IFT6803:
Génie logiciel du commerce électronique

Chapitre 1: Introduction

Section 1 :Génie logiciel: objectifs, défis et principes

Julie Vachon,
Hiver 2003

Sommaire

Chapitre 1, Section 1

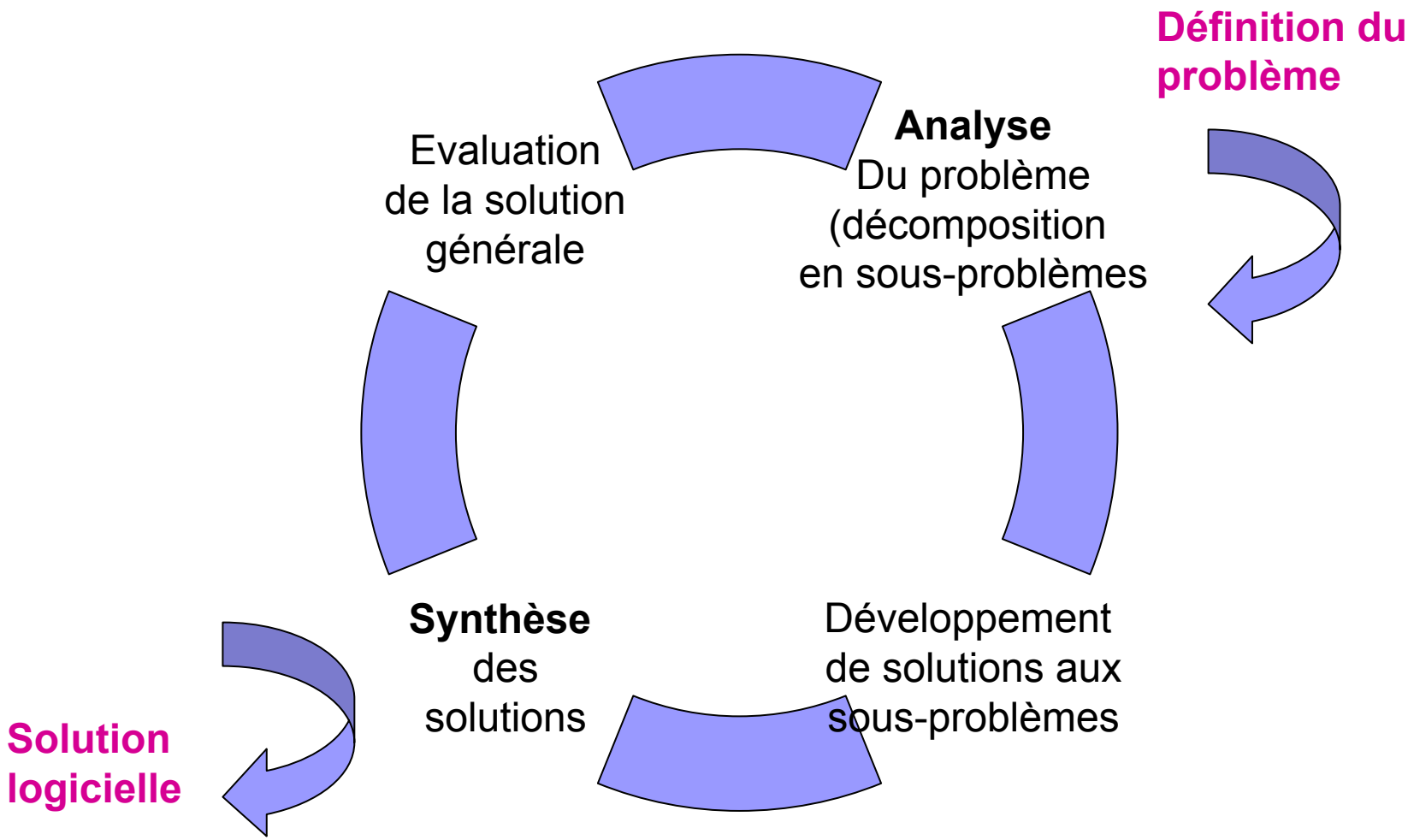
« Génie logiciel: objectifs, défis et principes »

1.1.1 Qu'est-ce que le génie logiciel?

Génie logiciel:

« Discipline de l'informatique qui regroupe un ensemble de connaissances, de procédés et des acquis scientifiques pour la conception, la mise en œuvre, la vérification et la documentation de logiciels dans le but d'en optimiser la production, le support et la qualité ». (Grand dict. terminologique)

Le génie logiciel c'est un processus de résolution de problème…

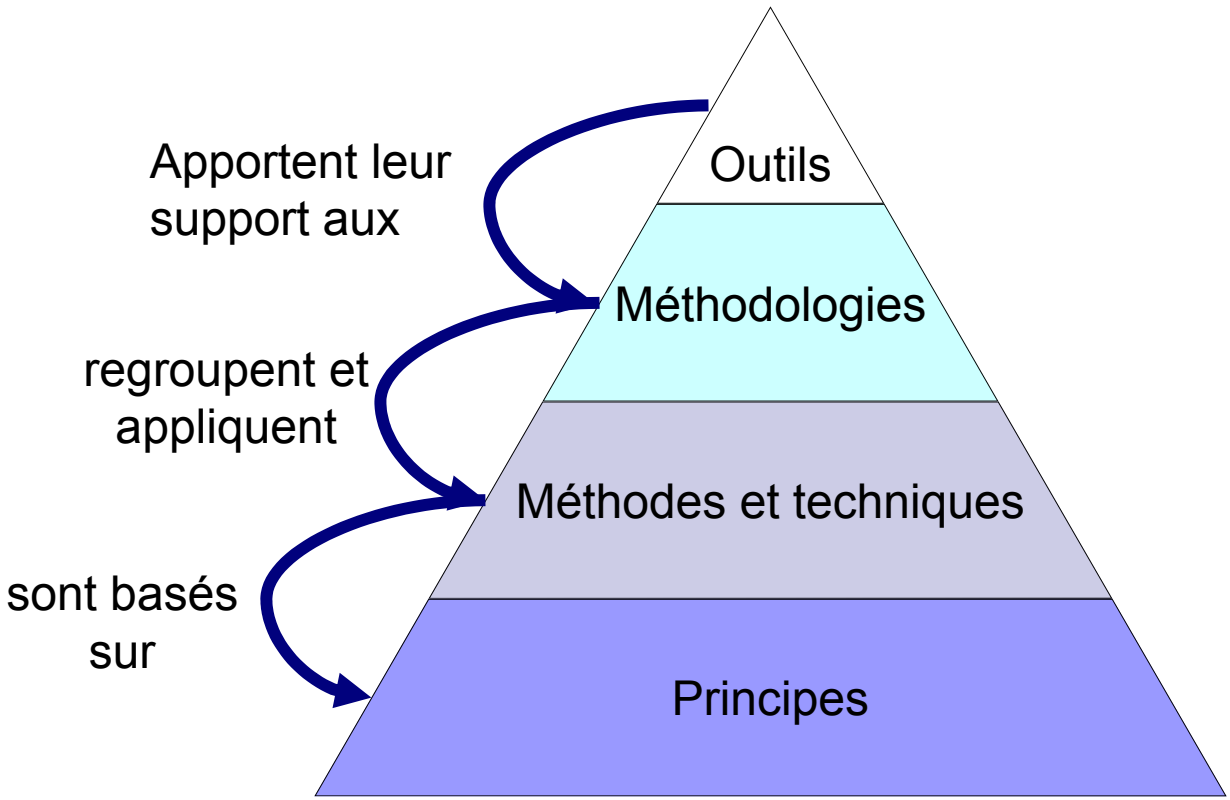


…En utilisant des méthodes, des techniques et des outils informatiques

Développer un logiciel c'est comme cuisiner.

- Processus = étapes de préparation d'un repas
- Paradigme = style de cuisine
- **Principes** = «Une sauce réussie est onctueuse.»
- **Techniques** = techniques culinaires pour faire les sauces, réussir les mayonnaises, etc.
- **Méthodes** = les recettes
- **Méthodologie** = livre de cuisine
- **Outils** = casseroles, bols, mixette, etc.

Relations entre principes, méthodes, techniques, méthodologies et outils



Certains problèmes du génie logiciel
Ou « quand la tarte aux pommes contient des pépins … »

- Erreur: Est commise par un humain, dans une spécification, dans un programme, en discutant avec le client, etc.
- Faute: état intermédiaire incorrect dans lequel un logiciel se trouve.
- Défaillance: situation symptomatique ou la manifestation externe d'une erreur.

  - Une défaillance est toujours engendrée par une faute.
  - Une faute est toujours engendrée par une erreur.
  - Mais une faute ne cause pas toujours une défaillance.
  - Une erreur ne cause pas toujours une faute.

Lacunes dans le développement: conséquences coûteuses…

- La sonde Mariner vers Vénus s'est perdue dans l'espace à cause d'une erreur dans un programme FORTRAN.
- En 1981, le premier lancement de la navette spatiale a été retardé de deux jours à cause d'un problème logiciel.
- L'explosion d'Ariane 5, le 4 juin 1996, qui a coûté un demi milliard de dollars (non assuré !), est due à une faute logicielle d'une composante dont le fonctionnement n'était pas indispensable durant le vol. (http://www.cnes.fr/espace_pro/communiques/cp96/rapport_501/rapport_501_1.html#description)
- Le bug de l'an 2000…

Les défis du génie logiciel

- Minimiser les **coûts** de développement tout en répondant aux exigences croissantes
- Réduire les **temps** de développement.
- Assurer la **qualité** des logiciels produits.
- Instituer l'usage des **nouvelles technologies** (méthodes et outils) du génie logiciel.

Le génie logiciel en crise?

Comment arriver à simultanément…

- Assurer la maintenance du nombre croissant de logiciels (vieillissants!!)?
- Conserver un rythme de production logicielle qui puisse réponde à la demande

Quelques mythes…

Vrai ou Faux ?

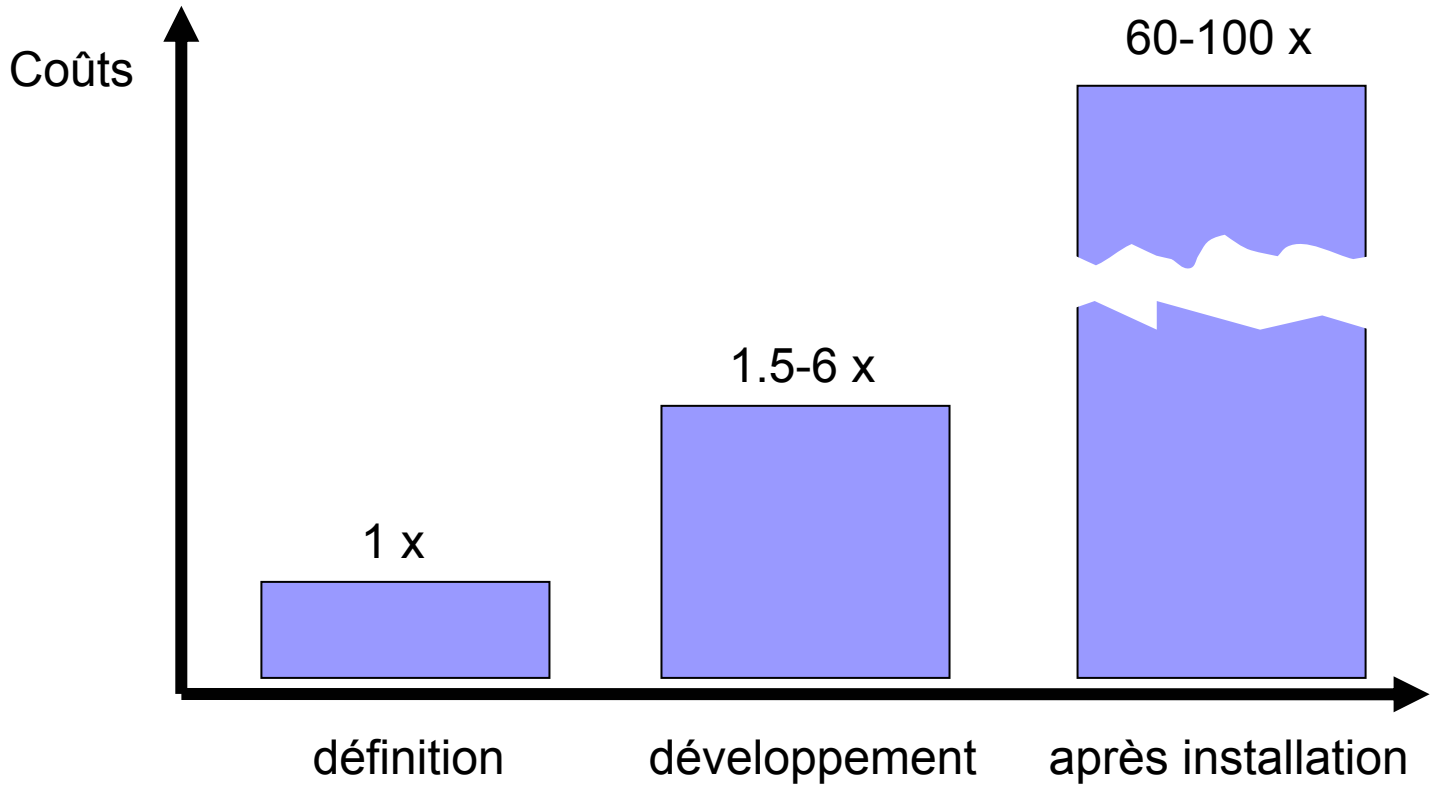
- Pour coordonner un projet, une méthode et un bon livre de directives à suivre est généralement suffisant pour guider les développeurs.
  - Faux.
- Un équipement informatique à la fine pointe de la technologie ne garantit en rien une production logicielle de qualité.
  - Vrai.
- En ajoutant des programmeurs on peut certainement accélérer la production logicielle.
  - Faux

Quelques mythes

Vrai ou faux?

- A partir d'une description générale des besoins, on peut immédiatement débuter le développement d'un logiciel.
  - Faux
- Si le logiciel est flexible, les modifications tardives peuvent être effectuées sans problème et sans surprise…
  - Faux

Coût d'une modification: surprise!



Quelques mythes…

Vrai ou faux?

- Le travail du développeur ne se limite pas à produire un programme qui fonctionne. En fait, 60% à 80% du travail devra être consacré à sa maintenance.
  - Vrai
- Le test est la seule méthode pour évaluer la qualité d'un logiciel.
  - Faux.
- La production de documents lors du développement logiciel n'est pas du temps perdu.
  - Vrai

1.1.2 Le logiciel: nature et qualités

- Le logiciel est un produit
  - Modifiable
  - Œuvre de création, plus que de manufacturation
  - Ensemble cohérent d'items ou d'objets qui forme une "configuration" incluant:
    - des programmes
    - des données
    - des documents

Applications logicielles

Critères de classification

- le type d'informations traité
  Ex. Image, son, texte …. Dans quel format?

- la disponibilité des informations et leur traitement dans le temps
  Quand les données arrivent-elles? Dans quel ordre? Quand doivent-elles être traitées? L'ordonnancement des opérations est-il prévisible?

Applications logicielles

Quelles sont leurs caractéristiques? Des exemples?

- Logiciels systèmes
- Logiciels temps réel
- Logiciels d'affaires
- Logiciels scientifiques et logiciels d'ingénierie

Applications logicielles

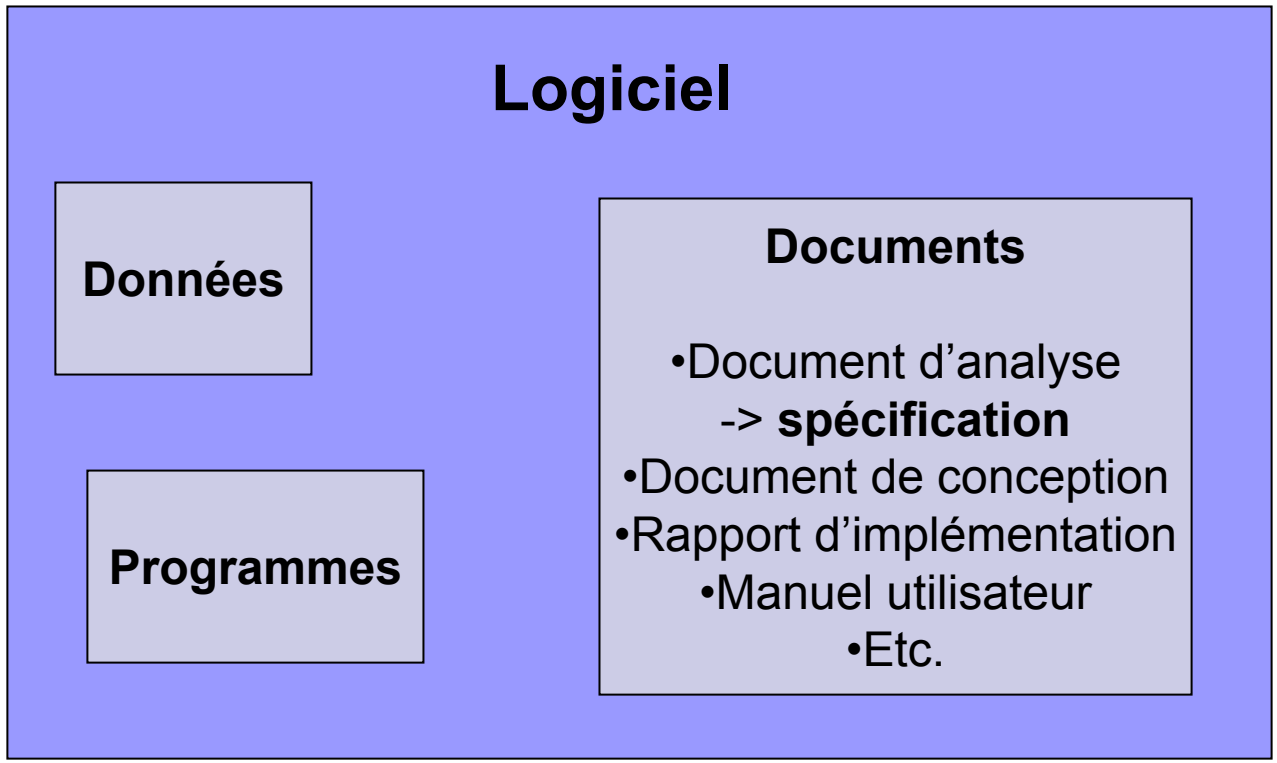
- Logiciels embarqués
- Logiciels pour ordinateur personnel
- Logiciels d'applications Web
- Logiciels d'intelligence artificielle

Qualités du logiciel

- Les qualités du logiciel:
  - **Externes**: observables par l'utilisateur.
  - **Internes**: d'intérêt pour le développeur du logiciel
    - Influencent directement les qualités externes.
    - Ex. Le logiciel doit être *vérifiable* (interne) pour pouvoir affirmer qu'il est *fiable* (externe).

Qualités du logiciel

Quelques préalables …



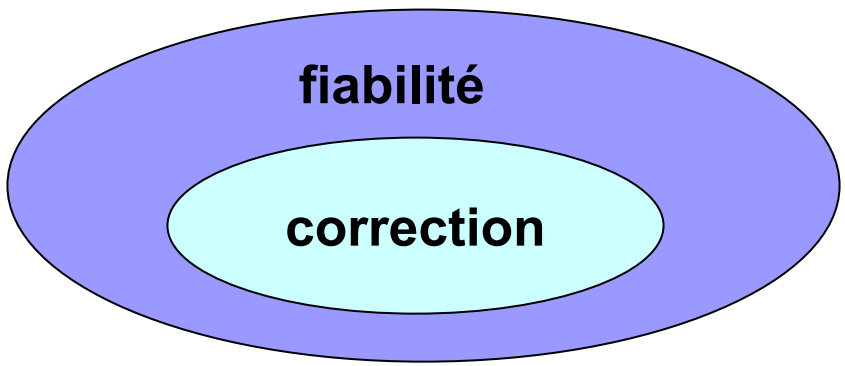
Qualités du logiciel

- Correction:
  - Qualité du logiciel qui se comporte exactement tel que décrit par sa spécification.
  - Qualité absolue.
  - *Hypothèse:* la spécification est sans ambiguïté et sans erreur -> décrit les propriétés attendues et désirables.

    Une spécification est dite complète si elle décrite toutes ces propriétés.

Qualités du logiciel

- Fiabilité:
  - Qualité du logiciel sur lequel on peut compter.
  - Aussi: probabilité que le logiciel se comporte tel qu'espéré durant un intervalle de temps donné.
  - Qualité relative (correction: qualité absolue).
  - Tout logiciel correct est fiable. Pourquoi?



Qualités du logiciel

- Robustesse:
  - Qualité du logiciel qui se comporte de façon convenable MÊME dans les circonstances non prévues par la spécification.
    - Comportement du logiciel possiblement non spécifié en cas de
      - mauvaises entrées
      - pannes matérielles
      - ajout de composants
      - etc.
  - Si la spécification décrit également les exigences concernant le traitement des cas exceptionnels:

    correction = robustesse…

Qualités du logiciel

- Performance:
  - Qualité du logiciel qui utilise les ressources de façon rentable.
    - peut changer en fonction de la technologie.
    - peut affecter la variabilité dimensionnelle du système.
      si $tcompil(p) = nbVar(p)^2$
  - Mesures de performance:
    - complexité algorithmique
    - mesure - monitoring
      (collecte de données, identif. bouchons)
    - analyse - construire un modèle et l'analyser
      (preuves math.)
    - simulation - construire un modèle et l'exécuter
      (stat. et prob.)

Qualités du logiciel

- Convivialité:
  - Qualité d'un logiciel que les utilisateurs trouvent agréable et facile à utiliser.
  - Types d'interface:
    - interface usager: *Type d'utilisateur? Expert ou novice?*
    - interface avec périphérique ou autre système: facile à configurer et à adapter avec autre système?

Qualités du logiciel

- Vérifiabilité:
  - Peut être vérifié facilement (par analyse formelle ou test).
  - Quelles propriétés peut-on vouloir vérifier?
    - correction
    - fiabilité
    - performance
    - etc.

Qualités du logiciel

- Facilité de maintenance
  - Qualité du logiciel dont l'entretien est facilement réalisable.
  - Types de maintenance:
    - Corrective(20% des coûts): Enlever les erreurs résiduelles ou introduites pendant la maintenance.
    - Adaptative (20%): Ajuster le logiciel suite à la modification de son environnement.
    - Perfective (+50%): Modifier le logiciel pour améliorer ses qualités (ajout de nlles fcts, performance, etc.)

Qualités du logiciel

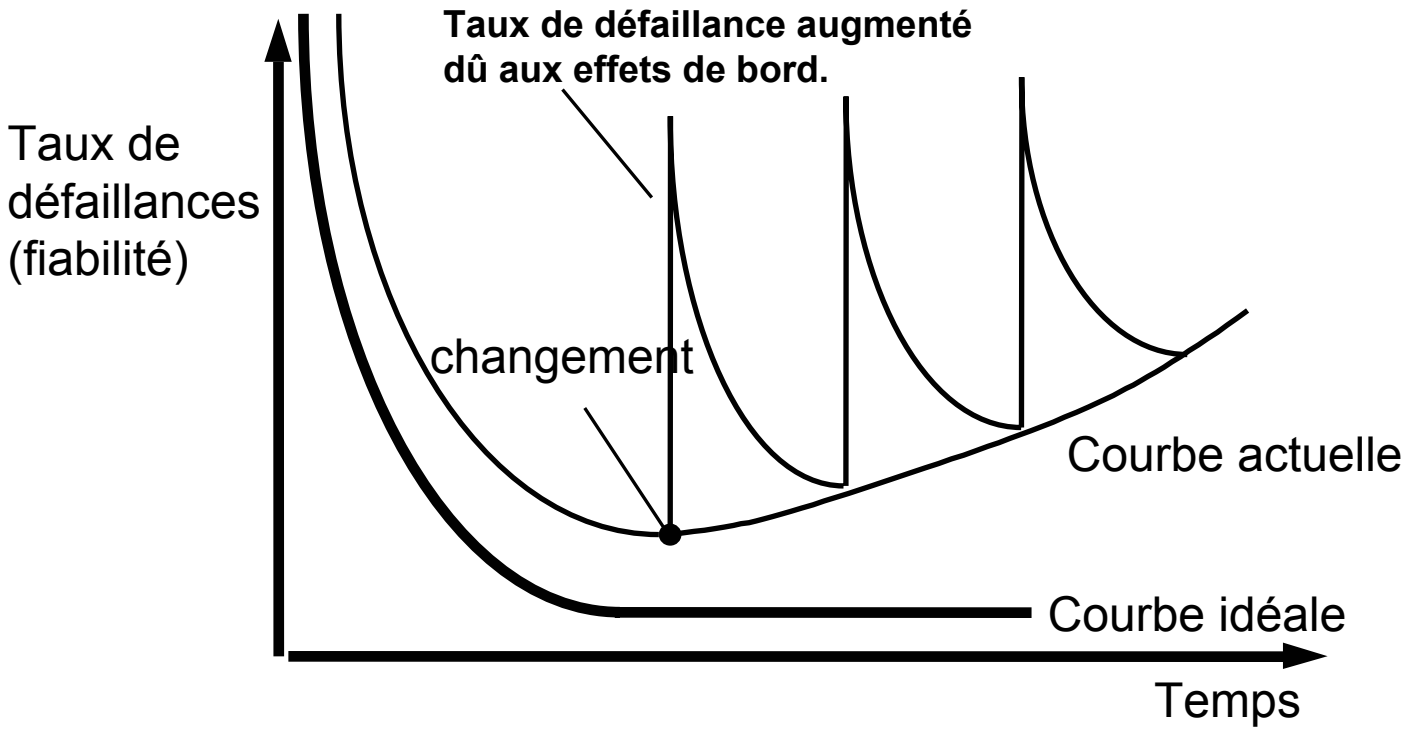
- Les deux (sous-) qualités à la base de la facilité de maintenance:
  - **Réparabilité:** Correction facile des erreurs.
  - **Évolubilité:** Réalisation facile des modifications formulées pour répondre à de nouvelles exigences.

Comment faire pour réaliser des programmes qu'on peut facilement déverminer et faire évoluer?

Qualités du logiciel

Comment la maintenance affecte-elle la fiabilité?

Changement (pour réparer ou faire évoluer)



Qualités du logiciel

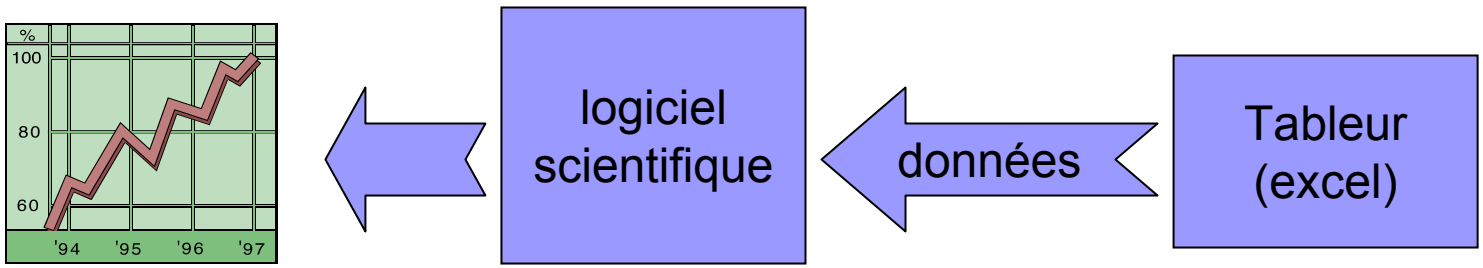
- Réutilisabilité :
  - Propriété de ce qui peut être utilisé à nouveau dans une autre application avec aucune ou peu de modifications.
  - Techniques favorisant la réutilisation:
    - tech. conception orientées objet (instanciation)
    - tech. conception à base de composants (code reuse)
    - design patterns (design reuse) :
    - modèle d'architecture permettant de résoudre un pbm.

Qualités du logiciel

- Portabilité:
  - Qualité d'un logiciel pouvant être utilisé sur des systèmes informatiques de types différents.
  - Solutions possibles pour développer des logiciels portables:
    - Faire l'hypothèse d'une configuration minimale (facilités et mémoire)
    - Utiliser technologies permettant de déterminer la config. et de s'adapter à elle.
    - Etc.

Qualités du logiciel

- Interopérabilité
  - Qualité du logiciel qui peut cohabiter et coopérer avec d'autres systèmes.



  - Système ouvert: Ensemble extensible d'applications indépendantes qui coopèrent pour fonctionner comme un système intégré.
    - Système d'interface ouvert permet d'ajouter de nouvelles fonctionnalités
  - CORBA: architecture ouverte indépendante standardisée.

Qualités du logiciel

- Le G.L. doit assurer la qualité
  - logiciel
  - processus de développement.
    - Productivité
      - diminuer les coûts
      - réutilisation (compromis)
      - Utilisation d'outils et environnements
    - Ponctualité
      - livrer logiciel d' « actualité »
      - développement incrémental: livraison précoce du logiciel
    - Visibilité (transparence)
      - documentation

Qualité du logiciel

Points de vue

- La qualité est perçue différemment selon
  - Le point de vue de celui qui l'observe: développeur, utilisateur, client, etc.
  - le domaine (contexte): logiciel critique? Editeur de texte?

Les différents points vue de la qualité selon Garvin

- **Vue transcendantale:** on sait reconnaître la qualité mais on ne peut la définir.
- **Vue "utilisateur":** Évalue si le logiciel répond à l'usage qu'on désire en faire.
- **Vue "fabriquant" (manufacturing):** Évalue si le logiciel est conforme à sa spécification.
- **Vue "produit":** Évalue si le logiciel offre les caractéristiques attendues pour le type de produit.
- **Vue basée sur la valeur:** Évalue ce que le client est prêt à payer pour le logiciel.

Qualités du logiciel

Exigences de qualité dans différents domaines

- Systèmes d'information et de commerce électronique
  - Gestion d'informations: création/destruction, consultation et mise à jour de données.
  - Information = un des produits les plus importants de nos jours!
  - Ex: systèmes bancaires, catalogue informatisé (biblio)
  - Qualités délicates:
    - Intégrité des données [c/f/r, verif]
    - Sécurité [c/f/r, vérif]
    - Disponibilité des données. [f/r, perf.]
    - Performance des transactions. [perf]
    - Convivialité.

Qualités du logiciel

- Systèmes temps réel
  - Doit répondre aux événements dans un laps de temps limité.
    - Temps de réponse est une question de correction et non de performance.
  - Ex.: système de monitoring de la température, système de guidage d'une sonde
  - Ordonnanceur (scheduler) d'actions.
    - Algorithme de type « priorité » ou « deadline ».
  - Qualités délicates
    - Correction, fiabilité, robustesse
    - Interopérabilité, convivialité

Qualités du logiciel

- Systèmes distribués
  - Ordinateurs indépendants connectés par un réseau de communication.
  - Caractéristiques
    - Distribution: données ou contrôle?
    - Tolérance aux fautes: partitionnement toléré?
  - Qualités délicates
    - fiabilité, robustesse
    - performance
    - interopérabilité

Qualités du logiciel

- Systèmes embarqués
  - Système intégré à un dispositif, une machine ou un autre système pour le piloter.
  - Exemples: système de freinage automobile, tableau de commande d'un micro-ondes, etc.
  - Caractéristique
    - Interface avec autre système (non humain).
  - Qualités délicates
    - Interopérabilité
    - réutilisation

Qualités du logiciel

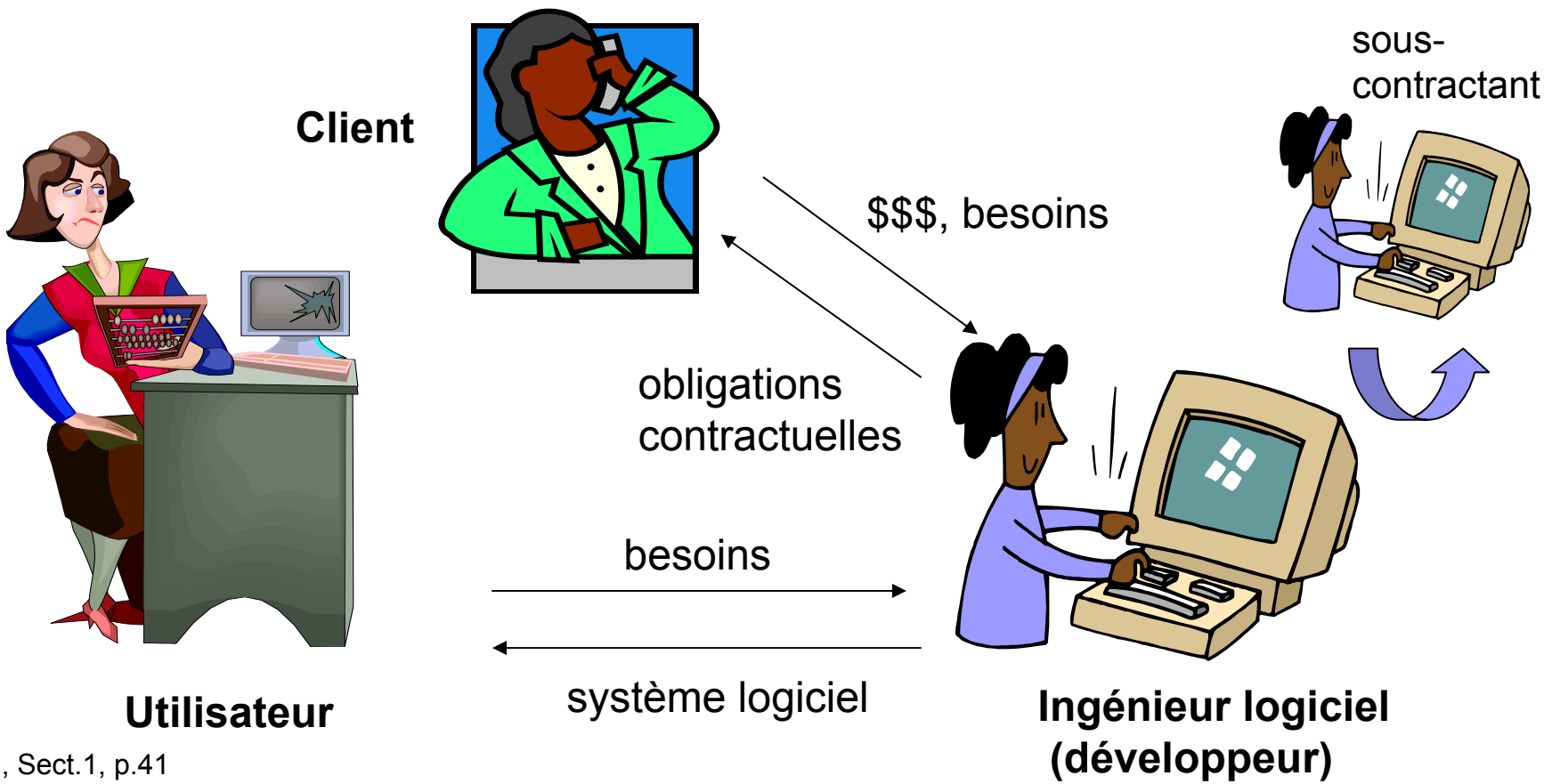
Comment mesure-t-on la qualité?

Ex. On évalue la fiabilité d'un pont en mesurant la pression qu'il peut supporter.

1. Définir avec précision la qualité à évaluer.
2. Etablir une métrique i.e. formule qui permet d'évaluer une qualité à partir de différents paramètres mesurables.

1.1.3 Les acteurs du génie logiciel

- Élément clef: communication
- Une personne peut assumer plusieurs rôles. Plusieurs personnes peuvent assumer un même rôle.



1.1.4 Une approche système (vue de l'extérieur…)

Pour résoudre le bon problème:

- Dans quel contexte le système s'inscrit-il?
- Quels éléments sont inclus ou exclus du système?

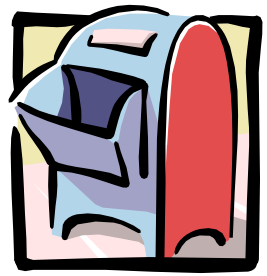
Description d'un système:

- les activités et objets qui le compose
- les relations entre ceux-ci
- ses frontières

Exemple – Déf. système

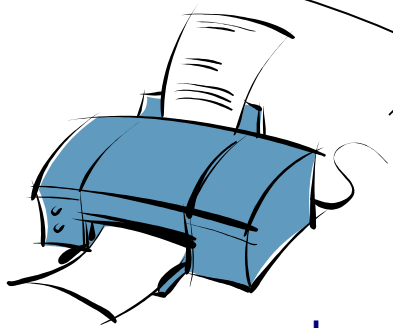
Ordinateur

Bureau du courrier



Information sur les paies
- …
- …

Validation de la date



Chèques de paie

Imprimer

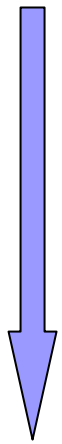
Calculer

Production des chèques de paie

1.1.5 Une approche d'ingénierie (vue de l'intérieur…)

Construire un logiciel c'est un peu comme construire une maison…

- Analyse des besoins et spécification
- Conception architecturale
- Conception détaillée
- Implémentation
- Tests unitaires
- Tests d'intégration
- Tests système
- Livraison du système
- Maintenance



Un processus de développement linéaire (dit « en cascade ») est-il adéquat? Pourquoi?

1.1.6 Évolution du génie logiciel

- Facteurs ayant modifié la pratique du génie logiciel:
  - Temps d'accès au marché de + en + critique
  - Changement de l'économie informatique:
    - Coût du matériel diminue
    - Coûts de développement et maintenance augmentent
  - Plus grande puissance de calcul
  - Réseaux locaux et grande distance étendus
  - Adoption des technologies orientées objets
  - Conception d'interface graphiques utilisateurs
  - Problèmes relatifs au processus de développement en cascade

1.1.7 Principes du génie logiciel

- Rigueur et formalité
  - Rigueur:
    - Logiciel doit offrir une solution précise.
    - Développement sérieux, soigné, non approximatif.
  - Formalité
    - summum de la rigueur
    - descriptions et validations s'appuient sur des lois mathématiques.

Pas besoin d'être tjrs formel. Mais les techniques formelles peuvent être très utiles, par exemple, pour la vérification.

Principes du génie logiciel

- Séparation des questions
  - Consiste à considérer séparément différents aspects d'un problème pour en maîtriser la complexité.
  - Peut prendre différentes formes:
    - séparation dans le temps (cycle de vie)
    - séparation des qualités à optimiser (ex. correction avant performance)
    - séparation des vues d'un système (vue des données, vue du contrôle)
    - séparation du système en parties (cf. modularité)

Principes du génie logiciel

- Modularité
  (séparation des parties logiques)
  - Principe qui consiste à décomposer un système en sous-systèmes plus simples (pour maîtriser complexité).
  - Permet de considérer séparément
    - contenu du module
    - relations entre modules
  - Caractéristiques d'une bonne conception modulaire:
    - Regroupement des aspects logiques semblables
    - Indépendance des modules

Principes du génie logiciel

- Abstraction (séparation des aspects)
  - Consiste à mettre les détails de côté pour ne considérer que les aspects jugés importants d'un système.
  - Il existe différents niveaux d'abstraction:

    Ex.: circuit électronique

    Niveau 1: Equation mathématique

    Niveau 2: Circuit logique des composants

    Niveau 3: Plan physique des composants réels au sein du circuit intégré.

Principes du génie logiciel

- Anticipation du changement
  - Consiste à prévoir les changements de façon à faciliter la gestion de ces évolutions inévitables.
  - Nature des changements:
    - perfectifs (nouveaux besoins)
    - correctifs
    - adaptatifs
  - Nature des problèmes:
    - localisation des erreurs (cf. modularité, couplage)
    - gestion des versions multiples du système

Principes du génie logiciel

- Généralité
  - Consiste à résoudre un problème plus général que le problème spécifique qui est adressé.
  - Solution pourra être réutilisée.
  - Ex. Au lieu de résoudre le problème de fusion de liste sans éléments identiques, on pourrait résoudre…

Principes du génie logiciel

- Construction incrémentale (raffinement)
  - Consiste à construire une solution étape par étape en s'approchant de plus en plus du but.
  - Chaque nouveau résultat est construit en étendant le précédent.
  - Ex.: réaliser les fonctions essentielles d'un système puis ajouter aspects secondaires.
  - Possibilité de produire des prototypes intermédiaires.