



Licence Sciences, Technologies, Santé
Mention SPI Parcours Informatique

Introduction au Génie Logiciel
L3 / 175EN002

C1 - Problématique du développement informatique

Thierry Lemeunier
thierry.lemeunier@univ-lemans.fr

Plan du cours

- L'industrie du logiciel
- Objectifs du génie logiciel
- Définition du génie logiciel
- La métaphore du génie civil
- La qualité au cœur du processus
- Mesurer la maturité

L'industrie du logiciel (1/3)

- ❑ Industrie du logiciel = création, correction et amélioration des logiciels
- ❑ Logiciels : programmes informatiques s'exécutant sur une machine réelle ou virtuelle (machine simulée)
- ❑ Machine réelle :
 - cela va des supercalculateurs
 - aux plateformes mobiles (téléphone, tablette, ordinateur de bord, etc.)
 - en passant par les ordinateurs « classiques » PC (*Personal Computer*) de bureau ou portable



L'industrie du logiciel (2/3)

□ Les trois acteurs de l'industrie du logiciel :

■ Editeur de logiciel

- Logiciels « standards » (progiciels) pour l'entreprise et pour les particuliers
- Exemples : logiciels de gestion, éditeurs, jeux, etc.
- Exemple d'éditeurs : Microsoft, Apple, IBM



■ Société de services et d'ingénierie informatique (SSII)

- Logiciels « sur mesure » généralement pour les entreprises
- Trois types de mode de travail :
 - Contrat en forfait : la SSII fournit un service clé en main au client
 - Contrat en régie : la SSII délègue ses employés chez le client (sorte d'intérim)
 - Contrat en centre de service : le centre regroupe des employés pour un « gros » client ou pour un domaine particulier ou pour une ou plusieurs technologies
- Exemples : gestion de contrats d'assurance, application ou site web, etc.
- SSII européennes : Sopra Group, Capgemini, Atos Origin, Logica, etc.



L'industrie du logiciel (3/3)

□ Les trois acteurs de l'industrie :

■ Communauté du logiciel libre

- Logiciels non propriétaires : tout le monde peut accéder aux codes sources et le modifier librement (enfin ceux qui en sont capables...)
- Un logiciel libre ne peut devenir un logiciel propriétaire
- Un logiciel libre peut être commercialisé
- Les logiciels libres sont développés bénévolement ou par des salariés
- Exemple de logiciel libre : GNU/Linux, The Gimp, Emacs, OpenOffice, etc.



□ Deux remarques :

- Un éditeur ou une SSII peut proposer et utiliser du logiciel libre
- Une société peut être à la fois éditeur et une SSII
- Une société peut être à la fois éditeur de logiciel et constructeur de matériel (Apple, IBM par exemples)

Objectifs du génie logiciel (1 / 2)

- ▣ Le génie logiciel a pour but de passer de ***l'artisanat à l'industrie***

Programming in the small ⇒ *Programming in the large*

Amateurs	⇒	Professionnels
Travail d'une personne	⇒	Une ou plusieurs équipes
Utilisateur = programmeur	⇒	Clients / fournisseurs
Produit = code (documenté ?)	⇒	Multitude de produits
Processus simple de production	⇒	Processus complexe

- ▣ L'objectif est de parvenir *in fine* à une **maîtrise parfaite du processus de développement**

Objectifs du génie logiciel (2/2)

- Pour maîtriser un développement logiciel :
 - Il faut suivre des méthodes strictes réutilisables intégrant un contrôle de la qualité
 - Ecrivez ce que vous faites
Documenter, documenter et encore documenter
 - Faites ce qui est écrit
Lire, lire et encore lire
 - Prouvez que vous le faites
Tester, tester et encore tester
- La maîtrise du développement permet de respecter la règle CQFD :
 - Les **Coûts** restent dans les limites prévues
 - Le système est de **Qualité**
 - Les **Fonctionnalités** répondent aux besoins
 - Les **Délais** restent dans les limites prévues

Règle CQFD = Coût Qualité Fonctionnalités Délai

Définition du génie logiciel (1 / 3)

□ Produit logiciel

Le programme + les produits nécessaires
pour sa **production**
son **exploitation**
sa **maintenance**

□ Processus logiciel

Ensemble d'**activités** effectuées
de **méthodes**
et de **pratiques** suivies

pour la production et la maintenance
du produit logiciel

Définition du génie logiciel (2/3)

- ❑ Le processus logiciel se compose de 4 branches :
 - *Le processus technique*
 - ❑ Il modélise la procédure à suivre pour réaliser le produit
 - ❑ Il est décrit par un cycle de vie
 - ❑ Il comporte notamment les activités suivantes : analyse, conception, codage & tests, validation
 - *Le processus de gestion*
 - ❑ Il modélise la procédure à suivre pour contrôler les délais et les coûts et gérer les équipes
 - ❑ Il comporte notamment les activités suivantes :
 - Estimation des durées, coûts et effort humains nécessaires pour réaliser chaque activité du cycle
 - Mesures d'avancement pour effectuer le suivi et le contrôle du déroulement du projet pour anticiper les écarts et rétablir la situation

Définition du génie logiciel (3/3)

■ *Le processus qualité*

- Il modélise la procédure à suivre pour :
 - Garantir la qualité du produit
 - Garantir que le processus de gestion est réalisé comme prévu
 - Garantir que le processus qualité s'effectue correctement
- La garantie qualité logiciel comporte les activités suivantes : définir les exigences du client, contrôler que ces exigences sont bien prises en compte dans les artefacts du cycle, améliorer *a posteriori* le processus technique

■ *Le processus de gestion des risques*

- Il modélise la procédure à suivre pour analyser les risques (techniques, de gestion ou de qualité)
- Processus peu connu mais qui se développe
- Il comporte les activités suivantes : identification du risque, évaluation du risque, réaction au risque, apprentissage

La métaphore (1/6)

□ La métaphore du génie civil :

- Développer un *logiciel* peut être comparé à la construction d'une *maison*

■ Comparons les étapes :

1. Analyser et comprendre les besoins

- Concevoir une maison adaptée aux personnes qui l'habitent, esthétiquement, fonctionnellement
- Élaborer un logiciel adapté aux besoins de l'utilisateur ; ce n'est pas le plus simple à faire : l'utilisateur ne sait pas toujours précisément ce qu'il veut et les besoins peuvent évoluer au cours du projet ; c'est le rôle de l'analyse des besoins



La métaphore (2/6)

2. Tenir compte des réalités économiques (le budget disponible)
 - ▣ Du marbre partout et 12 chambres n'est pas à la portée de toutes les bourses...il faut être capable de faire une prévision du coût de la maison.
 - ▣ Le logiciel est un produit abstrait dont il est difficile de mesurer le coût a priori ; c'est le rôle des techniques d'estimation du logiciel et de l'analyse de risque.
3. Proposer des solutions architecturales
 - ▣ Différentes propositions sont faites quant à l'agencement des pièces, de l'esthétique extérieure, de l'aménagement intérieur.
 - ▣ On étudie différentes solutions techniques pour répondre aux besoins exprimés ; c'est le rôle de la conception.

La métaphore (3/6)

4. Contacter tous les corps de métier nécessaires
 - ▣ Maçons, couvreurs, plombier, électricien, plâtrier, décorateur, etc.
 - ▣ Ergonome, spécialistes de base de données, de distribution, de télécommunication, de tests, d'interface homme-machine.
5. S'assurer que toutes les dispositions/contraintes légales sont satisfaites
 - ▣ Plan d'occupation des sols, permis de construire, etc.
 - ▣ Droits d'utilisation des logiciels, loi informatique et liberté.
6. Planifier l'enchaînement des interventions

La métaphore (4/6)

7. Faire faire les travaux

- Prévenir les différents corps de métier des dates d'intervention.
- C'est le rôle de la réalisation, du **codage**

8. S'assurer de l'avancement des travaux

9. S'assurer de la qualité des travaux réalisés

- ▣ Vis-à-vis des besoins exprimés, vérifier que ce qui est réalisé est conforme. C'est le rôle de la qualité et des tests

10. Gérer les éventuels conflits

11. Prévoir la recette

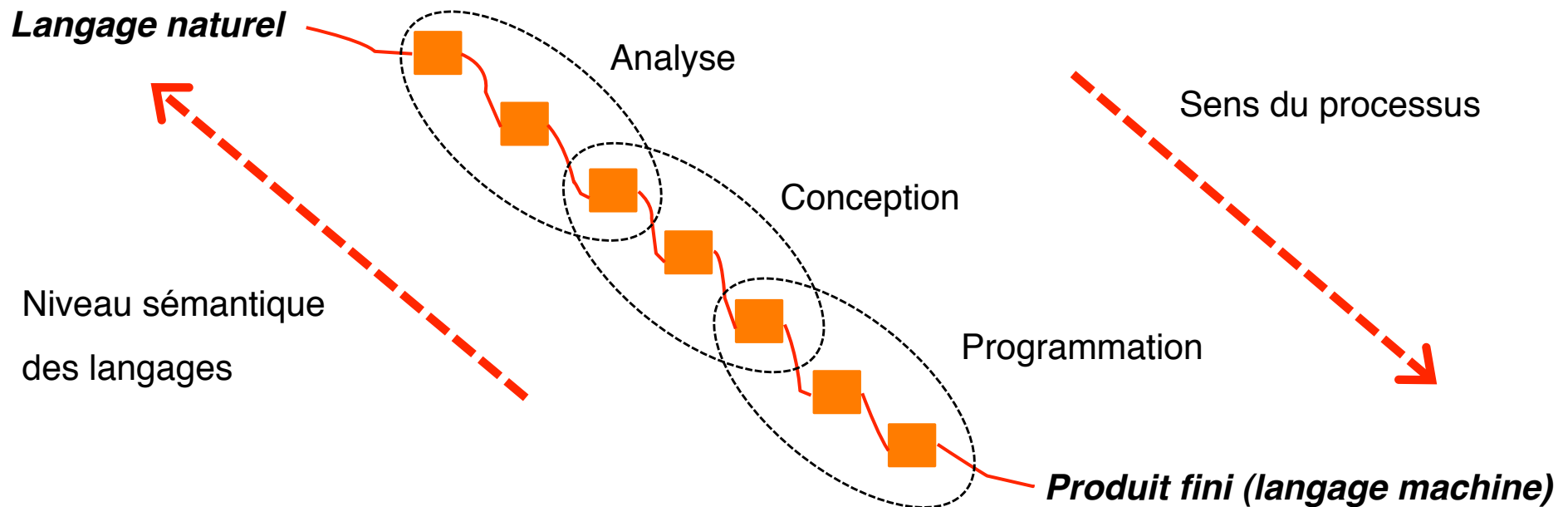
- ▣ Remise des clés de la maison
- ▣ Tests fonctionnels du logiciel

La métaphore (5/6)

- Pour **construire** / **développer** de nombreux outils seront utiles ou nécessaires :
 - Une **méthode** et des **principes**
 - Comment faire pour comprendre les besoins, comment concevoir, comment ne rien oublier, comment assurer la cohérence du tout, etc.
 - Des **documents**
 - Contrats précisant les engagements de chacun
 - Plans (élévations, détails, câblages) utiles à la description des travaux
 - Analyse du domaine, architectures logiciel et matérielle, schémas de conception, spécifications des IHM, etc.
 - Des **outils** spécifiques à chaque corps de métier
 - Pelleteuse, truelle, ciment, peinture, crayon, etc.
 - Editeurs, compilateurs, débogueurs, logiciel de gestion de version, etc.
 - La **compétence** et l'**expérience** des agents humains

La métaphore (6/6)

- En substance, le processus de développement consiste à traduire en **langage machine** des idées exprimées en **langage naturel**.
- La difficulté est due à la différence de **niveau sémantique** entre les deux langages : il faut passer par des langages de niveau sémantique intermédiaire.



La qualité – Définition

- ❑ La notion de qualité est une notion générale présente dans toute industrie
- ❑ Une entreprise sans démarche qualité est vouée à stagner ou à périr : accepteriez-vous de monter dans un avion dans lequel *presque tout* a été vérifié ?



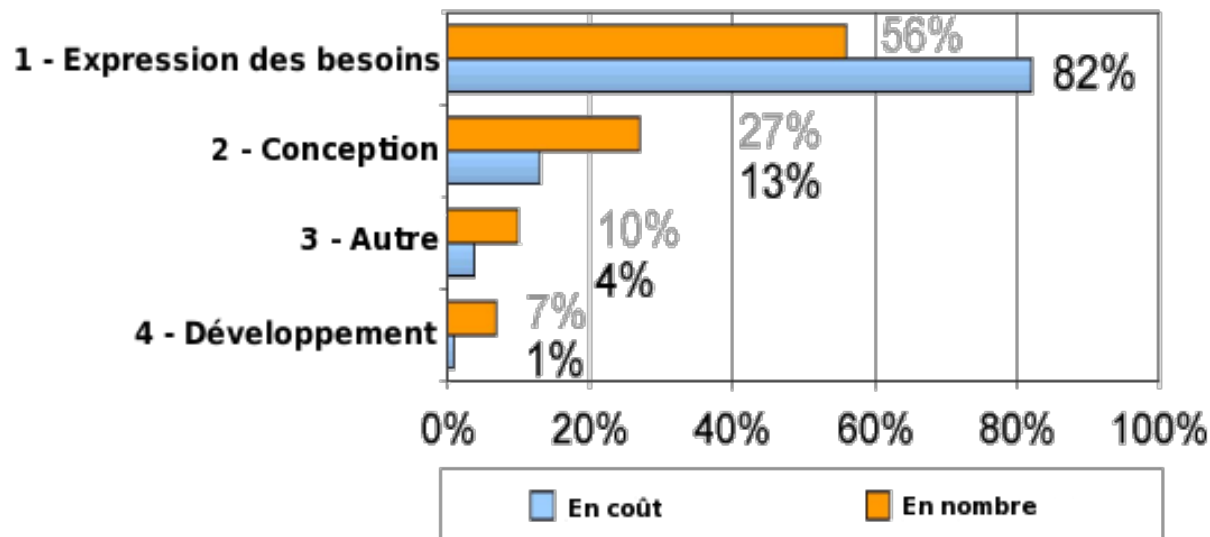
- ❑ Définition de l'ISO :
 - Aptitude d'un ensemble de **caractéristiques** intrinsèques d'un produit, d'un système ou d'un processus à **satisfaire** les exigences des **clients** et autres **parties** intéressées
 - Une caractéristique peut être physique, comportemental, temporel, ergonomique ou fonctionnelle
 - Une exigence peut correspondre à un besoin ou une attente implicite ou explicite

La qualité – Apports



- Répondre aux besoins des clients
- Être économe :
 - Le coût de la résolution d'une erreur augmente au fil du temps : plus l'erreur est corrigée tôt moins elle coûte à corriger

Répartition du nombre d'erreurs et de leur coût



[Source CHAOS 2009]

- Au final, un produit de qualité demande moins de correction et moins de maintenance

La qualité – Principes

- ❑ La qualité est une affaire de tout participant tout au long du projet
- ❑ Elle ne doit pas être considérée comme accessoire ou comme une obligation mais comme une opportunité de mieux travailler :
 - ❑ Il doit exister dans l'entreprise une culture de la qualité entretenue de manière centrale et continue (amélioration constante)
 - ❑ Faire de la qualité permet :
 - D'avoir la reconnaissance des collègues et de ses supérieurs
 - D'avoir de la satisfaction (moins de stress...)
 - ❑ Au contraire, faire un travail de moindre qualité entraîne :
 - Le stress (heures supplémentaires, recommencer plusieurs fois...)
 - L'insatisfaction (projets non rentables, clients mécontents...)
 - Des dysfonctionnements
- Comment faire des produits de qualité ?
 - Identifier les caractéristiques d'un produit de qualité
 - Définir et utiliser un processus qualité
 - Former les personnes à la qualité



La qualité – Facteurs de qualité (1/3)

- ❑ Le produit logiciel ne se détériore pas : sa qualité est uniquement due à sa construction
- ❑ On évalue la qualité selon :
 - Des facteurs externes
 - ❑ Directement observables par l'utilisateur
 - Des facteurs internes
 - ❑ Observables uniquement par le développeur
- ❑ Certains facteurs sont mesurables : les métriques
 - Pris en compte par le processus qualité
 - Les métriques permettent d'évaluer certains facteurs
 - Exemples de métriques du codage :
 - ❑ Nombre de lignes de code, d'opérandes, d'opérations
 - ❑ Complexité cyclomatique (nombre de chemins possibles dans une fonction ou une méthode)
 - ❑ Taux de commentaires



La qualité – Facteurs de qualité (2/3)

- Les facteurs de qualité en trois catégories :
 - Qualités opérationnelles
 - **Conformité** : satisfaire les besoins/spécifications
 - **Robustesse** : accomplir sans défaillance l'ensemble des fonctionnalités, dans un environnement opérationnel de référence et pour une durée d'utilisation donnée
 - **Efficacité** : se limiter à l'utilisation des ressources strictement nécessaires à l'accomplissement de ses fonctionnalités
 - **Sécurité** : surveiller, recenser, protéger et contrôler les accès aux codes sources et aux données
 - **Maniabilité** : minimiser l'effort nécessaire pour l'apprentissage, la mise en œuvre des entrées et l'exploitation des sorties

La qualité – Facteurs de qualité (3/3)

□ Les facteurs de qualité (suite)

■ Facilité à changer ou à corriger

- **Maintenabilité** : minimiser l'effort pour localiser et corriger les erreurs
- **Adaptabilité** : minimiser l'effort nécessaire pour le modifier par suite d'évolution des spécifications
- **Testabilité** : faciliter les procédures de test permettant de s'assurer de l'adéquation des fonctionnalités

■ Facilité à faire des transitions d'environnement

- **Portabilité** : minimiser l'effort pour se faire transposer dans un autre environnement matériel et/ou logiciel
- **Réutilisabilité** : être partiellement ou totalement utilisé dans une autre application
- **Interopérabilité** : s'interconnecter à d'autres produits

□ Les facteurs de qualité sont interdépendants !

Mesurer la maturité (1/2)

□ Le zéro bug n'existe pas :

- La validation de Windows 2000 avait fait appel à 600 000 bêta testeurs. Il restait pourtant au lancement 63 000 problèmes potentiels dans le code
- Un logiciel commercial type contient entre 20 et 30 bugs pour mille lignes !
- Tant que la programmation sera manuelle il aura des bugs !



□ Mesurer la maturité :

- La maturité d'un logiciel est directement fonction de la maturité de l'équipe qui l'a développé
- Le modèle CMMI (*Capability Maturity Model + Integration*)
 - Proposé par le *Software Engineering Institute* de *Carnegie Mellon University*
 - Il définit le niveau d'organisation d'une entreprise
 - Le modèle définit 5 niveaux (cf. tableau suivant)

Mesurer la maturité (2/2)

Niveau de maturité	Pratiques
Niveau 1 Initial	Niveau où le résultat final est imprévisible. L'atteinte des résultats repose plus sur les hommes, sur leur engagement et bonne volonté, que sur l'application disciplinée de bonnes pratiques. Une réussite éventuelle ne sera pas nécessairement reproductible. L'évaluation de l'efficacité et des performances est absente. La direction n'établit pas de plan ou de vision qui sont liés à des besoins. La documentation est inexistante.
Niveau 2 Discipliné	Une discipline est établie pour chaque projet et se matérialise essentiellement par des plans de projet (plan de développement, d'assurance qualité, de gestion de configuration ...). Le chef de projet a une forte responsabilité dans le niveau 2 : il doit définir, documenter, appliquer et maintenir à jour ses plans. D'un projet à l'autre, il capitalise et améliore ses pratiques de gestion de projet et d'ingénierie.
Niveau 3 Ajusté	Ce niveau est caractérisé par une standardisation adéquate des pratiques, une capitalisation centralisée (en particulier sur les mesures réalisées dans les projets) et une maîtrise du référentiel interne (ou Système Qualité). Il existe des lignes directrices, un plan stratégique et une planification de l'amélioration de processus. Les employés sont formés et conscients de leurs responsabilités ainsi que de leurs devoirs.
Niveau 4 Géré Quantitativement	Les projets sont pilotés sur la base d'objectifs quantitatifs de qualité produit et processus. La capacité des activités (ou sous-processus) critiques est déterminée par l'organisation, ainsi que les modèles de performance et de prévision associés. L'expression de la qualité demandée par le client est prise en compte pour quantifier les objectifs du projet et établir des plans selon la capacité des processus de l'organisation.
Niveau 5 Optimisé	Les processus qui sont gérés quantitativement pour le pilotage de projet (niveau de maturité 4) sont en optimisation constante afin d'anticiper les évolutions prévues (besoins clients, nouvelles technologies...).

[Source Wikipedia]