

# Génie Logiciel 1

## Cours 2

# PDL: Modèle des Phases du développement de logiciel

- ❑ La décomposition du projet en étapes facilite le contrôle du développement en définissant pour chacun des objectifs à atteindre planifiables et mesurables.
- ❑ Sert de base pour la conduite de projet
- ❑ Constitue une feuille de route pour la gestion d'un projet

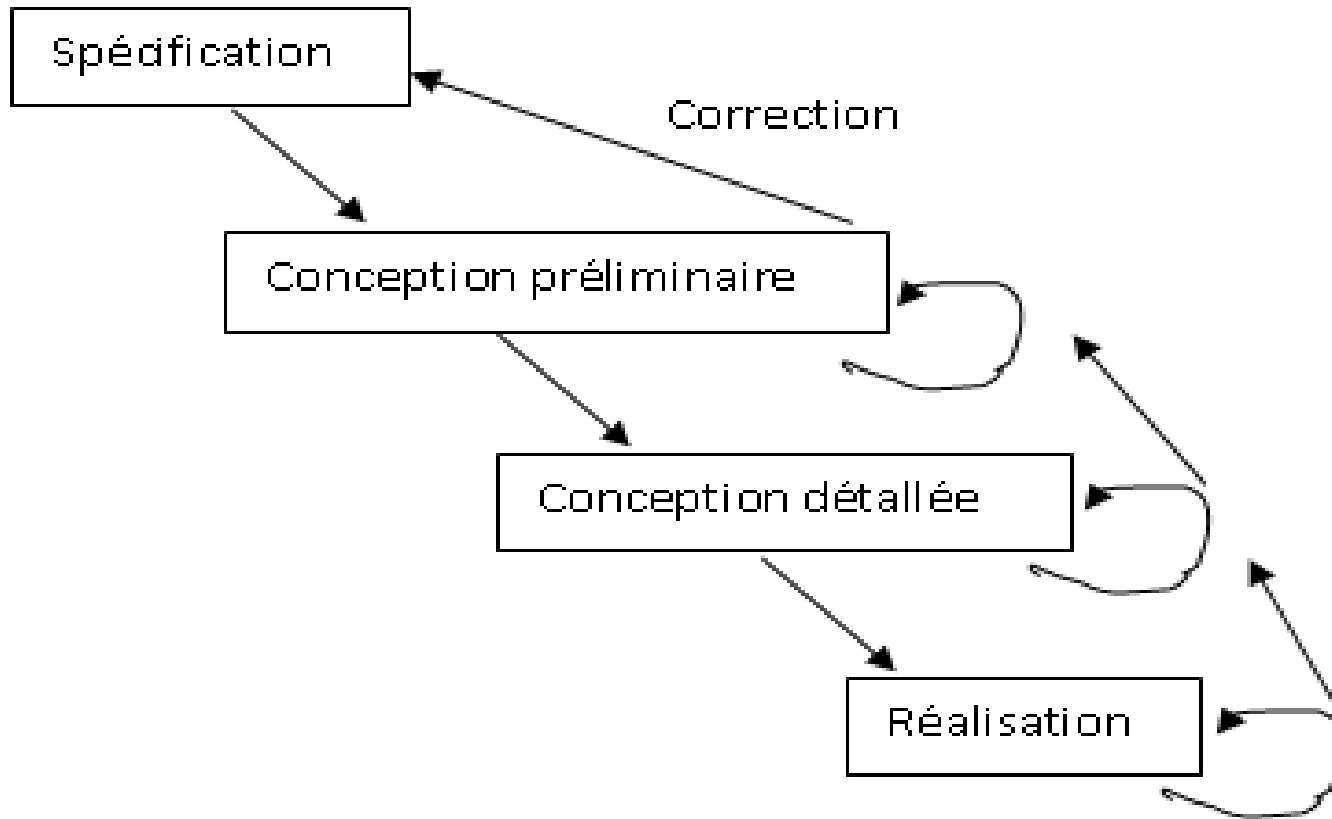
# PDL: Modèle des Phases du développement de logiciel

- ❑ Il est très rare d'appliquer un processus comme une unique séquence des activités précédentes.
- ❑ En général, un logiciel complet est le fruit de plusieurs **itérations**
- ❑ Plusieurs **modèles** ont été élaborés, ont tous en commun les phases essentielles de tout développement. Ainsi avec quelques variations en fonction des auteurs, on distingue:

# PDL: Modèle en Cascade

- ❑ premier modèle défini par [BOEHM 76]
- ❑ Décrit le cycle de vie comme une succession d'étapes conduisant à trouver des niveaux de descriptions du problème jusqu'à la réalisation, en partant de la définition jusqu'à l'exploitation et la maintenance.

# PDL: Modèle en Cascade



**Modèle Waterfall**

# PDL: Modèle en Cascade

- ❑ Chaque phase doit se terminer pour commencer la suivante.
- ❑ Des **documents** sont produits pour concrétiser la réalisation de chaque phase.
- ❑ Ce modèle montre qu'un développement ne peut pas se faire exclusivement selon une démarche purement descendante.

# PDL: Modèle en Cascade

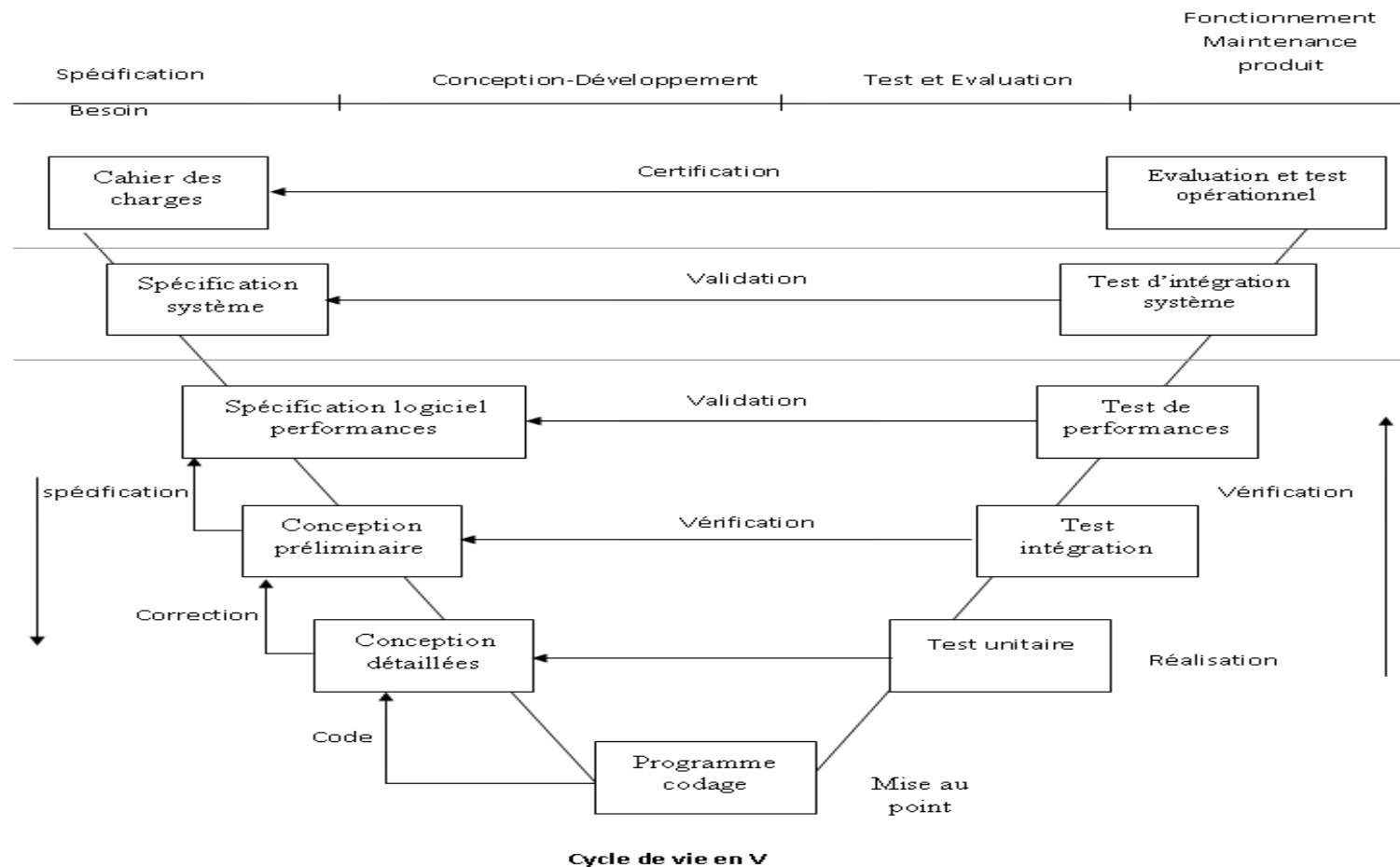
- Chaque étape est liée à une étape suivante pour représenter l'enchaînement, et à l'étape précédente pour représenter les corrections par retour en arrière.
- A chaque étape est associée une phase de vérification pour rassurer de la conformité de la solution retenue aux spécifications en entrée de l'étape.
- Un défaut de conformité implique de reprendre l'étape ou de revoir le résultat de l'étape précédente.
- Ce modèle montre qu'un développement ne peut pas se faire exclusivement selon une démarche purement descendante.
- Modèle limité, prend très partiellement en compte le caractère itératif du développement.

# PDL: le cycle en V

- ❑ Considère en supplément les phases de vérification et d'évaluation du projet pour chaque stade de sa réalisation.
- ❑ Démarche spécification-conception est globalement descendante.
- ❑ Phase de réalisation-test est globalement ascendante



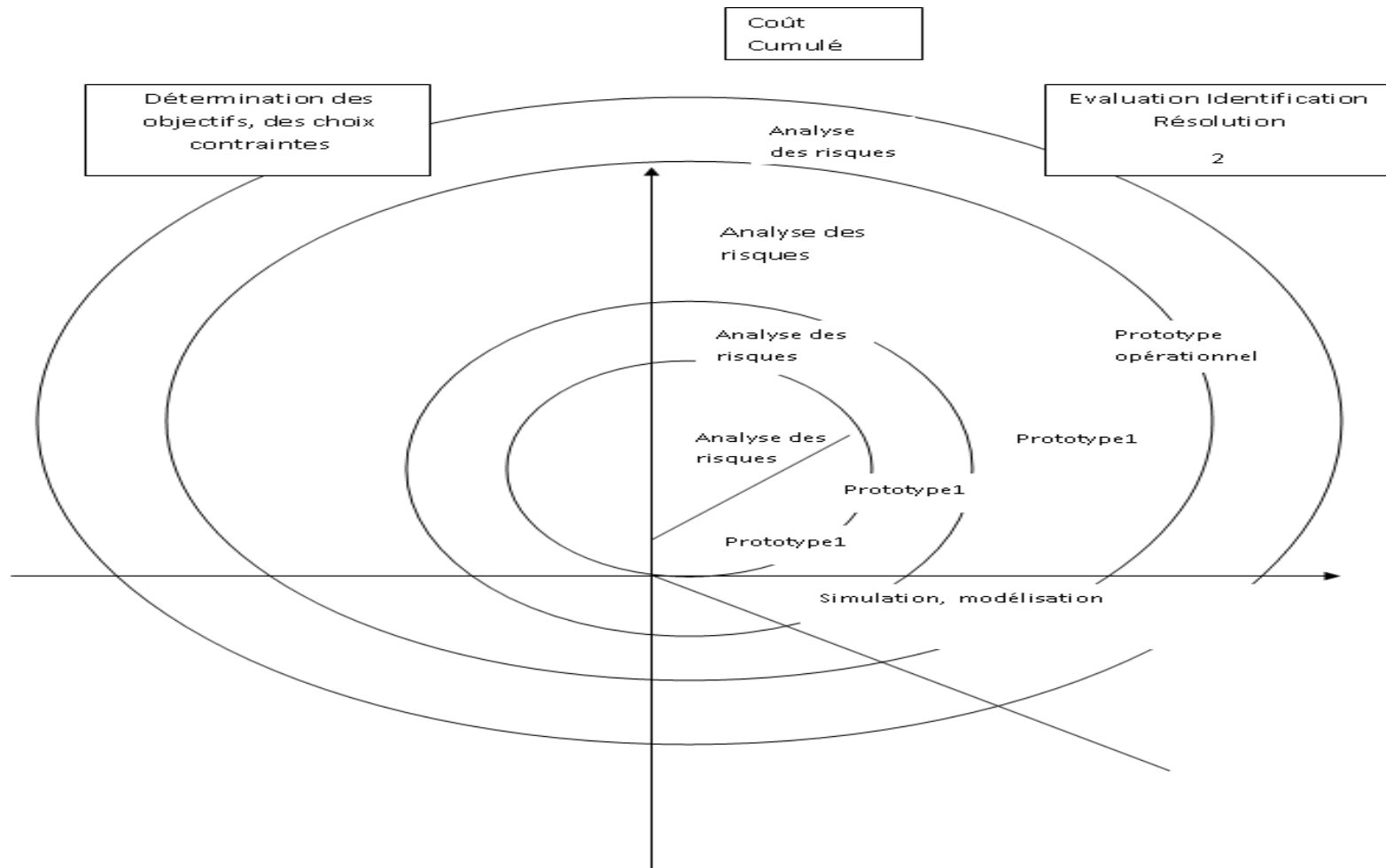
# PDL: le cycle en V



# PDL: le cycle en V

- ❑ L'axe horizontal représente les phases du projets et la durée de chacune.
- ❑ L'axe vertical représente le niveau d'abstraction pour l'application
- ❑ Les phases de spécification et de conception conduisent à définir des niveaux de description de plus en plus détaillées.
- ❑ Pour une application logicielle, le codage est la forme la plus détaillée.
- ❑ Les phases d'intégration, de test et de vérification, permettent d'évoluer la conformité de la réalisation pour chaque niveau de la conception en commençant par les parties les plus élémentaires puis en remontant progressivement jusqu'au produit complet.
- ❑ En correspondance de chaque phase de conception est associe une phase de test et d'évaluation de la conformité.
- ❑ Les deux démarches descendantes et ascendantes sont complémentaires.

# PDL: Modèle Spirale



# PDL: Modèle Spirale

- ❑ le modèle spirale défini par BOEHM 86 et WILLIAMS 88
- ❑ Modèle plus général que les précédents et peut les inclure
- ❑ Met l'accent sur une activité particulière, l'analyse des risques : chaque cycle de la spirale se déroule en quatre phases représentées par des quadrants

# PDL: Modèle Spirale

- ❑ La distance de tout point de la courbe au centre représente le coût cumulé qui a conduit à un tel stade de développement
- ❑ Le quadrant 03 correspond à un développement ou à une portion de développement classique, et un des modèles précède (de Waterfall ou V) peut s'appliquer : Son choix peut faire partie des alternatives à évaluer.
- ❑ L'originalité de ce « Super » modèle est d'encadrer le développement proprement dit par des phases consacrées à la détermination des objectifs et à l'analyse de risque.

# Modèle Spirale: La démarche

- Identifier les risques, leur affecter une priorité,
- Développer une série de prototypes pour identifier les risques **en commençant par le plus grand risque**
- utiliser un modèle en V ou en cascade pour implémenter chaque cycle
- si un cycle concernant un risque a été achevé avec succès,
- évaluer le résultat du cycle et planifier le cycle suivant
- si un risque n'a pu être résolu, terminer le projet immédiatement

# Références (Base)

## ❑ F. BROOKS

The Mythical Man-Month, Addison-Wesley, 1982

## ❑ B. BOEHM

Software Engineering Economics, Prentice-Hall,  
1981

## ❑ IAN SOMMERVILLE

*Software Engineering*, Pearson, 9th edition, 2010.