

# Notes de cours de Fiabilité Logicielle

Yann Miguel

21 janvier 2021

## Table des matières

1	Introduction	2
2	Cours 1	3
3	Cours 2	4
3.1	Spécification . . . . .	4
3.1.1	En langue naturelle . . . . .	4
3.1.2	En langage formel . . . . .	4
3.1.3	Par formule logique . . . . .	4
4	Cours 3	6
5	Informations importantes	7

# 1 Introduction

Il y a des concepts mis en cause par les bugs:

- Erreur d'exécution
- Faille de sécurité
- Erreur de spécification
- Manque de maintenance
- Manque de performance

En informatique, on construit des logiciels critiques, qui sont des objets complexes. Cela demande donc une forme de professionnalisme, tel que un code de bonne qualité, fiable, et sécurisé, et ceux pour un coût limité.

Il faut donc que le système construit puisse résister aux attaques malveillantes, tels que les virus, vers, et autres malwares.

## 2 Cours 1

## 3 Cours 2

Comment définit-on correct? L'absence d'erreur à l'exécution?  
Le code a le comportement voulu?  
On définit la correction du code et de la fonction par rapport à la spécification donnée.

### 3.1 Spécification

Il y a trois types de spécification:

1. en langue naturelle
2. en langage formel
3. par formule logique

#### 3.1.1 En langue naturelle

Une spécification en langue naturelle est écrite dans un langage utilisé tout les jours, comme le français ou l'anglais.  
Tout doit être écrit comme si la personne n'avait jamais rien fait d'autre que coder de sa vie, mais, attention, il ne faut pas écrire comment on déduit le résultat de l'entrée. La personne qui code doit avoir un peu de job, quand même.

#### 3.1.2 En langage formel

On décrit la spécification avec du vocabulaire formel, et en utilisant des symboles logiques et ou mathématiques, voir en utilisant des automates pour représenter les interactions entre les différents composants du programme final voulu.

#### 3.1.3 Par formule logique

On peut spécifier via plusieurs logiques:

- La logique du premier ordre: cela permet d'écrire de manière propre des spécifications. Elle demande trois notions: la **précondition**, ce qui est requis avant l'exécution d'une fonction, la **postcondition**, ce qu'on a après l'exécution de la fonction, et le **contrat**, qui détermine si les données en précondition sont bonnes, et que le résultat en postcondition est bon si le calcul termine.

- La logique temporelle: souvent interprétée dans les systèmes ou il y a du temps discret, c'est à dire, des instants les uns après les autres.
- La logique d'ordre supérieur: on a pas eu le temps d'en parler.

## 4 Cours 3

Il y a en moyenne entre 30 et 85 erreurs pour 1000 lignes de code. On utilise des tests pour les faire apparaître.

Il est important de se souvenir que les tests ne font que montrer qu'il y a des erreurs. Ils ne disent **jamais** quand il n'y en a plus.

L'analyse de qualité est basée sur des suites métriques mesurent la qualité d'une suite de test, et non pas d'un test isolé.

## 5 Informations importantes