

28/04/2021

# Génie logiciel

- Plan :
- I. Introduction au GL.
  - II. Qualité d'un logiciel.
  - III. Cycle de vie d'un logiciel.
  - IV) Modèles de cycle de vie.
    - 5) Langage de modélisation UML.
    - 6) Diagramme UML.
      - 6.1) Diagramme des cas d'utilisation.
      - 6.2) " d'activité.
      - 6.3) " de classes.
      - 6.4) " d'objet.
      - 6.5) " de séquence.
      - 6.6) " d'états.

## I) Introduction au G.L :

I.1 Définition du G.L.

I.2 Définition d'un logiciel.

## II) Maintenance de logiciel.

## III) But du génie logiciel.

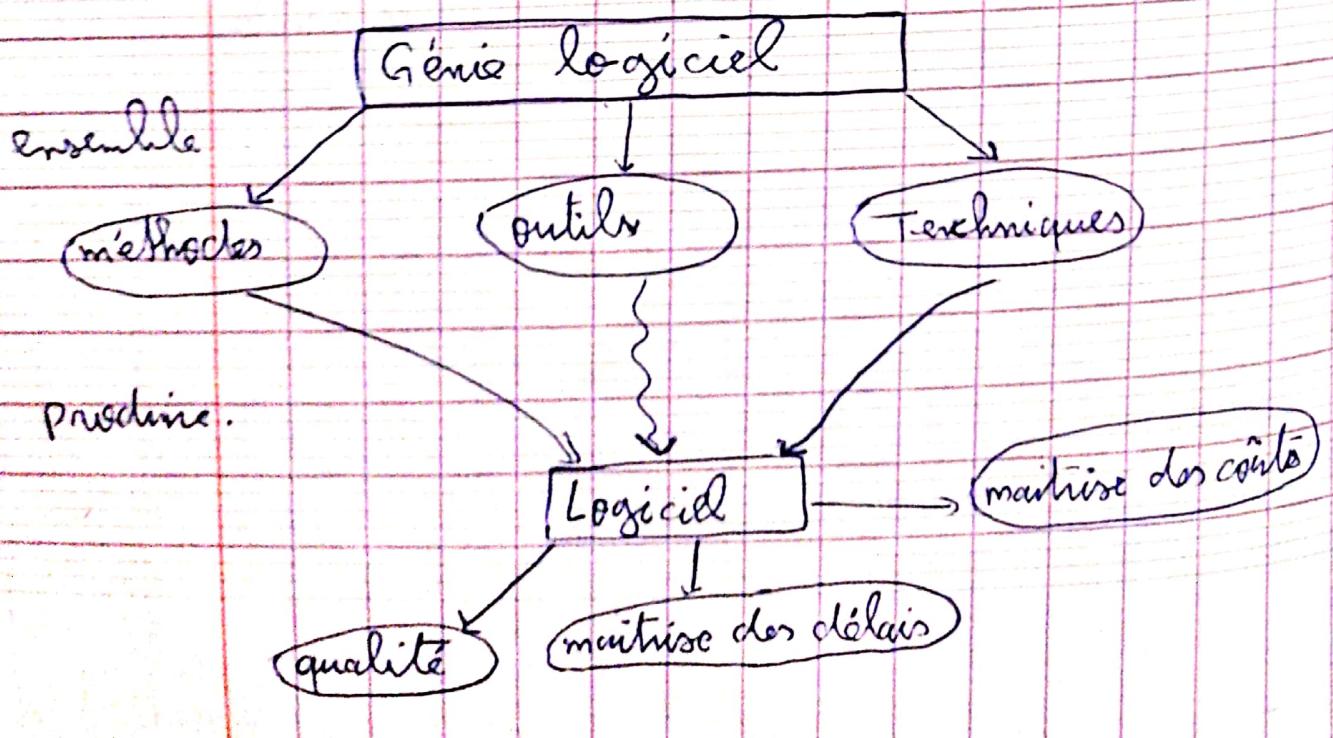
## IV) Problèmes de G.L.

## V) Enjeux du G.L.

## VI) Références.

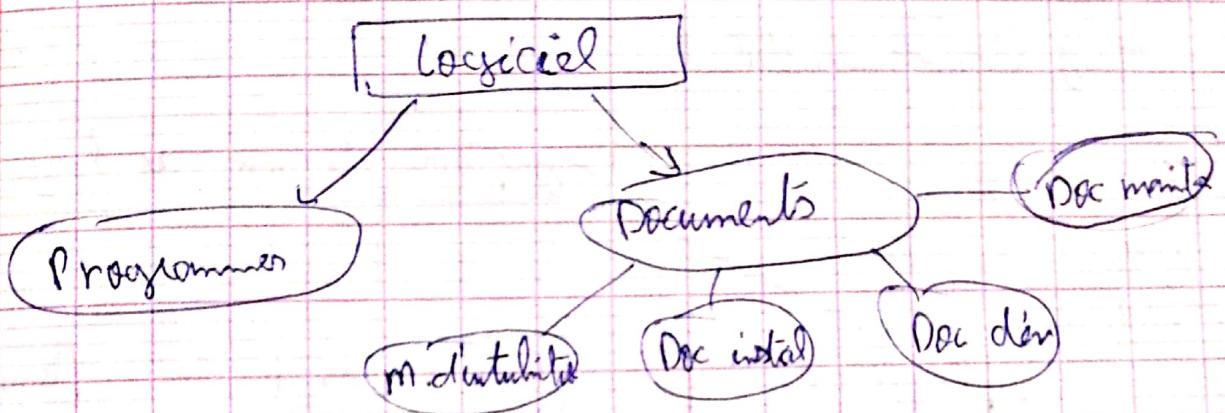
### I.1) Définition du Génie logiciel:

- C'est l'ensemble des méthodes, outils, techniques dans le but de créer un logiciel de qualité et qui maîtrise les délais et les coûts.

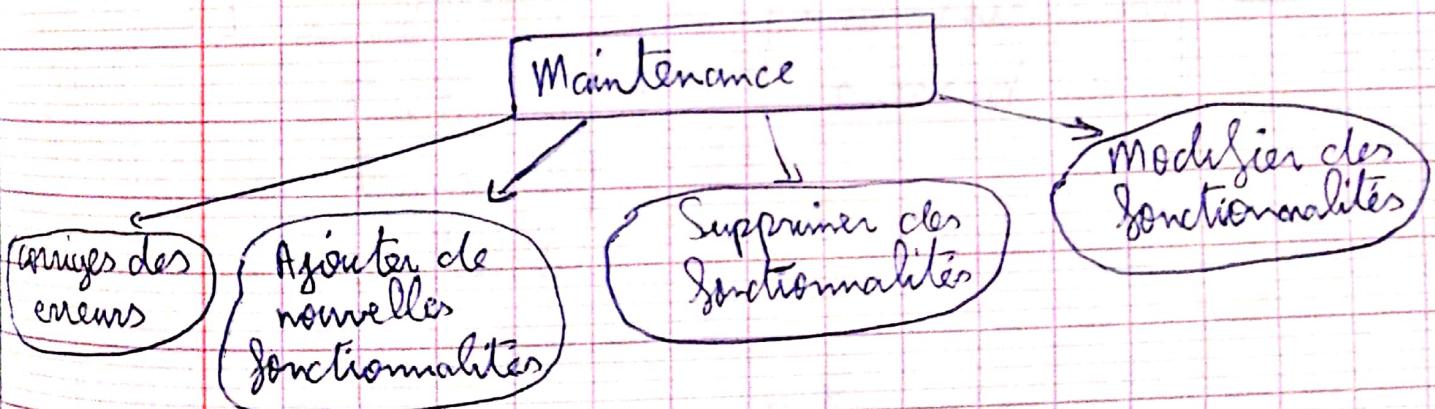


## I-2) Définition d'un logiciel:

C'est l'ensemble des programmes et les documents (Manuel d'utilisation, Doc d'installation, Doc de développement, Doc de maintenance).



## II) Maintenance d'un logiciel



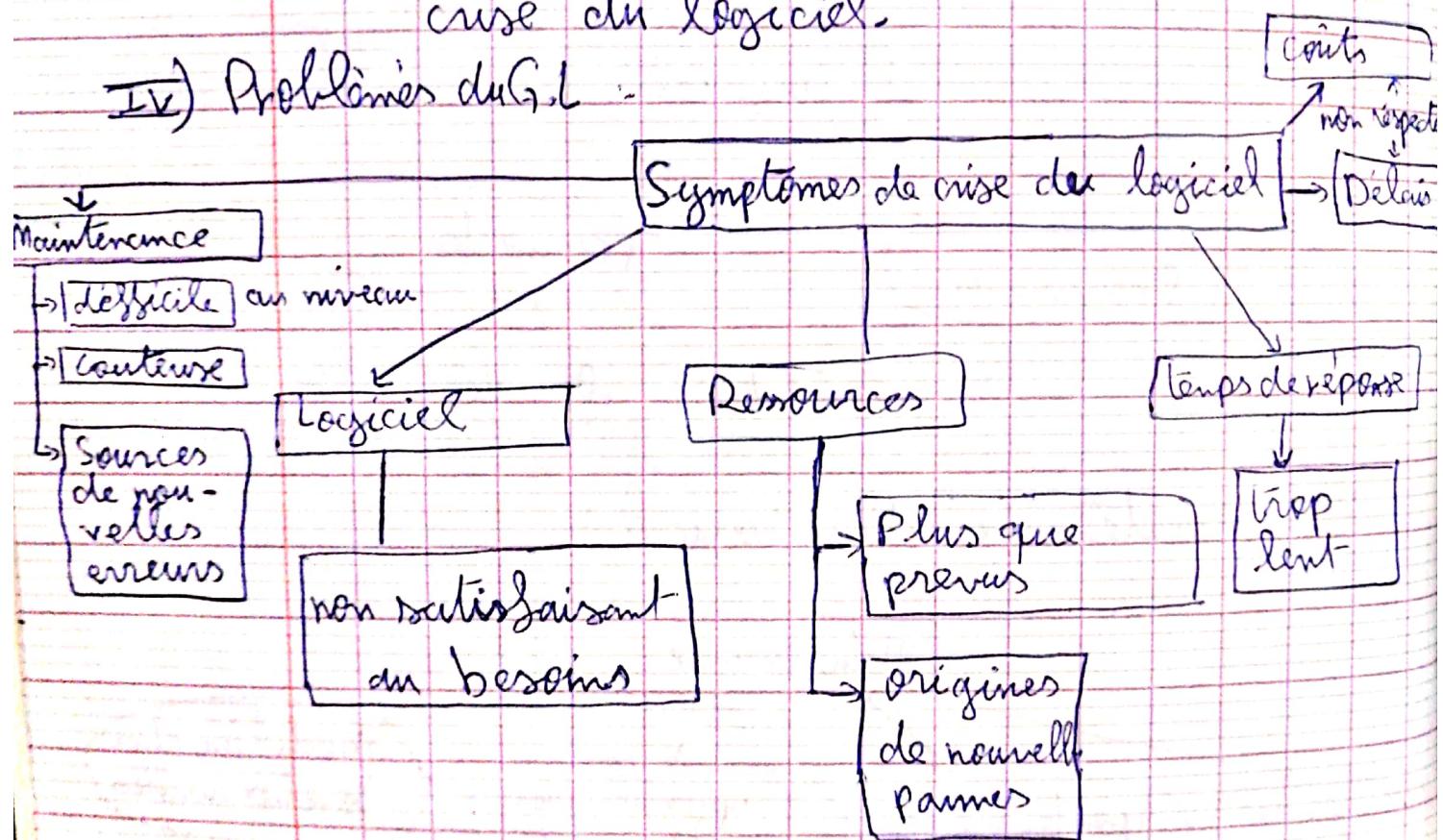
### III) But du Génie Logiciel :

• Le mot "Génie Logiciel" :

1. est apparu à les années 60.

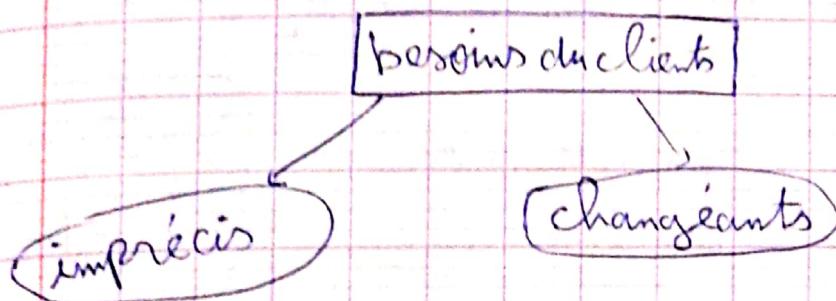
2. pour répondre (apporter des solutions) à la crise du logiciel.

### IV) Problèmes du G.L. :



## II) Enjeux du G.L :

Problèmes :



Solution 1 :

- Le client doit :

1. Préciser ces besoins.
2. Fixer ces besoins. (Fixer).
3. Au début de projet

Solution 2 :

- Développer :

1. Graduellement.
2. Tolérer les imprécisions.
3. Tolérer les changements.

Rq : Ces deux solutions sont deux pratiques du G.L qui s'opposent sur la manière de traiter les 2 problèmes (imprécisions et changement des besoins).

## VII) Enjeux du G.T.

Bien développer le logiciel

Comment produire  
un logiciel  
de bonne qualité

Qu'attend-on  
du logiciel

Quelles sont  
les intérêts de  
qualités ...

## 6.1. Cons 2 : Qualité d'un Logiciel :

05/05/2021

Plan :

I) Introduction.

II) Qualité d'un logiciel du point-vue utilisation

III) " " " " " Developpem.

### I) Introduction :

- Un logiciel devra répondre à un certain nombre de critères de qualité.
- Ces critères requises de qualité font partie des exigences non fonctionnelles du logiciel.
- Pour rappel : les besoins fonctionnelles (exigences fonctionnelles) et les besoins non fonctionnelles font partie du Cahier de charge.
- Les exigences non fonctionnelles sont subdivisées en :

① : Exigences techniques (on peut les programmer)

② : " non techniques (on ne peut pas les programmer mais il faut veiller à les respecter dans la conception du logiciel)

## Exigences non fonctionnelles :

	Point de vue utilisateurs	Point de vue développement
Exigences non fonctionnelles Techniques	<ul style="list-style-type: none"><li>- Interprétabilité, Validité.</li><li>- Performance.</li><li>- Fiabilité.</li><li>- Disponibilité.</li><li>- Robustesse.</li><li>- Sécurité</li><li>- Utilisabilité.</li><li>- Convivialité</li></ul>	<ul style="list-style-type: none"><li>- Implementation.</li><li>- Instabilité.</li><li>- Maintenabilité.</li><li>- Opérabilité.</li><li>- Portabilités</li><li>- Réutilisabilité.</li><li>- Scalabilité.</li><li>- Testabilité.</li><li>- Tracabilité.</li></ul>
Exigences non fonctionnelles non techniques	<ul style="list-style-type: none"><li>- Intégrité de l'entreprise</li><li>- Culturel.</li><li>- Politique.</li></ul>	<ul style="list-style-type: none"><li>- Legal</li><li>- Conformité aux normes et standards</li></ul>

## II)

1) Qualité d'un logiciel du point-vue utilisateurs:

1. Validité : (Logiciel doit être correct, juste et conforme) = doit répondre exactement aux exigences.

2. Fiabilité : La capacité à répondre sans planter (sans "Bog") à la rencontre d'une condition anormale. Par exemple : Division par 0  
• suppression d'un fichier inexistant.  
• Désconnection du réseau.

3. Utilisabilité : (Facilité d'utilisation) :

. Le logiciel doit être ~~assez~~ simple, facile à utiliser, intuitive et naturelle dans ces différentes fonctionnalités.

4. Efficacité :

. Un logiciel doit utiliser un minimum de ressources, matériels, temps machine, espace de stockage.

5. Disponibilité :

. Le logiciel est disponible tant les fois où certains de ces services sont requis.

## 6. Intégrité :

La capacité d'un logiciel à empêcher l'accès non autorisé à des informations ou des programmes. (c'est une mesure de sécurité)

## 7. Scalabilité :

Capacité d'un logiciel à traiter avec une demande croissante de données et de services.

## 8. Convivialité :

Pour cela on impose certaines contraintes lors de sa conception.

Ex :

- Interface graphique