



COURS « GÉNIE LOGICIEL 1 »

Niveau: II2

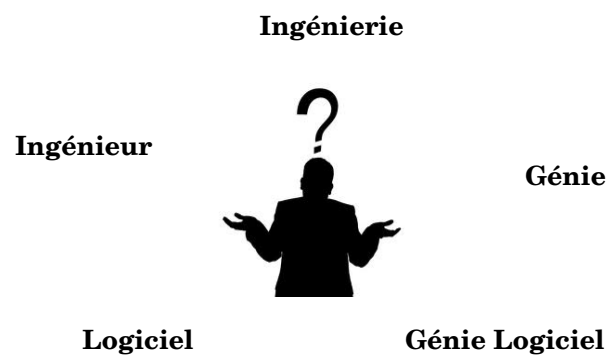
Enseignante: Rim DRIRA

drirarim@gmail.com

Bureau 215

AU: 2013/2014

TERMINOLOGIE



2

L'INGÉNIEUR, C'EST QUI?



- un professionnel chargé de **résoudre des problèmes de nature technique** en s'appuyant sur des **compétences avant tout scientifiques**, mais également **économiques et humaines**.



3

R.DRIRA

ENSI-GLI 2013-2014

Chapitre1:Introduction au GL

L'INGÉNIERIE, C'EST QUOI?

- On définit l'ingénierie comme étant le métier qui couvre **l'étendue des activités d'un ingénieur, de la naissance du besoin d'un «objet» à la fin de son utilisation.**



4

R.DRIRA

ENSI-GLI 2013-2014

Chapitre1:Introduction au GL

GÉNIE ?



- Génie= science de l'ingénieur (=engineering selon le dictionnaire *Hachette-Oxford*)
- utilisé en général accompagné d'un adjectif (exple: Génie civil, génie chimique, génie mécanique, etc.)
- **Génie:** les connaissances et techniques de l'ingénieur (*le Petit Robert*)
- **Génie** : l'application des sciences et mathématiques par laquelle les propriétés de la matière et les sources d'énergie de la nature sont rendues utiles à l'homme dans des constructions, machines, produits, systèmes et procédés (*traduit de l'anglais d'après le dictionnaire Webster*)

5

R.DRIRA

ENSI-GLI 2013-2014

Chapitre1:Introduction au GL

LE LOGICIEL, C'EST QUOI?

- Un **logiciel** est un ensemble d'**entités** nécessaires au fonctionnement d'un processus de traitement automatique de l'information.
- Parmi ces entités :
 - ✓ des programmes (en format *code source ou exécutables*) ;
 - ✓ des documentations d'utilisation ;
 - ✓ des informations de configuration.



6

R.DRIRA

ENSI-GLI 2013-2014

Chapitre1:Introduction au GL

QU'EST-CE QUE LE GÉNIE LOGICIEL

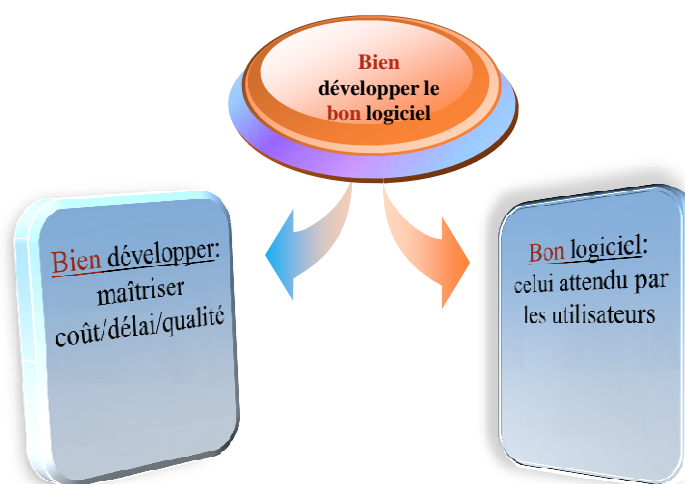
- l'application des sciences et mathématiques par laquelle les possibilités d'un équipement informatique sont rendues utiles à l'homme à l'aide de programmes, de procédures, de règles et de la documentation associée.[7]



L'art et la manière de construire un logiciel

7

ENJEUX DU GL?



8

OBJECTIFS DU COURS « GÉNIE LOGICIEL 1 »

- Avoir un aperçu général sur le génie logiciel et comprendre ses principes
- Etudier et comparer des processus de développement
- Etudier les étapes de développement
- Développer des capacités d'analyse, de conception et de tests

9

RÉFÉRENCES

- [1] I.Sommerville, **Le génie logiciel**, Addison-Wesley 1996
- [2] A. Strohmeier, **Génie logiciel : principes, méthodes et techniques**, Presse Polytechniques 1996
- [3] PressmanR.S., **Software Engineering : A practionner's Approach**, Mc Graw Hill, 2000
- Autres:
 - [4] I.Sommerville, **Le génie logiciel et ses applications**, InterEditions1991
 - [5] **Cours GL1, ENSI, 2012-2013**
 - [6] V.Berzins, **Software Engineering with Abstraction**, Addison-Wesley 1994
 - [7] G. Booch, **Ingénierie du logiciel avec ADA**, InterEditions1988
 - [8] Boehm B. W., **Software Engineering Economics**, PrenticeHall, 1982
 - [9] Journaux spécialisés : IEEE, ACM, ...

10



COURS « GÉNIE LOGICIEL 1 »

Chapitre1:Introduction au Génie Logiciel

Niveau: II2

Enseignante: Rim DRIRA

drirarim@gmail.com

Bureau 215

AU: 2013/2014

PLAN

- 1 Le logiciel
- 2 La crise du logiciel
- 3 Le génie logiciel

12

PLAN

- 1 Le logiciel
- 2 La crise du logiciel
- 3 Le génie logiciel

13

LE LOGICIEL

- Un **logiciel** est un ensemble d'**entités** nécessaires au fonctionnement d'un *processus de traitement automatique de l'information*.
- Parmi ces entités :
 - ✓ des programmes (en format *code source ou exécutables*) ;
 - ✓ des documentations d'utilisation ;
 - ✓ des informations de configuration.



14

LE LOGICIEL

- **Est un produit**
 - **une forme exécutable** : séquence d'instructions exécutables par une machine
 - **une forme non exécutable** : données, documentation (développement, installation, utilisation et maintenance)
- **Peut interagir avec des clients, qui peuvent être :**
 - des opérateurs humains (utilisateurs, administrateurs, etc.) ;
 - d'autres logiciels ;
 - des contrôleurs matériels.
- **Peut être développé par**
 - une personne seule;
 - une petite équipe;
 - un ensemble d'équipes coordonnées.

15

LE LOGICIEL

Classification des projets logiciels

- **projet trivial:** 1 programmeur, 1 à 4 semaines, moins de 500 lignes
- **petit projet:** 1 programmeur, 1 à 6 mois, 1'000 à 2'000 lignes
- **projet moyen:** 2 à 5 programmeurs, 1 à 2 ans, 5'000 à 50'000 lignes (ex: compilateurs, petits systèmes pour la gestion, etc.)
- **grand projet:** 5 à 20 programmeurs, 2 à 3 ans, 50'000 à 100'000 lignes (ex: grands compilateurs, gestionnaire de bases de données, etc.)
- **très grand projet:** 100 à 1'000 programmeurs, 4 à 5 ans, environ un million de lignes (ex: OS/360 d'IBM fut développé par 5'000 programmeurs, plus de 5 ans, plus d'un million de lignes)
- **extrêmement grand projet:** 2'000 à 5'000 programmeurs, 5 à 10 ans, 1 à 10 millions de lignes (ex: systèmes de contrôle de trafic aérien)

16

CLASSIFICATION DES LOGICIELS

Une 1^{ère} classification

- Deux grandes catégories :
 - **Logiciel générique** (progiciel) : système destiné à un même type d'application et conçu pour différents clients (les ERP).
 - **Logiciel sur mesure** : système spécifique développé pour les besoins d'un client.

17

CLASSIFICATION DES LOGICIELS

Une 2^{ème} classification

- **Logiciels systèmes:** réalisent des opérations basiques que peuvent effectuer les appareils informatiques (enregistrer, imprimer, allouer de l'espace mémoire, etc.)
- **Logiciels temps réels:** interagissent avec un environnement externe qui évolue avec le temps et réalisent certaines fonctionnalités en relation avec cet environnement dans des limites temporelles bien précises. (contrôleur d'un satellite de télécommunication)
- **Logiciels de gestion:** comptabilité, stock, bibliothèque
- **Logiciels scientifiques et logiciels d'ingénierie:** conception, calcul scientifique
- **Logiciels d'applications Web:** conteneur web, site Web, etc.
- **Logiciels d'intelligence artificielle:** aide à la décision, reconnaissance de forme (code barre, visage, empreinte digitale, de la parole, etc.)
- **Logiciels embarqués:** permettent de faire fonctionner une machine, équipée d'un ou plusieurs microprocesseurs, afin de réaliser une tâche spécifique avec une intervention humaine limitée. (guidage des lanceurs de satellites, pilote automatique dans les avions, etc.)

18

CLASSIFICATION DES LOGICIELS

Une 3^{ème} classification

Selon ce qu'offre l'éditeur du logiciel :

- La liberté d'utilisation
- La liberté de copie
- La liberté de modification et donc d'accès au code source
- La liberté de redistribution des modifications

19

CLASSIFICATION DES LOGICIELS

Une 3^{ème} classification

- **Logiciel propriétaire** : l'auteur se réserve le droit de diffuser et de modifier le logiciel. L'acquisition du droit d'utiliser le logiciel est souvent payante.
 - **Partagiciel** (*shareware*): logiciel qui s'accompagne de la permission de redistribuer des copies, mais toute personne qui continue à l'utiliser est obligé d'acheter une licence.
- **Logiciel gratuit** (*freeware*). L'auteur autorise l'utilisation et la copie par autrui gratuitement, mais garde le droit de modifier le logiciel.
- **Logiciel libre** : le code source est libre et modifiable, garantie la liberté de l'utilisateur dans son utilisation d'un logiciel.
 - Si un logiciel libre est sous la licence GNU GPL (Licence Publique Générale GNU), il est interdit, aux nouveaux distributeurs d'ajouter des restrictions supplémentaires sur les versions modifiées du logiciel.

20

QUALITÉ D'UN LOGICIEL

- Deux types :
 - Externes : Observables par l'utilisateur
 - Internes : concerne le développeur du logiciel
- Quelques facteurs de qualité d'un bon logiciel:
 - ✓ Validité
 - ✓ Fiabilité
 - ✓ Extensibilité (maintenance)
 - ✓ Réutilisabilité
 - ✓ Compatibilité
 - ✓ Efficacité
 - ✓ Portabilité
 - ✓ Vérifiabilité
 - ✓ Intégrité
 - ✓ Facilité d'emploi

21

QUALITÉ D'UN LOGICIEL

- **Validité ou conformité ou adéquation** : aptitude d'un produit logiciel à remplir exactement les fonctions définies par le cahier des charges et les spécifications.
- **Fiabilité** : aptitude d'un produit logiciel à fonctionner dans des conditions anormales sans dysfonctionnements et sans erreurs.
- **Extensibilité ou maintenance ou évolution** : facilité avec laquelle un logiciel se prête à sa maintenance, c'est-à-dire à une modification ou à une extension des fonctions qui lui sont demandées.
- **Réutilisabilité** : aptitude d'un logiciel à être réutilisé, en tout ou en partie, dans de nouvelles applications.

22

QUALITÉ D'UN LOGICIEL

- **Compatibilité** : facilité avec laquelle un logiciel peut être combiné avec d'autres logiciels.
- **Efficacité** : pas de gaspillage de ressources.
- **Portabilité** : facilité avec laquelle un logiciel peut être transféré sous différents environnements matériels et logiciels.
- **Vérifiabilité** : facilité de préparation des procédures de test.
- **Intégrité** : aptitude d'un logiciel à protéger son code et ses données contre des accès non autorisés.
- **Utilisabilité ou Facilité d'emploi** : facilité d'apprentissage, d'utilisation, de préparation des données, d'interprétation des erreurs et de rattrapage en cas d'erreur d'utilisation.

23

QUALITÉ D'UN LOGICIEL

- L'importance de ces facteurs de qualité variera selon la nature du projet de développement.
- Ces facteurs sont parfois contradictoires, le choix des compromis doit s'effectuer en fonction du contexte.

24

CONSTAT

- Systèmes informatiques
 - 80 % de logiciel
 - 20 % de matériel
- Depuis quelques années, la fabrication du matériel est assurée par quelques fabricants seulement
 - Le matériel est relativement stable
 - Le marché est standardisé
- **Les problèmes liés à l'informatique sont essentiellement des problèmes de Logiciel**

PLAN

- 1 Le logiciel
- 2 **La crise du logiciel**
- 3 Le génie logiciel

LA CRISE DU LOGICIEL

- A été décelée dans les années 70 (1968) et elle dure toujours
 - certains auteurs parlent d'une **maladie chronique**
- Les raisons de fond de cette crise: il est beaucoup plus difficile de créer des logiciels que le suggère notre intuition
 - On sous estime facilement **la complexité des logiciels** parce qu'ils sont essentiellement constituées de composants **immatériels** (programmes, données, documentation, etc.)

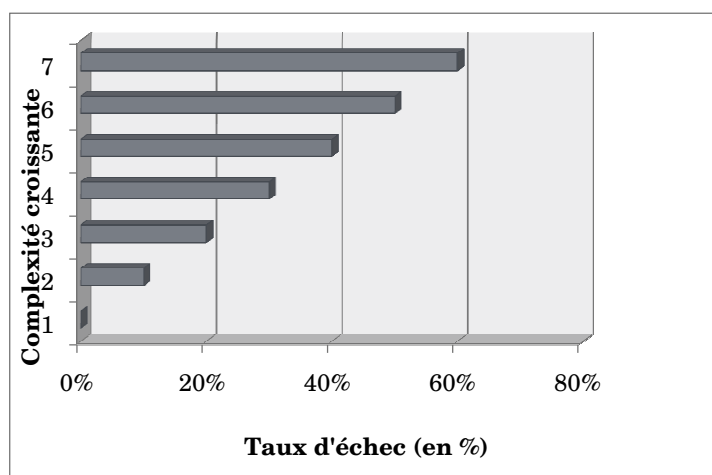
27

R.DRIRA

ENSI-GLI 2013-2014

Chapitre1:Introduction au GL

LOGICIELS DE PLUS EN PLUS COMPLEXES



Le taux de succès décroît avec la taille des projets

28

R.DRIRA

ENSI-GLI 2013-2014

Chapitre1:Introduction au GL

LOGICIELS DE PLUS EN PLUS COMPLEXES

○ Exemples d'échecs

- Confirm (1992) : projet d'American Airlines de système de réservation commun (avions, voitures, hôtels, etc.).
 - Investissement : **125 Millions \$** sur **4 ans, plus de 200 ingénieurs**.
 - Résultat : les différentes parties n'ont pas pu être assemblées en raison de la **diversité des méthodes de développement**.
- Taurus (1993) : projet d'automatisation des transactions pour la bourse de **Londres annulé après 5 années de développement**.
 - Pertes estimées a **75 Millions \$** pour la société et **450 Millions \$** pour les clients.

○ Exemple de bug qui a conduit à une catastrophe

- Ariane V (1996) : explosion de la fusée en vol due a **une erreur de débordement lors de la conversion d'un nombre flottant 64 bits vers un entier 16 bits**. Code hérité de Ariane IV. Coût : **500 Millions \$**.

29

LOGICIELS DE PLUS EN PLUS COMPLEXES: POURQUOI?

○ Évolutivité croissante

- De la partie matérielle
- De la partie métier:
 - La diminution des coûts du matériel (hardware) a conduit à associer l'ordinateur à de plus en plus de produits.
 - Les domaines de plus en plus sensibles : nucléaire, systèmes bancaires, télécommunications, etc.

○ Volume croissant

- Des données
- Du code

○ Hétérogénéité croissante

- Des supports de données et des protocoles d'accès
- Des plates-formes, des langages
- Des technologies
 - Le rythme d'arrivée des nouvelles technologies s'accélère sans cesse
 - Les vieilles technologies persistent

30

LOGICIELS DE PLUS EN PLUS COMPLEXES

Volume croissant du code: quelques exemples

- Chez Thomson, les programmes sont passés en 5 ans de 20.000 lignes assembleur en moyenne à 150.000 lignes ADA.
- La gestion d'un central téléphonique comporte 1 million de lignes.
- La navette spatiale en demande 50 millions, soit plusieurs milliers «d'hommes-années».

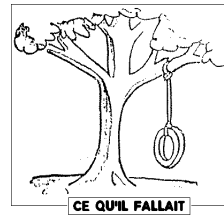
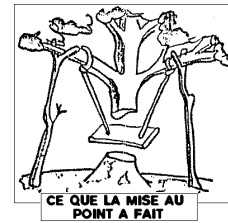
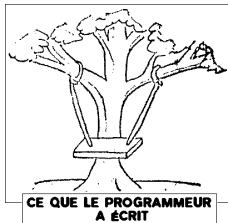
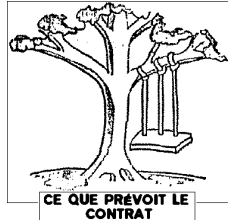
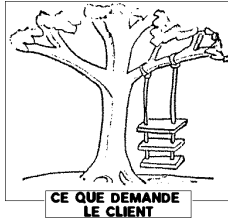
31

LA CRISE DU LOGICIEL: SYMPTÔMES

- **Inadéquation**
 - Logiciels réalisés \neq besoins utilisateurs
- **Délais**
 - Logiciels souvent livrés en retard
- **Coûts**
 - Dépassement du budget prévu
- **Fiabilité**
 - Logiciels souvent en panne
- **Maintenance**
 - Difficile et coûteuse

32

SYMPTÔMES : INADÉQUATION



33

R.DRIRA

ENSI-GLI 2013-2014

Chapitre1:Introduction au GL

SYMPTÔMES : INADÉQUATION

- 1979: USA, *Government Accounting Office*, enquête sur un certain nombre de projets

- [47%] délivré, jamais utilisé
- [29%] payé, non fourni
- [19%] utilisé puis modifié ou abandonné
- [3%] utilisé après quelques retouches
- [2%] utilisé tel quel

34

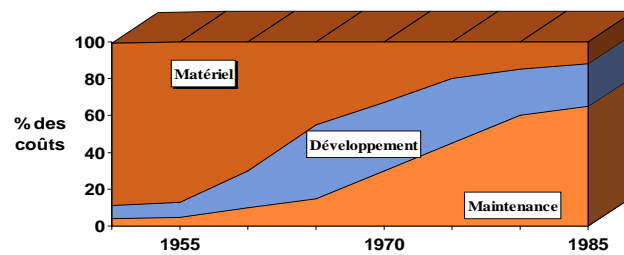
R.DRIRA

ENSI-GLI 2013-2014

Chapitre1:Introduction au GL

SYMPTÔMES: COÛT

- Coût élevé et investissements non rentabilisés
- Les coûts des logiciels connaissent une très forte croissance, **dominant** maintenant le coût du matériel et peuvent représenter plus de 80 % du coût total d'un système informatique.



35

R.DRIRA

ENSI-GLI 2013-2014

Chapitre1:Introduction au GL

SYMPTÔMES : DÉLAI & COÛT

- Enquête dans les années 70 sur 100 programmes:
 - délais dépassés : 52%
 - budgets logiciels dépassés : 72%
 - budgets matériels dépassés : 15%
- Exemple
 - L'OS-360 d'IBM fut livré en retard, nécessitait plus de mémoire que prévu, son coût dépassait les estimations et il comportait de nombreuses erreurs (années 60 - Fred Brooks)

36

R.DRIRA

ENSI-GLI 2013-2014

Chapitre1:Introduction au GL

SYMPTÔMES : FIABILITÉ

- Exemples de projets non fiables
 - Sonde Mariner I vers Vénus perdue suite à une erreur d'un programme Fortran (1962)
 - Expérience météorologique en France: 72 ballons contenant des instruments de mesures furent détruits suite à une erreur de programmation (1971)
 - Un défaut logiciel retarde le lancement de la navette spatiale américaine (1981)
 - Un changement de version logiciel des centraux téléphoniques ATT empêche les appels longue distance (1990)
 - Octobre 2004 : Importantes perturbations du réseau France Telecom pendant 48h
 - Novembre 2004 : paralysie totale du réseau Bouygues Telecom pendant une vingtaine d'heures
 - Les pannes causées par le « bug de l'an 2000 » ont coûté environ 175 milliards de dollars aux entreprises du monde, (Le Monde – 23 oct. 2001).

37

R.DRIRA

ENSI-GLI 2013-2014

Chapitre1:Introduction au GL

SYMPTÔMES : MAINTENANCE

- L'inflation des coûts des logiciels est surtout due aux frais de maintenance :
 - situés entre 40% et 75% du coût total d'un système informatique (matériel et logiciel)
- La maintenance est souvent à l'origine de nouvelles erreurs

38

R.DRIRA

ENSI-GLI 2013-2014

Chapitre1:Introduction au GL

SYMPTÔMES : MAINTENANCE

- D'une étude sur 487 sites de toutes sortes de développement de logiciels, il ressort que:
 - 42% des modifications sont dues à des demandes de l'utilisateur ;
 - 30% des exigences de spécification changent entre la 1ère édition d'une documentation et la première sortie du produit ;
 - 17% à des changements de format des données ;
 - 12% problèmes d'urgence résolus à la hâte ;
 - 9% débogages de programmes ;
 - 6% à des changements de matériel ;
 - 6% problèmes de documentation ;
 - 4% améliorations de l'efficacité ;
 - 3% autres...

39

GÉNIE CIVIL VS. GÉNIE LOGICIEL

- Le pont de Verrazano Narrow Bridge à New York (1959)
 - prix estimé à 325 millions de \$ pour 1965.
 - terminé en novembre 1964 et le budget ne fut pas dépassé;
- le projet OS/360 d'IBM a eu des années de retard.
- **Pourquoi la prévision pour la construction d'un pont est-elle exacte, alors qu'il y a de tels problèmes avec le logiciel ?**
 - un ingénieur en génie civil voit plus facilement le surplus de complexité lié à la réalisation d'un **énorme pont**, qu'un informaticien ne voit la complexité d'un **énorme programme**.

40

QUE FAIRE ?

- Sachant que ne rien faire est une option de plus en plus coûteuse et dangereuse
- Sachant que la complexité a atteint un tel niveau qu'il est hors de portée d'un seul individu d'avoir une vision globale sur un système en évolution
 - Accélérer encore la course aux nouvelles technologies dans l'espoir qu'une nouvelle solution magique se présente ? Fuite en avant ?
 - Pour que la réalisation d'un logiciel ne soit pas un échec, il est important d'utiliser des techniques adéquates lors de son développement. Ceci est d'autant plus nécessaire que le logiciel est important (par sa taille).

41

NAISSANCE DU GÉNIE LOGICIEL

- **Le génie logiciel :**
 - La notion de génie logiciel a été mentionnée pour la première fois lors d'une conférence concernant la crise du logiciel en 1968
 - Conférence à Garmish en Allemagne (7-10-1968/11-10-1968) sous le patronage de l'OTAN (Organisation du Traité de l'Atlantique Nord)
 - 45 ans déjà, seulement!

42

PLAN

- 1 Le logiciel
- 2 La crise du logiciel
- 3 Le génie logiciel

43

QU'EST-CE QUE LE GÉNIE LOGICIEL

- **Une première définition:** l'application des sciences et mathématiques par laquelle les possibilités d'un équipement informatique sont rendues utiles à l'homme à l'aide de programmes, de procédures, de règles et de la documentation associée.[7]
- **Une deuxième définition:** cherche à établir et à utiliser des **principes sains d'ingénierie** dans le but de développer **économiquement** du logiciel qui **satisfait les besoins**, qui est **fiable** et qui fonctionne efficacement sur des machines réelles (D'après Fritz Bauer, 1969)



L'art et la manière de construire un logiciel

44

LE GÉNIE LOGICIEL

- Une Discipline (= méthodes, techniques et outils)
 - basée sur le savoir (théorique)
 - le savoir-faire (pragmatique)
 - et le faire savoir (communication)
 - pour produire (développement)
 - de façon industrielle (taille, diffusion)
 - des logiciels (= les produits)
 - *de qualité au meilleur prix et en respectant les délais.*

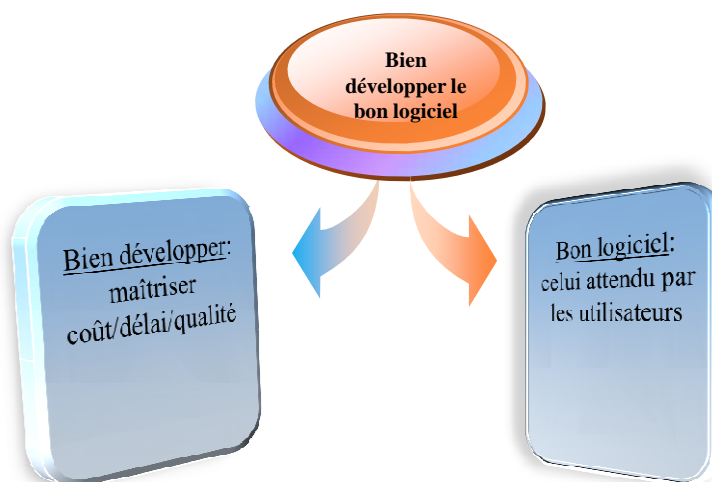
45

R.DRIRA

ENSI-GLI 2013-2014

Chapitre1:Introduction au GL

OBJECTIFS



46

R.DRIRA

ENSI-GLI 2013-2014

Chapitre1:Introduction au GL

OBJECTIFS

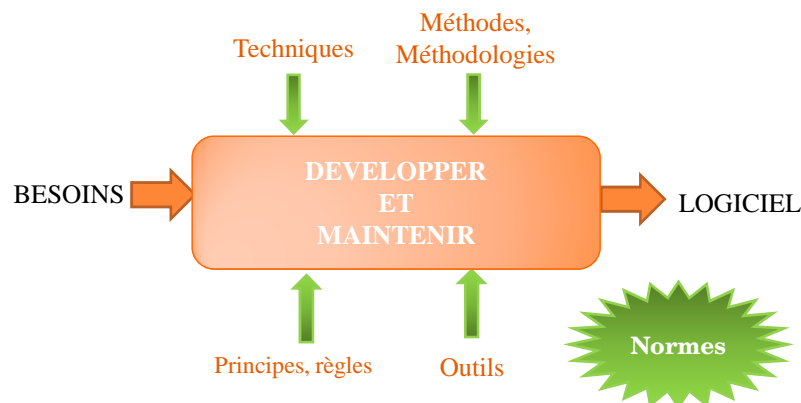
Le génie logiciel vise à garantir que :

- le logiciel respecte sa spécification ;
- la spécification répond aux besoins réels de ses clients ;
- les coûts alloués pour sa réalisation sont respectés ;
- les délais de réalisation sont respectés.
- le logiciel est de bonne qualité.

47

COMMENT BIEN CONSTRUIRE LE BON LOGICIEL?

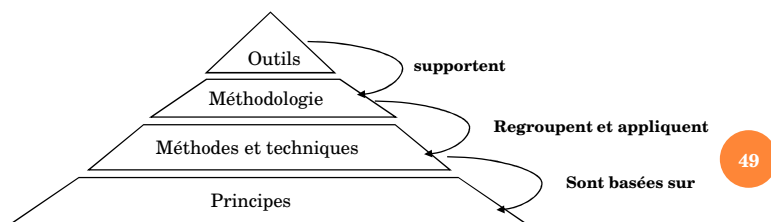
- Le génie logiciel est un domaine en pleine évolution qui offre une grande palette d'outils et de méthodes pour parvenir à construire des logiciels de qualité.



48

DÉFINITIONS

- **Principe** = loi générale régissant un ensemble de phénomènes et vérifiée par l'exactitude de ses conséquences
- **Règle** = convention qui s'impose dans un cas donné
- **Technique** = Ensemble de moyens pratiques propres à une activité souvent associée à un savoir-faire .
- **Méthode** = un processus discipliné qui génère un ensemble de modèles décrivant les différents aspects d'un logiciel en utilisant une notation bien définie.
- **Méthodologie** = ensemble structuré et cohérent de méthodes et d'outils permettant de déduire la manière de résoudre un problème.
- **Norme** = un document établi par consensus, qui fournit, pour des usages communs et répétés, des règles, des lignes directrices ou des caractéristiques, pour des activités ou leurs résultats, garantissant un niveau d'ordre optimal dans un contexte donné (extrait du Guide ISO/CEI 2).



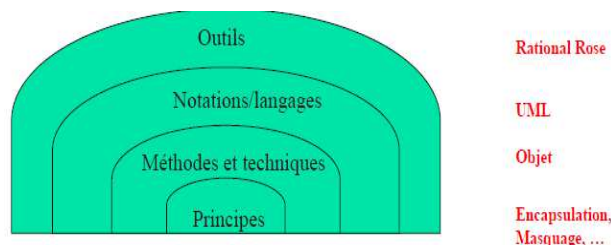
R.DRIRA

ENSI-GLI 2013-2014

Chapitre1:Introduction au GL

DÉFINITIONS

- Illustration



50

R.DRIRA

ENSI-GLI 2013-2014

Chapitre1:Introduction au GL

NORMES DU GL

- Comme toute science de l'ingénieur, le GL est régi par des normes.
- L'IEEE (Institute of Electrical and Electronics Engineers) a élaboré une classification des normes du GL ayant la référence ANSI/IEEE 1002 selon le contenu de ces normes: Produit, Profession, Processus, Notation.
 - Produits: les résultats documentés des activités de développement et de maintenance.
 - Processus: décrivent la suite d'actions à entreprendre pour développer un produit.
 - Profession: ces normes traitent des aspects du GL qui en font une profession (ex: plan d'étude)
 - Notation: décrivent le moyen de communication entre les professionnels.

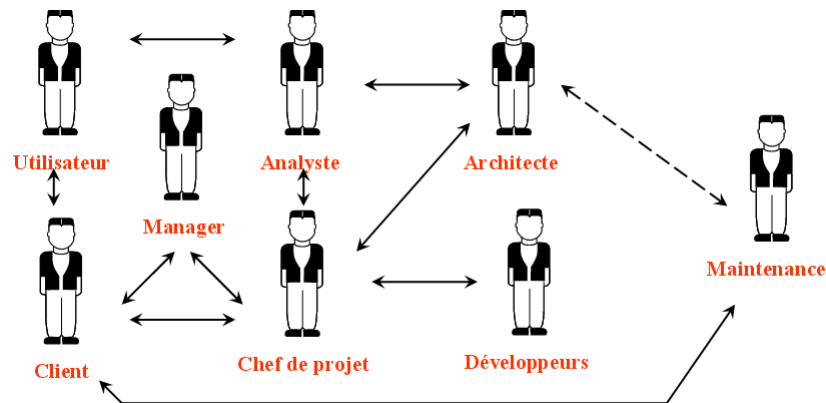
51

LES ACTEURS

- Différents acteurs interviennent, parmi lesquels:
 - Utilisateurs
 - Clients
 - Manager/ingénieur d'affaires: gérer la relation avec les clients
 - Chef de projet: suivre et superviser toutes les étapes de développement (suivi et contrôle)
 - Architecte: S'occupe de l'architecture logicielle
 - Analyste: Analyser
 - Développeur: Développer
 - Maintenance:Maintenir
- Ils n'ont pas les mêmes intérêts
- **Élément clef : la communication**
- Une personne peut assumer plusieurs rôles.
- Plusieurs personnes peuvent assumer un même rôle

52

LES ACTEURS



Organiser le travail entre différents acteurs est l'un des apports du GL

53

R.DRIRA

ENSI-GLI 2013-2014

Chapitre1:Introduction au GL

SYNTHÈSE

- La production de logiciel est beaucoup plus qu'une activité personnelle de programmation:
 - Les programmes ne sont qu'une partie du logiciel
 - Logiciels de plus en plus complexe: Travail en équipe, planification, documentation, assurance qualité, etc.
- Le génie logiciel est une démarche d'ingénierie qui traite tous les aspects de la production de logiciels
 - du cahier des charges jusqu'aux activités de maintenance
 - dépasse le cadre purement technique
- Le GL vise à assurer la production de logiciels en respectant le coût, l'échéance et la qualité.

54

R.DRIRA

ENSI-GLI 2013-2014

Chapitre1:Introduction au GL

SYNTHÈSE

- Les méthodes, outils, méthodologies, techniques du GL sont nombreux.
- En fonction de leur nature, les projets ont des besoins très différents
 - Le produit souhaité et le contexte déterminent la bonne approche
 - Une bonne approche aidera à livrer un logiciel de qualité tout en respectant les délais et en minimisant les coûts.
 - Le travail d'un ingénieur en informatique est de choisir la bonne approche de GL en fonction du projet

55