

# Gestions des risques et gestions des rendus

Projet GL groupe 13

12 janvier 2022

## Gestion des risques

### Méthodes de vérification manuel du code

Chaque partie de code, une fois développée, est soumise à une relecture par les pairs afin d'identifier des erreurs évidentes de syntax ou un mauvais formatage du code. Tout code est donc relu afin de pouvoir mieux identifier la direction prise et les choix des différents développeurs. Une fois le code relu, il est mis en phase de tests.

### Méthodes de validation des tests

Plusieurs moyens sont mis en oeuvre afin de ne pas introduire de bugs massifs dans notre code. La première étant de prendre et de développer les fonctionnalités une par une, puis une fois validée d'en développer une autre. Par exemple, si nous devons développer les opérations arithmétiques, chaque étape du développement sera testée, tout d'abord par un test de lexing, puis par un test de parsing, puis par un test de contexte et enfin un test de compilation avec bien sûr des tests sur les cas limites. En effet, par exemple pour les Opérations Arithmétiques les cas limites testés sont ceux d'opération entre des types différents, avec des nombres trop grand (Overflow) ou un résultat trop grand. Ainsi, si le test de la partie correspondante est bon, la fonctionnalité est bonne et on peut passer à la partie suivante, sinon, un retour sur le bug est créé et une correction est proposée puis retestée. Le Schéma ci-dessous explique plus en détail la méthode de validation du travail:

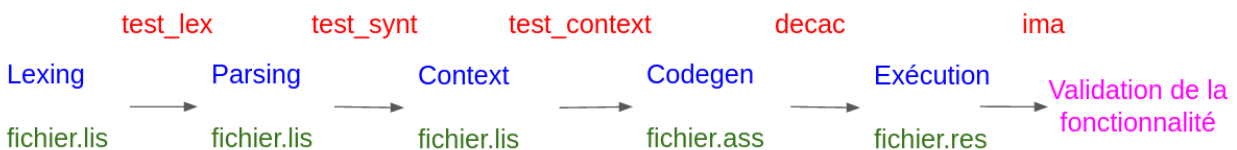


Figure 1: Etapes de validations

### Couverture des tests

Pour savoir si notre code est bien testé sous toutes ses coutures, en plus d'une large gamme de tests, nous utilisons l'outil de couverture de tests Jacoco qui nous permet d'identifier les parties du code source qui n'ont pas encore été contrôlées. Ainsi, si l'on remarque que pour un programme, la couverture de tests est mauvaise, on en déduit qu'il faut créer des nouveaux tests afin de bien tester cette partie.

## Gestions des rendus

Avant chaque rendu, il est important de tester et de relire la branche principal du git. C'est pour cela que nous travaillons de manière quotidienne sur la branche git develop, sur laquelle nous effectuons nos tests, nos relectures et nos tentatives de résolutions. Il n'est push sur la branche master de notre git, celle qui fera fois

lors des rendus, que les fonctionnalités qui ont passées toutes les étapes de validation précédemment décrite. Ainsi, même si le jour du rendu une fonctionnalité n'est pas totalement terminée, elle ne sera pas rendue pour éviter que les potentiels bugs qu'elle induirait détruise toute la structure du compilateur qui lui fonctionne déjà. Cela est fait pour limiter au maximum les bugs majeurs qui seraient totalement contreproductif et nuirait au bon fonctionnement de notre compilateur.