

Analyse et Conception Objet de Logiciels

Introduction



« Crise du logiciel »

Depuis les années 1970, les projets logiciels sont célèbres pour :

- leurs dépassements de budget et de délais ;
- leurs cahiers des charges non respectés.

Problèmes

- Elargissement des domaines d'application
- Croissance de la puissance des machines
- Complexité du logiciel (taille, algorithmes)
- Difficultés à établir et stabiliser les besoins des utilisateurs

« Crise du logiciel »

Selon une étude du gouvernement américain (1979)

Logiciel	Coût
Payés, jamais livrés	3.2 M \$
Livrés, jamais utilisés	2.0 M \$
Abandonnés ou recommencés	1.3 M \$
Utilisés après modification	0.2 M \$
Utilisés en l'état	0.1 M \$

Seulement 5 % du coût est « acceptable »

« Crise du logiciel »

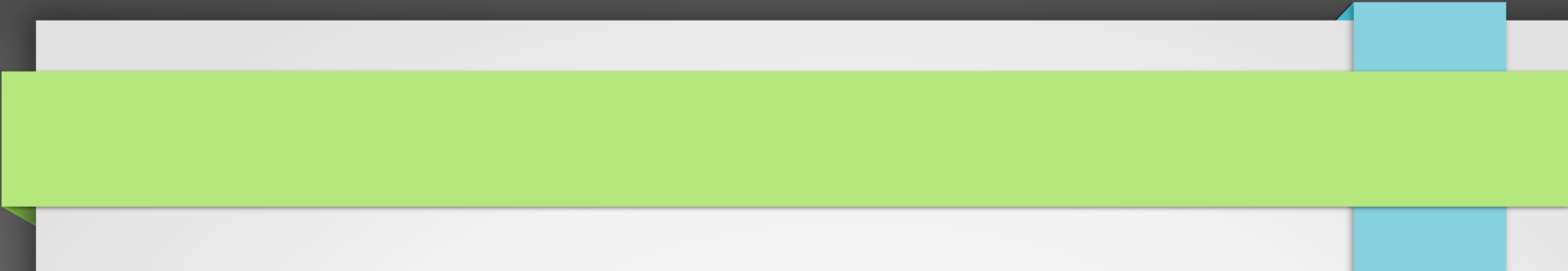
Selon une étude de 1995, aux Etats-Unis :

Logiciels	Coût
Abandonnés, ou jamais utilisés	31%
Coûts largement dépassés	53%
Réalisés suivant les plans	16%

Génie logiciel : « art de produire de bons logiciels, au meilleur rapport qualité/prix »

Etapes du cycle de vie du logiciel

- **Analyse et définition des besoins**
 - cahier des charges
 - étude de faisabilité
- **Analyse et conception**
 - spécification (description des fonctionnalités)
 - conception architecturale (structure)
 - conception détaillée (composants, algorithmes)
- **Mise en œuvre (programmation)**

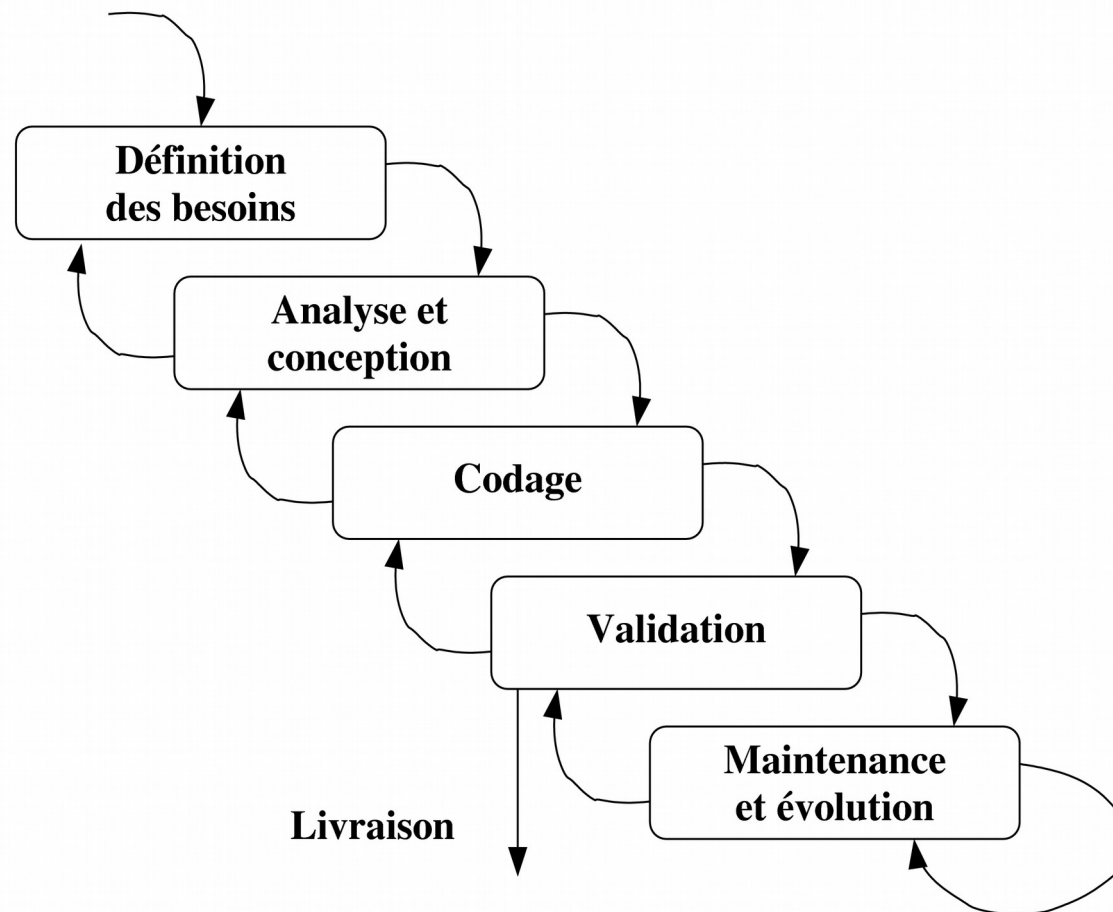
- 
- **Validation** : consiste à s'assurer que le logiciel est correct
 - analyse statique (typage, conventions ...)
 - preuve (peu utilisée)
 - revue de code (efficace)
 - tests (indispensables)
 - **Evolution et maintenance**
 - maintenance corrective (correction d'erreurs)
 - maintenance adaptative (portage du logiciel)
 - maintenance évolutive (évolution du logiciel)

Répartition de l'activité

- **Place de la maintenance**
 - développement initial : 20 %
 - maintenance : 80 %
- **Place de la mise en œuvre**
 - analyse et conception : 40 %
 - programmation : 20 %
 - validation : 40 %

Modèles du cycle de vie du logiciel

Modèle en cascade



Modèles du cycle de vie du logiciel

Problème du modèle en cascade

- les difficultés apparaissent très tard : lors de la validation, ou de la livraison au client

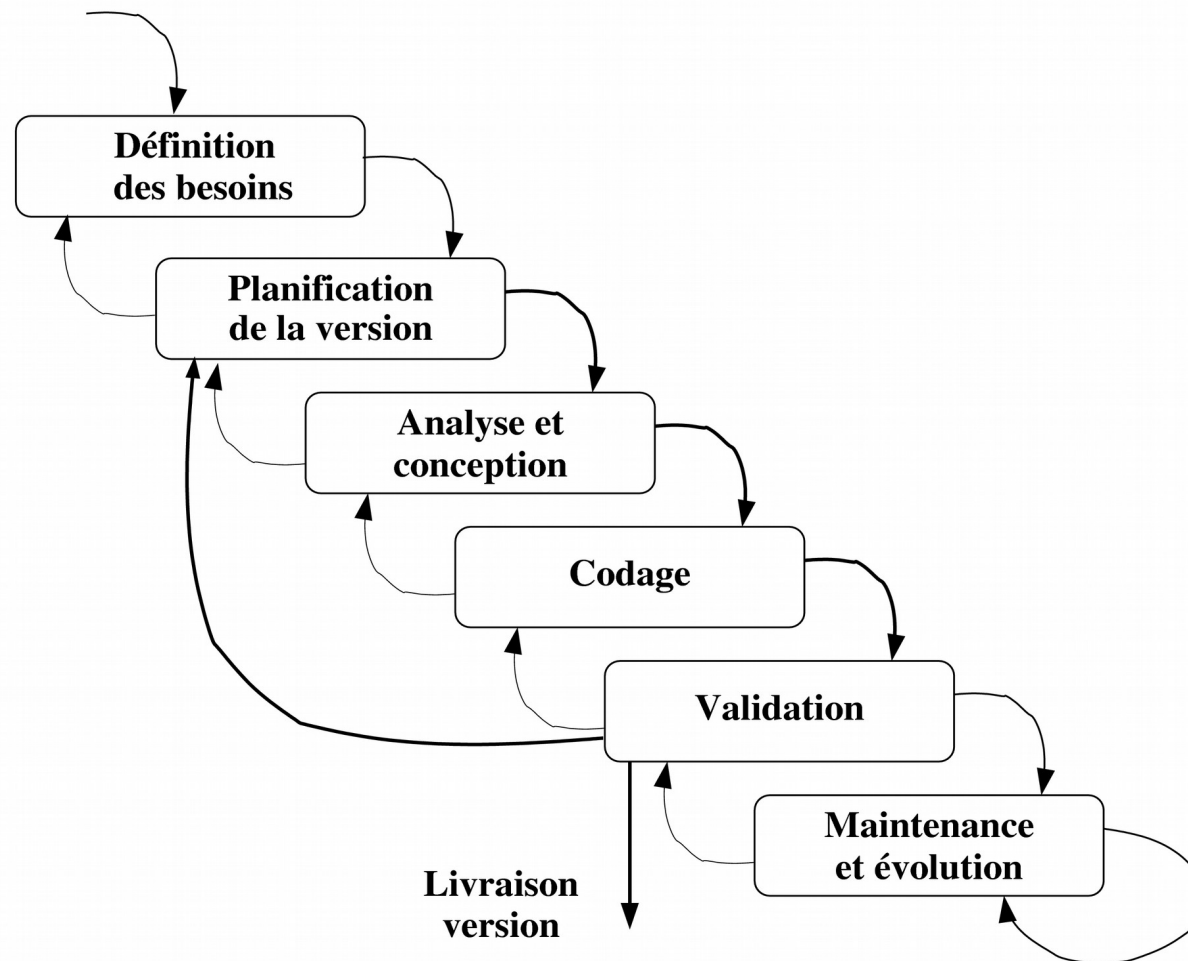
Développement agile

Modèles itératifs, qui procèdent par incréments successifs (petit nombre de fonctionnalités, analysées, conçues, programmées, validées et livrées au client)

- permet de révéler certains problèmes plus tôt
- permet d'avoir des retours du client régulièrement

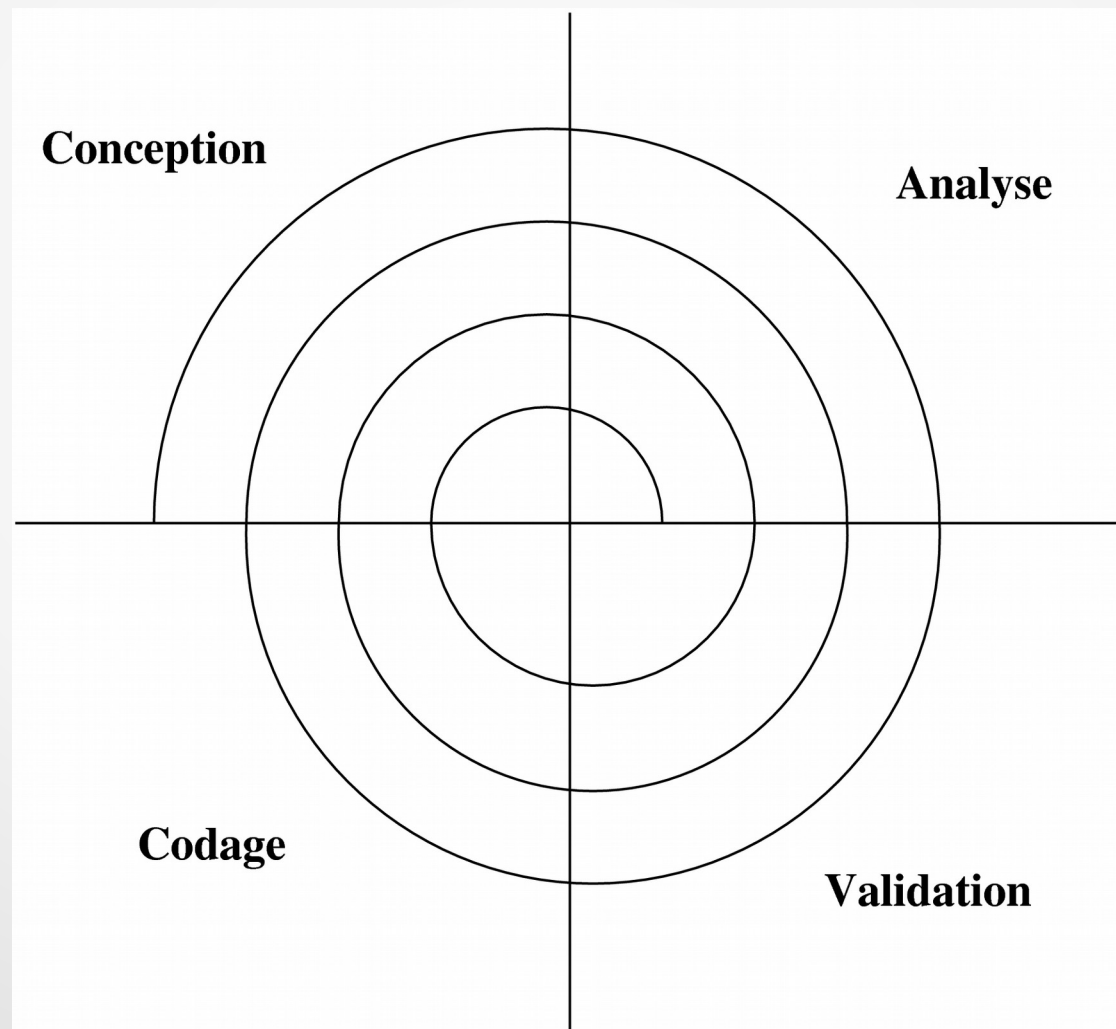
Modèles du cycle de vie du logiciel

Modèle incrémental



Modèles du cycle de vie du logiciel

Modèle en spirale



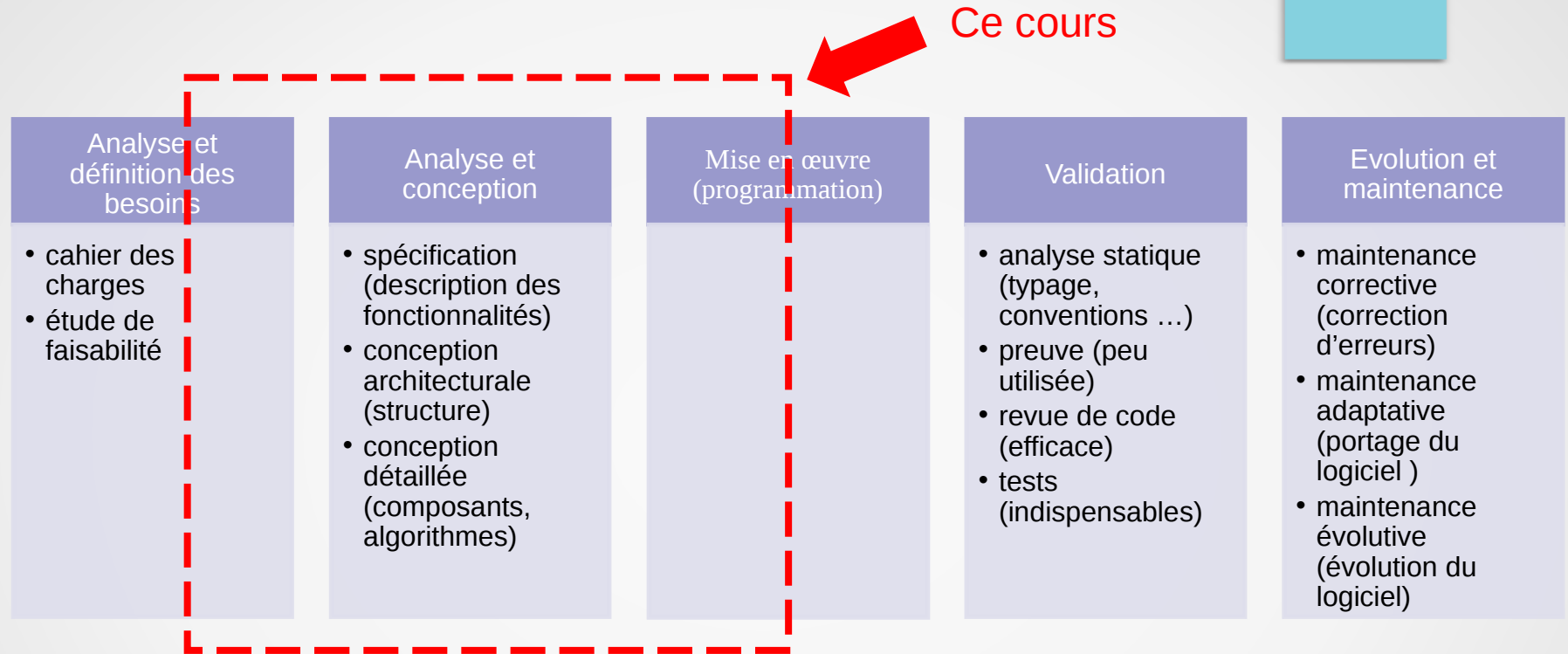
Méthodes de développement du logiciel

But : assister une ou plusieurs étapes du développement du logiciel

Processus de développement pour s'assurer que le logiciel soit :

- conforme aux attentes des utilisateurs ;
- livré dans les délais et au coût prévu.

Dans ce cours...



- Comment communiquer en amont du développement
- Comment analyser et concevoir des logiciels
- Comment développer des logiciels maintenables

UML : Unified Modeling Language

Langage UML : permet d'élaborer des « modèles objet », à l'aide de différents diagrammes

Modèle : abstraction de la réalité, c'est-à-dire une vue simplifiée, mais pertinente par rapport à ce qu'on veut réaliser

Aspects d'une modélisation :

- fonctionnel (fonctionnalités du logiciel)
- statique : objets et relations entre ces objets
- dynamique : états et comportements

Objectifs du cours

Maîtrise des principaux diagrammes UML

Utiliser ces diagrammes pour l'analyse et la conception de logiciels

Utiliser des outils pour concevoir et manipuler des diagrammes UML (ArgoUML, Umbrello, Dia...)

Réalisation d'un TP (équipes de 3 ou 4 étudiants)

Filière ISI : TP commun avec le cours de Construction d'Applications Web

Notation : Examen 3/4, TP 1/4

Plan du cours

Introduction

UML

Analyse

Conception

Patrons de conception