

## Objectifs du cours

- Définir le génie logiciel
- Montrer l'utilité du génie logiciel
- Recenser des méthodes et des outils utilisés en génie logiciel
- Acquérir la maîtrise de quelques techniques de génie logiciel

# Bibliographie

- Le génie logiciel et ses applications, lan Sommerville, 1988, ISBN 2-7296-0180-5
- Spécification et conception des systèmes. Une méthodologie, Jean-Paul Calvez, Masson, Paris, 1991, ISBN 2-225-82107-0
- Génie logiciel. Les méthodes SADT, SA, E-A, SA, Patrick Jaulent, 1990, ISBN 2-200-42002-1
- Génie Logiciel : principes, méthodes et techniques, Presses polytechniques et Universitaires Romandes, 1996, ISBN 2-88074-296-X
- Précis de génie logiciel, Marie-Claude Gaudel, Bruno Marre, Françoise Schlienger et Gilles Bernot,
   Masson, Paris, 1996, ISBN 2-225-85189-1
- Génie logiciel, Jacques Printz, Que sais-je?, 1995, ISBN 2-13-046894-2
- Le génie logiciel orienté objet, Ivar Jacobson, Addison-Wesley, 1993, ISBN 2-87908-042-8
- Conception et programmation orientées objets, Bertrand Meyer, Eyrolles 2000, ISBN 2-212-09111-7
- Bases de la programmation et du génie logiciel, Guy Pierra, Dunod, 1991, 2-04-020722-8
- Méthodologie de la programmation en C Bibliothèque standard API POSIX, Jean-Pierre Braquelaire, Masson, 1998, ISBN 2-225-83269-2
- Eléments de génie logiciel. Outils de développement en C, Jacques Chauché, 1990, 2-7298-9058-0
- Qualité du logiciel et système qualité L'industrialisation par la certification, Jean-Pierre Martin, 1992, ISBN 2-225-82801-6
- L'assurance qualité La nouvelle version de la norme ISO 9001 en pratique, Claude Jambart, Economica, Paris, 2001, ISBN 2-7178-4199-7
- Maîtriser la qualité Tout sur la certification et la qualité totale les nouvelles normes ISO 9001
   version décembre 2000, Daniel Boéri et Mélina Cuguen, Maxima, Paris, 2001, ISBN 2-84001-271-5

## Plan du cours

- Les problèmes du développement logiciel
- Le cycle de vie
- Méthodes et modèles
- La spécification
- La conception
- La programmation
- Le test

# Le génie logiciel

- Expression née en 1968 (conférence de l'OTAN)
- Définition : Méthodes, techniques et outils de conception et réalisation des logiciels de qualité industrielle
- Objectifs : Maîtrise des coûts, des délais, de la qualité
- Moyens: Etude des méthodes et techniques suivies pour le développement et la maintenance des logiciels → capitalisation du savoirfaire, modularité / réutilisabilité, automatisation partielle, standardisation et normalisation, mesures, modèles, assurance qualité
- Difficultés : Secret industriel qui nuit à la diffusion du savoir-faire (y compris en présence de problèmes), spécificités liées aux domaines d'application, impression de facilité, etc.

## Deux aspects complémentaires du génie logiciel

- Transformation d'un problème informel en une solution logicielle
- Maintenance du logiciel jusqu'à la fin de son cycle de vie (mise à jour, correction des erreurs résiduelles, évolutions, etc.)

#### → au-delà de la simple programmation

Un programmeur écrit un programme alors qu'un ingénieur GL écrit (conçoit, programme, documente, valide) un composant appelé à être inséré au sein d'un système

Produit logiciel = programmes + documents

# Types de logiciel

- Logiciel système
- Logiciel temps réel
- Logiciel de gestion
- Logiciel scientifique
- Logiciel combinatoire
- Logiciel critique

Logiciel / Progiciel



## Données économiques sur le coût des logiciels

- Volume, effort, délai, durée de vie
- Evolution de la demande
- Evolution des coûts
- Nature du risque logiciel

# Volume, effort, délai, durée de vie

#### Quelques coûts

Compilateur C	10 HA

<ul><li>Compilateur ADA</li></ul>	120-150 HA
-----------------------------------	------------

Système temps réel de la navette spatiale >1000 HA

2200 KLS

6 ans

Système d'exploitation (MVS, VMS, etc.)
2500-5000 HA

5000-15000 KLS

15-20 ans

#### L'évolution de la demande

Le logiciel est utilisé presque partout :

montre 2 K instructions

TV 100 K instructions

automobile, téléphone mobile 150 K instructions

centrale téléphonique1 M instructions

- système de combat du porte-avions Charles de Gaulle 8 M instructions
- Tout ce qui était analogique devient digital et est candidat à être traité par ordinateur : GED (Gestion électronique de documents) et multimédia, images de synthèse, etc.
- Les services demandés à l'informatique sont en augmentation constante

#### L'évolution de la demande

- L'explosion du logiciel continue :
  - En 1995, le développement de Microsoft Exchange Server a coûté 1000 HA pour 7 000 KLS,
     soit une productivité de 30 lignes par homme et par jour.
  - De 1965 à 1995, le volume de chaque logiciel a été multiplié par 100, alors que la productivité n'augmentait que d'un facteur 3.
- Le progrès exponentiel de l'informatique devrait se poursuivre encore longtemps

#### L'évolution des coûts

- Inversion complète des coûts matériel / logiciel
- La part des coûts logiciels de maintenance est de plus en plus grande
- Le coût humain devient le facteur de coût principal

## La nature du risque logiciel

- La sûreté de fonctionnement devient essentielle
  - Risques humains
    - commandes de vol des avions
    - contrôle aérien, ferroviaire, nucléaire
    - etc.
  - Risques économiques
    - crach boursier d'octobre 87
    - effondrement du réseau de télécommunication AT&T sur le côté est des Etats-Unis d'Amérique
  - Risques sociaux
    - le système Socrate de la SNCF
    - la confidentialité des informations
    - les virus et les "chevaux de Troie"

#### Le constat

- Présence de nombreuses erreurs
  - Cas du système de la navette spatiale
    - logiciel embarqué0.11 par KLS (1 toutes les 10000 lignes)
    - logiciel au sol0.40 par KLS (1 toutes les 2500 lignes)
  - Logiciel grand public  $\approx$  5-10 par KLS
- Evolutivité problématique
  - Cas des grands systèmes

## Des exemples de bogues

- 2 jours sans courant pour la station Mir, du 14 au 16 novembre 1997.
   Cause : plantage d'un ordinateur qui contrôlait l'orientation des panneaux solaires.
- Mission Vénus : passage à 5 000 000 km de la planète au lieu des 5 000 km prévus.
  Cause : remplacement d'une virgule par un point.
- Perte de satellites dans les années 70.
   Cause : +I au lieu de +1 dans une instruction du programme source FORTRAN (le langage admet sans le signaler des identificateurs non déclarés, leur valeur est alors aléatoire).
- Refus illégal de prestations sociales à des ayants droits.
   Cause : le cahier des charges n'avait pas prévu tous les cas.
- Inondation de la vallée du Colorado en 1983.
   Cause : mauvaise modélisation du temps d'ouverture du barrage.
- Echec du 1er lancement de la fusée Ariane V qui a explosé en vol.
  Cause : reprise du logiciel de plate-forme inertielle d'Ariane IV sans nouvelle validation. Le logiciel a considéré que l'inclinaison d'Ariane V (+ importante que celle d'Ariane IV car ses moteurs étaient plus puissants) n'était pas conforme au plan de tir et a provoqué l'ordre d'auto-destruction. Coût du programme d'étude d'Ariane V : 38 milliards de francs.

## Et le bogue de l'an 2000

La lutte contre le bogue de l'an 2000 a coûté, seulement pour la France, des milliards de dollars. Des dysfonctionnements ont quand même été constatés, par exemple :

- Défaillances des systèmes informatiques de détection des vents de surface, dans plusieurs aéroports américains.
- Données fournies par des satellites de renseignement du Pentagone perdues à jamais.
- Bébé danois enregistré comme ayant 100 ans.
- 48 incidents répertoriés en France par le ministère de l'économie.

Cause : Codage de la donnée "année" sur 2 caractères pour gagner de la place.

### La mauvaise maîtrise du développement de logiciel

- De nombreux projets informatiques n'ont jamais abouti :
  - système de réservation de places de United Airlines
  - système d'exploitation Rhapsody
- Ou n'ont pas réalisé les fonctionnalité attendues :
  - Windows 3
  - Windows 2000 / ME
- Ou ont été des catastrophes économiques :
  - système de réservation de places Socrate de la SNCF
  - système d'information de la Bibliothèque Nationale de France : dépassement de 40% du budget initial (280MF)
  - informatisation des caisses de sécurité sociale, carte Vitale

#### Les causes

- Erreurs humaines
  - activité logico-mathématique
  - généralisation / construction des abstractions
  - modélisation des systèmes
  - communication (comprendre et être compris)
  - traduction
- Complexité du problème
- Multiples versions
- Durée de vie sur plusieurs années
- Instabilité des spécifications

# La qualité logicielle

- Un système logiciel peut être observé selon ses qualités :
  - externes

détectables par les utilisateurs du produit et les clients

ex : validité, robustesse, performance, à coût optimal, extensibilité, réutilisabilité, compatibilité, efficacité, portabilité, vérifiabilité, intégrité, facilité d'utilisation, facilité d'apprentissage, ergonomie, ...

internes

perceptibles par des informaticiens (concepteurs, implémenteurs) déterminants pour l'obtention des qualités externes

ex : modularité, lisibilité, maintenabilité, ...

- Les facteurs qualité ne sont pas nécessairement compatibles 2 à 2
  - → compromis à trouver

## Le problème du génie logiciel

- L'objectif est de construire des logiciels
  - ergonomiques
  - fiables
  - évolutifs
  - économiques
  - satisfaisant les critères CQFD
     (Coût / Qualité / Fonctionnalités / Délais de réalisation)
- Le problème central est la présence d'erreurs et leur détection

#### Les solutions

#### Objectifs du développement logiciel

- Satisfaire les besoins du demandeur (client)
- Respecter les délais et les coûts
- Satisfaire des critères de qualité

Maîtrise des constituants de la réalisation Interprétation correcte du domaine du client Développement en équipe de plusieurs personnes (cas d'applications complexes)

à défaut d'assurer la qualité d'un logiciel, assurons celle des méthodes de développement et de maintenance

Utilisation d'une méthodologie de développement

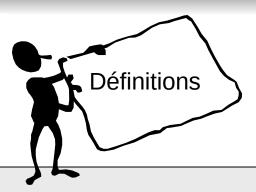
Mise en place d'une démarche qualité

## Conclusions

- Il est impossible de créer un logiciel sans défaut.
- La correction d'erreurs peut entraîner d'autres erreurs
- Le coût du développement d'un logiciel est très élevé. Il peut être estimé, très globalement, à :
   75 (développement) + 2250 (maintenance) € par instruction.
- L'effort de développement est proportionnel au nombre d'instructions
- La productivité obéit à une loi empirique, la loi des rendements décroissants :
   MH = 3,5 KISL<sup>1,2</sup>
   et n'augmente pas avec l'augmentation de l'investissement en hommes.
- Les moyens de sortir de cette crise du logiciel sont en partie connus
   méthodologie de développement + processus d'assurance qualité
- Les moyens mis en œuvre pour le développement d'un logiciel définissent son niveau de maturité. Plus ce niveau est élevé, moins il y a de bogues par ligne.



# Le développement d'un logiciel

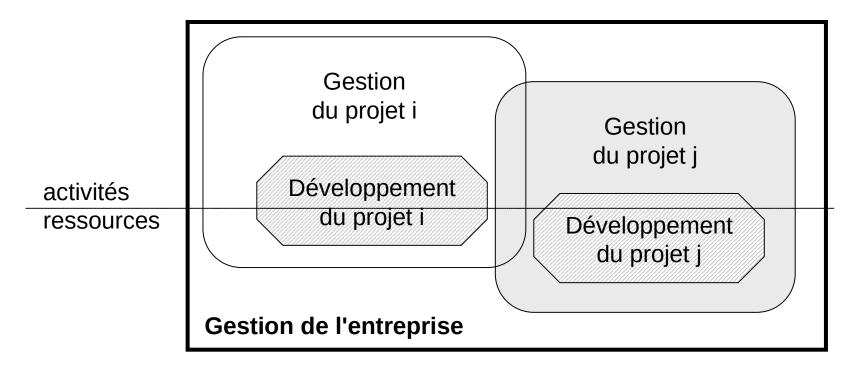


Le développement désigne l'ensemble des activités qui permettent de passer d'un cahier des charges au produit industriel répondant au besoin.

Le cahier des charges est un document décrivant les besoins du client.

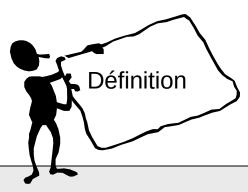
## Le contexte d'un développement

L'activité de développement doit être replacée dans le contexte de l'entreprise :



La gestion de projet nécessite de modéliser le processus de développement luimême → cycle de vie

# Le cycle de vie d'un logiciel



Le cycle de vie décompose le processus de développement selon une série d'activités couplées entre elles, partant du besoin initial pour aboutir au produit opérationnel.

# Les modèles de cycle de vie

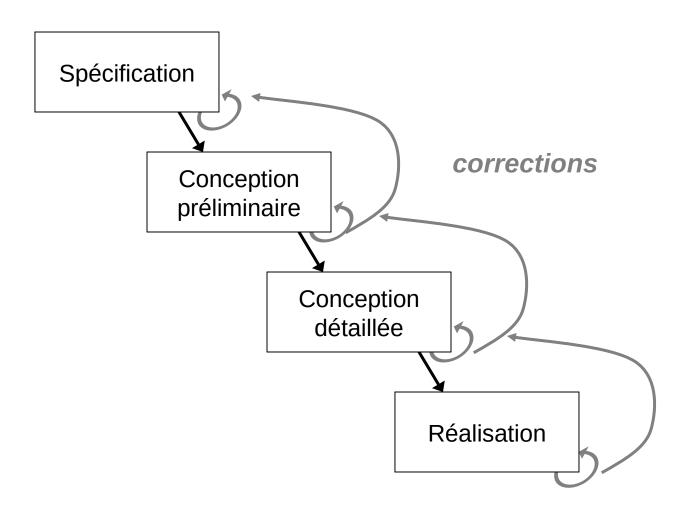
- Plusieurs modèles de cycle de vie ont été élaborés. Tous ont en commun au minimum les 5 phases essentielles de tout développement :

  - conception
  - réalisation
  - test
  - exploitation
- Ces modèles permettent de structurer le travail et facilitent le contrôle.

### V&V

- Validation externe
   Conformité des spécifications aux besoins des utilisateurs
- Vérification (validation interne)
   Conformité d'une réalisation logicielle ou d'une spécification détaillée à une spécification plus abstraite

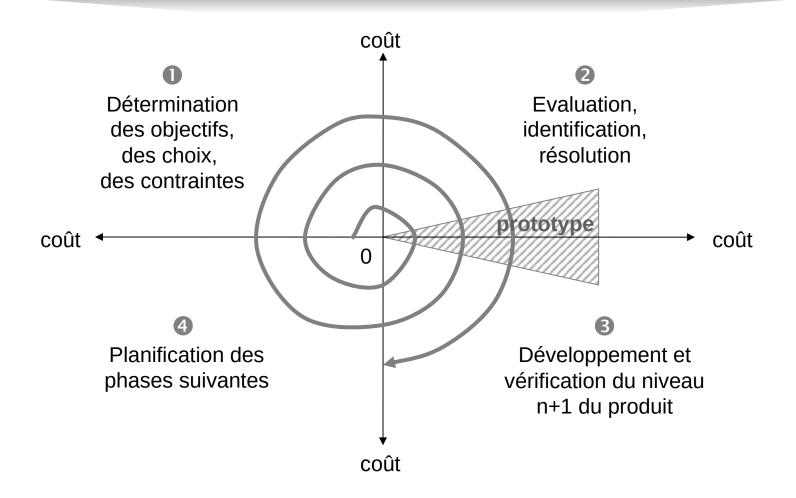
### Le modèle de la cascade



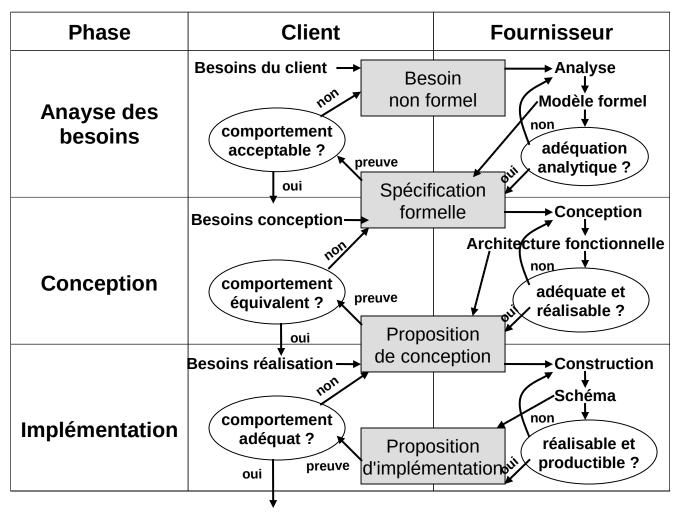
### Le modèle en V

**Besoins Produit** certification Cahier des charges Recette démarche globalement ascendante démarche globalement descendante validation **Spécifications Tests** Conception Tests d' vérification <u>intégration</u> préliminaire Tests des Conception détaillée composants Codage des composants

# Le modèle en spirale

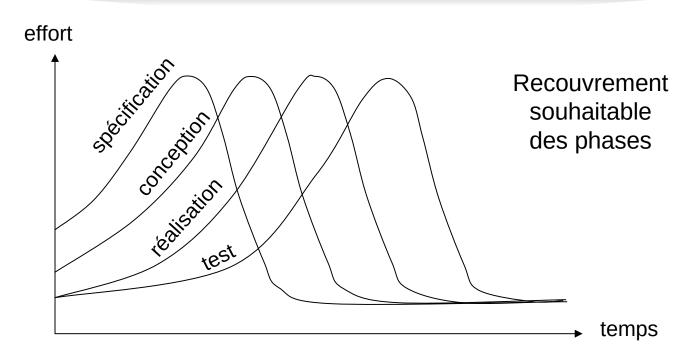


### Le modèle contractuel



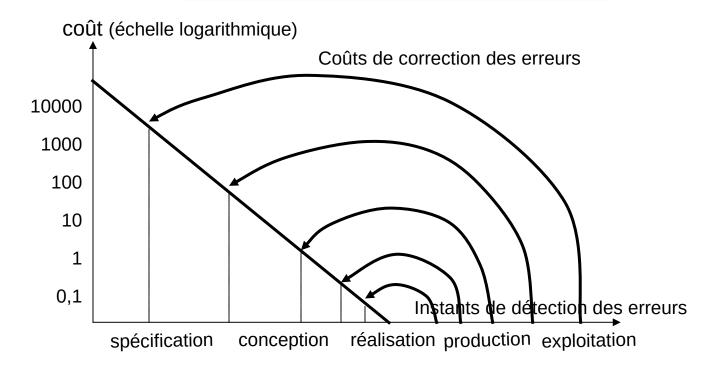
**Génie Logiciel** 

## Recouvrement des phases



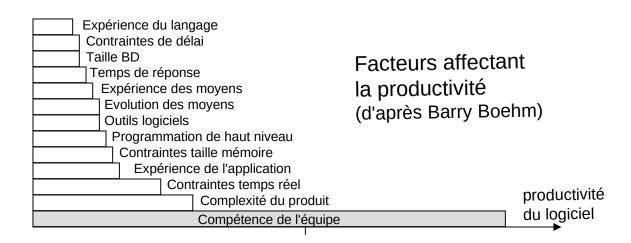
- Toutes les phases débutent presque simultanément, mais avec un effort qui varie en fonction de la progression.
- Les spécifications sont rarement achevées car des évolutions apparaissent côté demandeur

#### Coût de correction des erreurs



→ Plus les erreurs sont détectées tard dans le cycle de vie, plus elles sont difficiles à cerner et à corriger : le coût de correction d'une erreur est à chaque étape multiplié par 10.

# Facteurs de productivité



- → Faible contribution des facteurs d'expérience
- Le facteur essentiel est la compétence de l'équipe qui est intimement liée aux méthodes utilisées

# Répartition de l'effort entre les phases

La répartition de l'effort dépend de la nature du problème et de la stratégie de développement souhaitée.

Pour réduire au minimum possible le coût des erreurs et pour que la réalisation soit efficace, l'effort doit être porté sur les deux premières phases : spécification, puis conception.

Les activités de spécification et de conception peuvent être jugées comme improductives. La tendance naturelle est de penser qu'on est plus efficace et plus productif quand on est affairé sur son programme que quand on élabore au préalable des documents de spécification et de conception.