rendre compte des points délicats qui auraient pu être négligés et aussi de nous améliorer.

Pour cela nous avons opté pour une réunion quotidienne le matin, pour fixer nos objectifs de la journée. Tous les matins nous pouvons donc mettre à jour l'emploi de temps et affiner au mieux le calendrier prévisionnel.

De plus, ces réunions permettent d'entretenir des discussions permanentes sur l'avancement de chacun et du projet en général.

Pour renforcer ce travail d'équipe, nous sommes travaillons constamment tous les cinq en présentielle. En plus de renforcer le travail d'équipe, cela permet de facilement déboguer des éléments, d'avoir une vue d'ensemble sur le travail de chacun et de comprendre l'implication de chaque éléments du projet.

## Release management process

- ☐ Vérifier que mvn compile
- ☐ Vérifier les tests de non régression
- ☐ Vérification que les fichiers sont bien les bons, dans la dernière version

Cela permet de vérifier la qualité du code à rendre et éviter des erreur inutiles.

- ☐ Mise à jour du Trello

Pour être certain de ne pas dupliquer des actions.

- ☐ Review de la documentation

Permet d'assurer une qualité du document avec un regard extérieur.

- ☐ Mise en conformité du planning des tâches

Cela assure le bon avancement du projet et le respect des dates limites.

- ☐ Evaluation des performances des nouveaux composants

Pour éviter d'utiliser des implémentations inutiles, voir même plus lente.

- ☐ Préparation des comptes rendus, gestion du temps de parole et des points à souligner