

IFT159

Analyse et programmation

Introduction

Gabriel Girard

Département d'informatique



2009



Analyse et programmation

1/46

Introduction

- 1 Historique
- 2 Environnement matériel
 - Mémoire centrale
 - UCT
 - Unités d'entrées et de sorties
- 3 Environnement logiciel
- 4 Développement de logiciels
 - Analyse
 - Conception
 - Implantation et essais
 - Prévisions des coûts
- 5 Éthique et professionnalisme



Analyse et programmation

2/46

Historique

- Début dans les années 40 ;
- Ordinateur = outil inanimé ;
- Exécute les instructions données ;
- Programme = suite d'instructions ;
- Programme écrit dans le langage de l'ordinateur.



Analyse et programmation

4/46

Historique

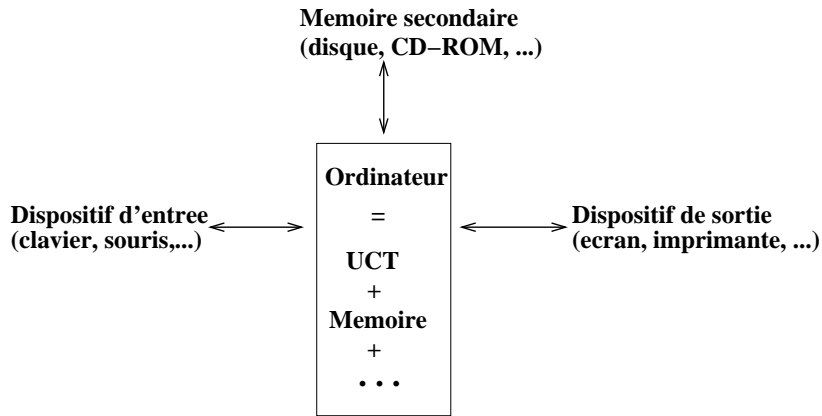
- Babbage ;
- ABC ;
- ENIAC ;
- Von Neumann ;
- Évolution technologique + catégories.



Analyse et programmation

5/46

Environnement matériel



Ordinateur et ses périphériques

Mémoire centrale

- Séquence ordonnée de cellules (0 à N-1) ;
- N est la taille exprimée en Meg ou Gig ;
- Numéro de la cellule = adresse ;
- Cellule = 1 octet = 8 bits ;
- valeur occupe une ou plusieurs cellules ;

UCT

- Unité de contrôle ;
- Unités fonctionnelles (UAL, ...) ;
- Registres (généraux, PC, PS) ;
- ift249, gei201, gei301 ;

Unités d'entrées et de sorties

- Clavier, souris, crayon, guichet, ...
- Écran, imprimante, ...

Mémoire secondaire

- Disque souple ;
- Disque rigide ;
- Carte de mémoire (flash, ...) ;
- Bande magnétique ;
- Disque compact (CD, DVD, Blu-ray) ;
- Concept de fichiers

11/46

Réseau

- Local (Ethernet, sans-fil) ;
- Modem ;
- Ift585 ;

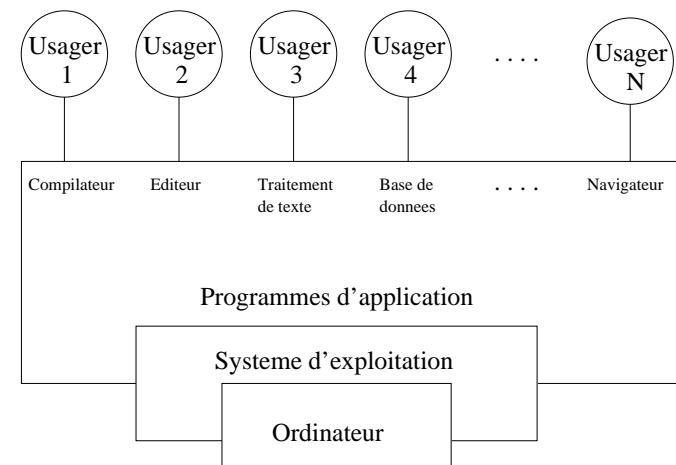
12/46

Environnement Logiciel

- Système d'exploitation ;
- Applications ;
- Outils de développement (langage, compilateurs, ...) ;

14/46

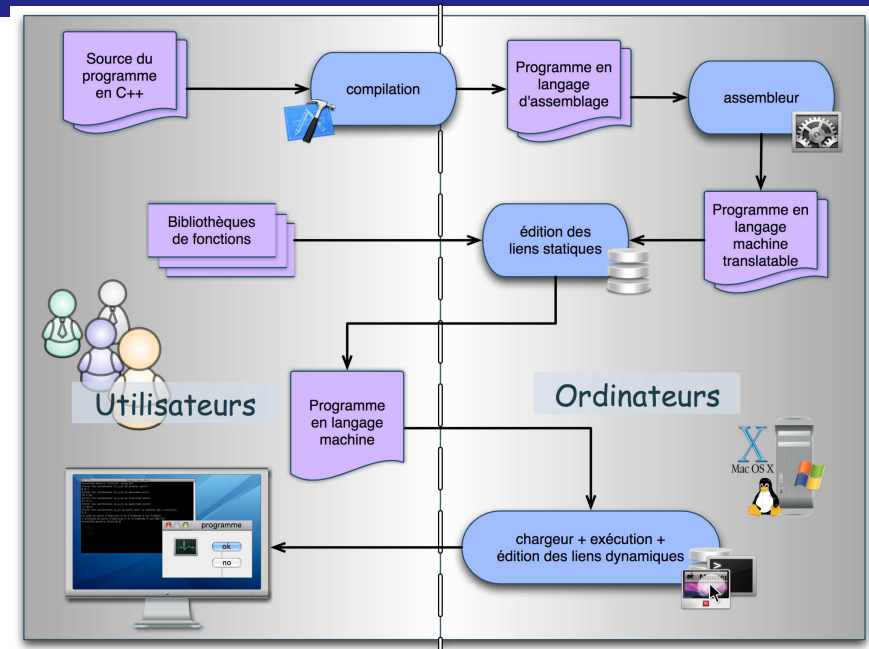
Environnement Logiciel



15/46

Environnement de développement

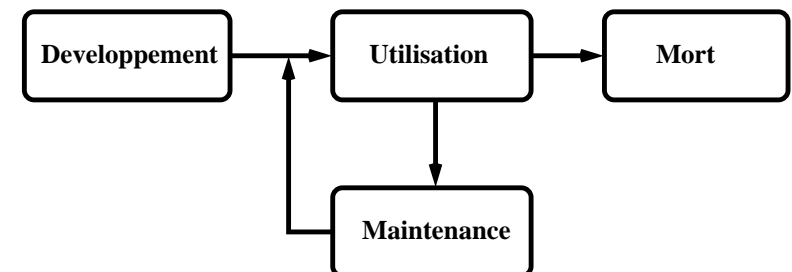
- Langage de programmation ;
- Compilateur ;
- Éditeurs, outil de mise au point, éditeur de liens, ... ;
- Environnement de développement intégré ;



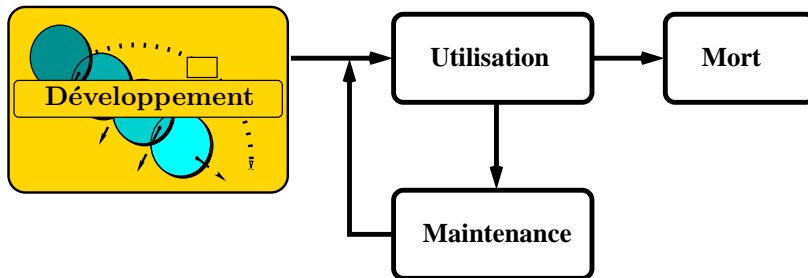
Développement = résolution de problèmes

- But ultime → programmes ;
- Étapes préliminaires :
 - Comprendre le problème (résolution de problèmes) ;
 - Analyser le problème ;
 - Concevoir une solution générale.

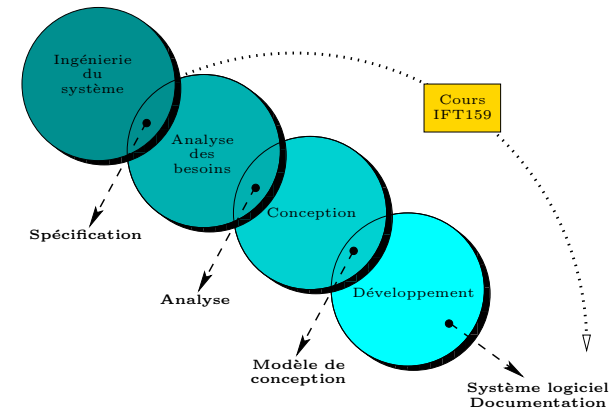
Cycle de vie d'un logiciel



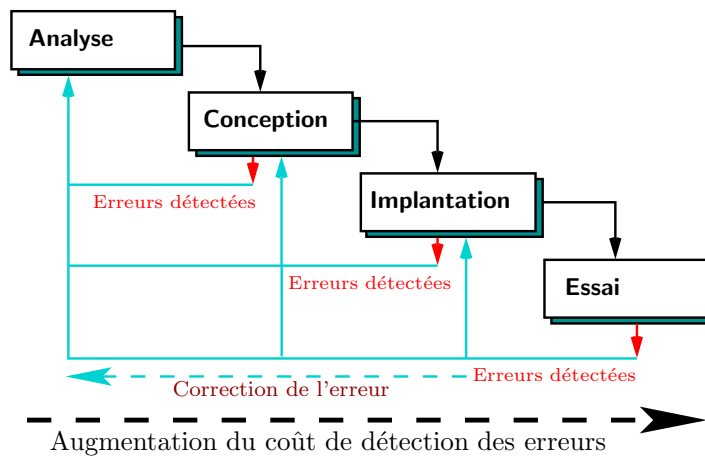
Cycle de vie d'un logiciel : Le développement



Phases de développement



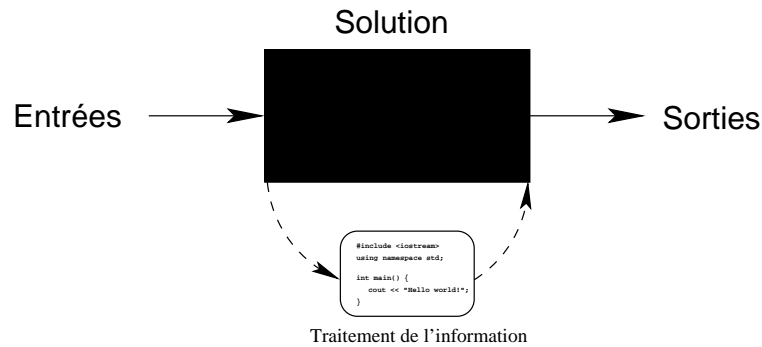
Modèle de la chute d'eau



Analyse

- Déterminer les attentes de l'utilisateur ;
- Comprendre et clarifier le problème ;
- Déterminer les tenants et aboutissants ;
- IMPORTANT : séparer l'essentiel du détail, c'est-à-dire procéder par abstraction ;
- On travaille sans se préoccuper du « comment » ;

Analyse — boîte noire



Analyse — méthodologie

- Comprendre le domaine du problème ;
- Définir les fonctions du logiciel ;
- Représenter le comportement du logiciel ;
- Hiérarchiser le modèle d'analyse contenant l'information, la fonction et le comportement du logiciel ;
- Aller de l'information essentielle vers les détails du système.

Questions pertinentes pour faire l'analyse

- 1 Quelles sont les données initiales ?
⇒ Entrées
- 2 Par quel biais le programme les obtient-elles et à quel ensemble appartiennent-elles ?
⇒ média et types
- 3 Lesquelles sont à considérer comme des constantes ?
⇒ constantes
- 4 Quels sont les résultats recherchés ?
⇒ Sorties
- 5 Que dois-en faire le programme et à quel ensemble appartiennent-ils ?
⇒ média et types
- 6 Quelles sont les relations entre les données initiales et les résultats recherchés ?
⇒ Relation entre les E/S, les formules

Objectifs de la conception

- Créer les « plans » du système
- Détailler son architecture (module/classe/composante etc.)
- Détailler le comportement de chaque élément
- Respecter les besoins explicités par l'analyse

Méthodologie de conception

- On part avec le problème principal
- Décomposition du problème en sous-problèmes (modules)
- On fait l'analyse et la conception des sous-problèmes (modules) afin de les diviser en sous-problèmes.
- À chaque étape on ajoute des détails
- Niveau de détails suffisants → production de la liste d'étapes à suivre (algorithme).

Questions pertinentes pour faire la conception

- 1 Quelles sont les étapes à réaliser (les calculs à faire) pour obtenir les résultats escomptés à partir des données du problème ?
⇒ Algorithme

Conception — Outil graphique

Nous verrons trois outils :

- Diagramme structurel
- Diagramme structurel avec flots de données
- Diagrammes UML

Implémentation

- Choix du langage en fonction des besoins et des ressources.
- Chaque module est codé dans le langage choisi.
- La documentation et la lisibilité du code sont primordiaux.
- ⇒ La maintenance coûte très cher.

Essais — l'importance

- Coûts des erreurs annuellement ?

\$ 59,5 milliards

\$ 59,5 milliards en 2002 juste aux États-Unis

- Combien aurait-on pu économiser avec de meilleurs tests ?

\$22,2 milliards

Le tier de ces coûts (\$22,2 milliards) aurait pu être évité par une meilleure infra-structure de tests.

Essais — l'importance – Exemples

- Ariane 5 : coût \equiv \$500 millions
- Therac-25 - machine de radiothérapie : coût \equiv 3 morts, 3 blessés graves
- Échec du missile Patriot : coût \equiv 28 morts, 100 blessés
- Écrasement du "Mars Climate Orbiter" : coût \equiv \$ 125 millions
- Bug de l'an 2000 : coût \equiv \$500 milliards
- ... Il y a plein d'autres exemples...

Essais — l'importance

- La phase de tests est la partie la plus coûteuse de l'ingénierie logicielle.
- Détection tardive des erreurs \Rightarrow hausse du coût exponentielle.
- Établir une stratégie de tests (jeu de tests) au plus tôt.
- Des tests hasardeux \Rightarrow erreurs critiques.
- Il faut se poser deux questions pour chaque étape :
 - 1 Construisons nous le bon produit ? (validation)
 - 2 Construisons nous le produit correctement ? (vérification)

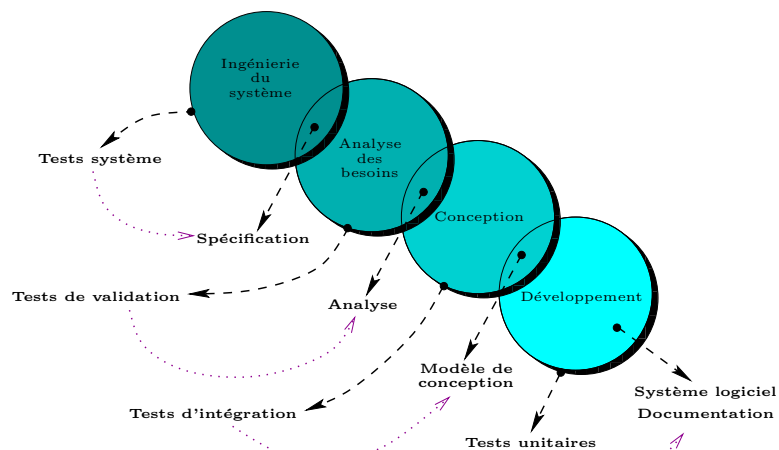
Attention

Les tests découvrent des erreurs mais ne prouvent rien.

Essais — les catégories

- Tests unitaire \Rightarrow valide l'implémentation.
- Tests d'intégration \Rightarrow valide le modèle de conception.
- Tests systèmes \Rightarrow valide le modèle d'analyse.
- Tests d'acceptance \Rightarrow valide la spécification du système.

Cycle de tests



Prévisions des coûts

Définition de l'effort.

L'**effort** est le produit du nombre de personnes par le nombre d'heures disponible par personne.

Dans le cours, il faut prévoir

- Analyse entre 5 % et 15 % des efforts
- Conception entre 20 % et 50 % des efforts
- Implémentation entre 10 % et 30 % des efforts
- Tests entre 20 % et 50 % des efforts

Planification des efforts

Format du document

- Nombres d'étudiantEs dans l'équipe
- Effort disponible (en *heure.personne*)
- Pour chaque activité décrire
 - Activité
 - Effort alloué (en *heure.personne*)
 - Effort consommé (en *heure.personne*)
 - Justification de la différence
- Effort consommé total (en *heure.personne*)

Document disponible sur internet.

Étude de cas Consommation automobile

Entre deux pleins d'essence un automobiliste désire connaître sa consommation d'essence.

Confidentialité et mauvais usage

- Les informaticiens peuvent avoir accès à des données sensibles.
→ Ne pas :
 - les accéder si cela n'est pas utile pour la tâche
 - les utiliser pour des gains personnels
 - les utiliser pour des activités illégales, non éthiques ou nuisibles à autrui
 - les modifier pour nos besoins personnels
- Doivent faire comme les médecins ou les avocats

Effraction informatique (computer hacking)

- Cette activité est illégale peu importe l'objectif visé
- Elle peut entraîner des poursuites judiciaires
- Créer/propager un virus peut être considéré comme une effraction informatique

Plagia ou piratage

- Ne pas utiliser les programmes des autres sans leur permission
- Ne pas modifier les programmes d'un autre et affirmer que c'est le nôtre
- Ne pas pirater de logiciels ou produits commerciaux

Mauvaise utilisation des ressources informatiques

- Les codes et droits d'accès sont des propriétés privées
- Ils vous sont attribués pour des besoins précis
- Ne pas les prêter, les louer ou les partager
- Ne pas les utiliser pour d'autres fins que celles prévues

En cas de doute....

On s'informe!!!