

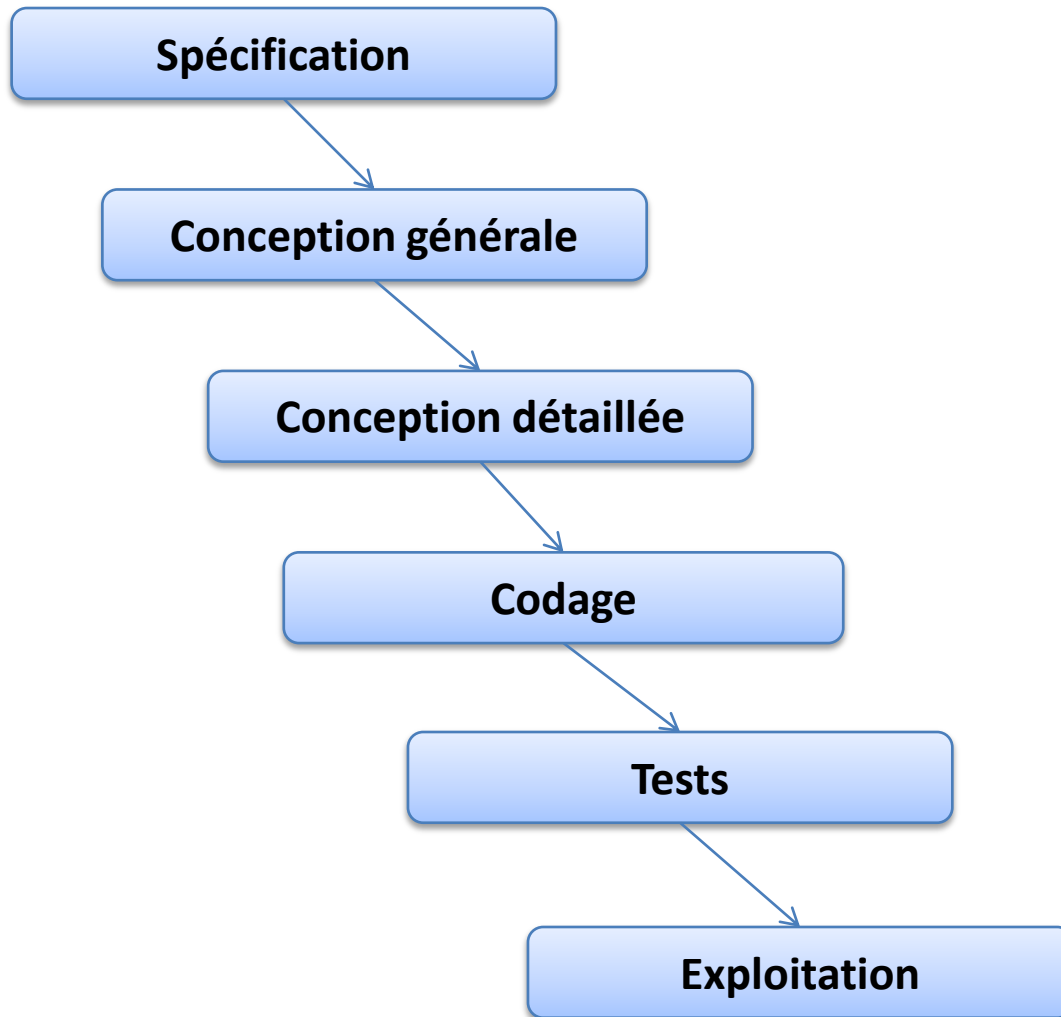
Les modèles de cycle de vie de logiciel

- Le modèle en cascade
- Modèle en V
- Développement incrémental (prototypage)
- Le modèle en spirale

Modèle en Cascade (Modèle linéaire)

- Décrit le cycle de vie comme une **succession d'étapes** de la **descriptions** du **problème jusqu'à la réalisation**.
- Chaque phase doit se **terminer** pour **commencer** la **suivante**.
- 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 à une phase de vérification pour rassurer de la **conformité** de la **solution** retenue **aux spécifications** en **entrée** de **l'étape**.

Modèle en Cascade (Winston Walker Royce)



Modèle en Cascade

Les étapes

Spécification : définition précise

- Des objets manipulés
- Des tâches à effectuer sur ces objets
- Des contraintes de performance

Conception générale

- **Architecture** du **système**
- **Principales structures** de **données**
- **Décomposition** du système en **modules**

Conception détaillée

- Raffinement des éléments précédents jusqu'à l'obtention d'une forme permettant d'écrire immédiatement les programmes.

Modèle en Cascade

Les étapes

Codage

- Écriture des textes des programmes

Tests

- On **exécute** le **système** avec des cas de tests issus de la spécification de **données réelles** du système futur .

Tests unitaires

- Les composants sont testés individuellement

Tests d'intégration

- Test du système global
- Test avec des données clients pour vérifier que le système répond aux exigences du client

Exploitation

- Mise en place du système dans son environnement opérationnel

Modèle en Cascade

Vérification et Validation

Vérification

- Le système est **conforme** à la **spécification**

Validation

- Le système répond aux **exigences** du client

Modèle en Cascade

Avantages

- Facile à utiliser et à comprendre
- Structure simple pour une équipe inexpérimentée
- Fonctionne bien quand la qualité est beaucoup plus importante que les coûts et le temps

Inconvénients

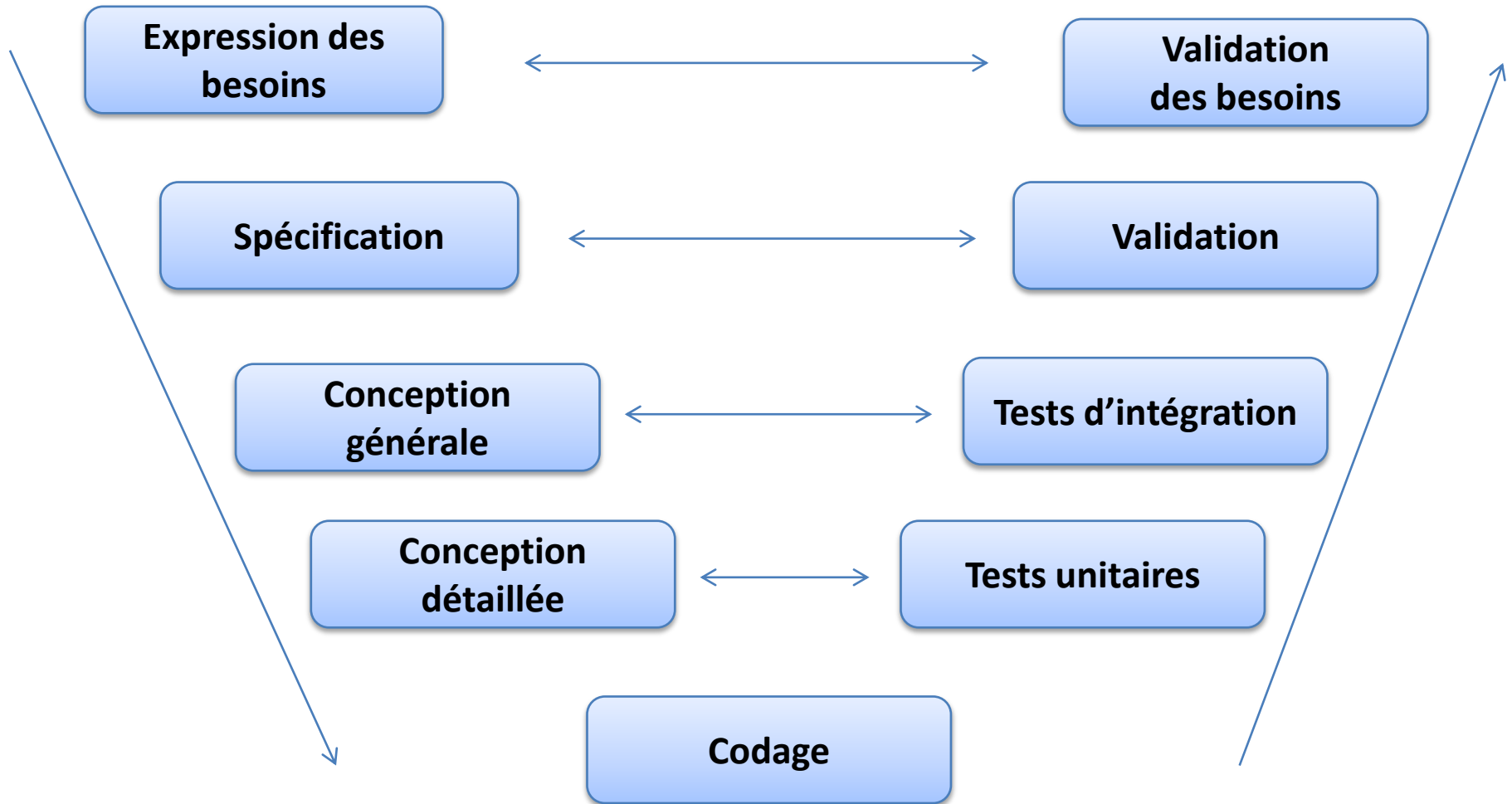
- Sensibilité aux nouveaux besoins : refaire tout le procédé
- Une phase ne peut démarrer que si l'étape précédente est finie
- Le produit n'est visible qu'à la fin
- Les risques se décalent vers la fin
- Très faible implication du client

Modèle en Cascade

Quand l'utiliser ?

- Quand les besoins sont connus et stables
- Quand la technologie à utiliser est maîtrisée
- Lors de la création d'une nouvelle version d'un produit existant

Le modèle en V (1980)



Le modèle en V

- Variante du modèle en cascade qui fait l'accent sur la vérification et la validation
- Le test du produit se fait en parallèle par rapport aux autres activités

Quand l'utiliser:

- Quand le produit à développer à de très hautes exigences de qualité
- Quand les besoins sont connus à l'avance
- Les technologies à utiliser sont connues à l'avance

Le modèle en V

Avantages

- Met l'accent sur les tests et la validation et donc accroît la qualité
- Chaque livrable doit être testable
- Facile à utiliser et planifier

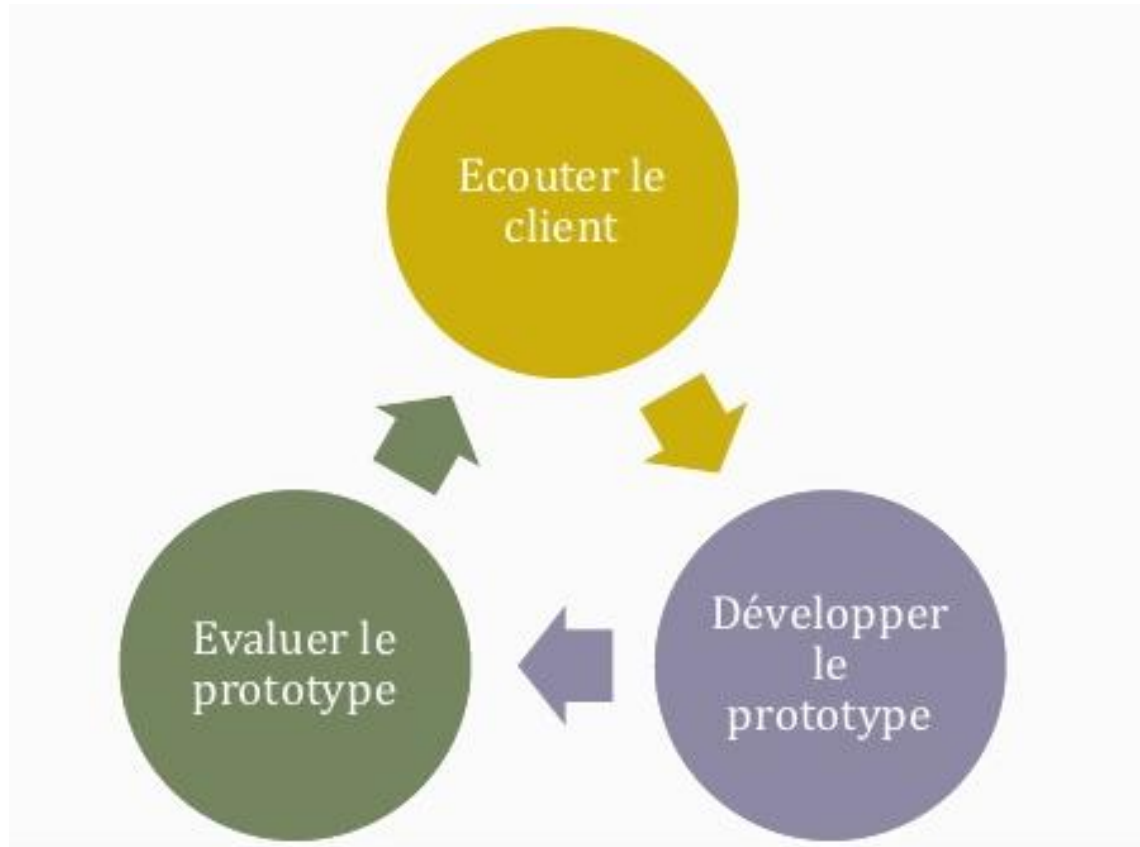
Inconvénients

- Ne gère pas les changements des spécifications
- Ne contient pas d'activités d'analyse de risque

Développement par prototypage

- Le projet se fait sur plusieurs itérations
- Les développeurs construisent un prototype selon les attentes du client
- Le prototype est évalué par le client
- Le client donne son feedback
- Les développeurs adaptent le prototype selon les besoins du client
- Quand le prototype satisfait le client, le code est normalisé selon les standards et les bonnes pratiques

Prototypage



Développement par prototypage

Principe

- Développement **rapide** d'un prototype **avec** le **client** pour **valider** ses besoins
- Écriture de la **spécification** à partir du **prototype**, puis processus de développement linéaire.

Avantage : **Validation concrète** des besoins, **moins de risques d'erreur** de spécification

Prototypage

Avantages

- Implication active du client
- Le développeur apprend directement du client
- S'adapte rapidement aux changements des besoins
- Progrès constant et visible
- Une grande interaction avec le produit

Inconvénients

- Le processus peut ne jamais s'arrêter
- Très difficile d'établir un planning

Prototypage

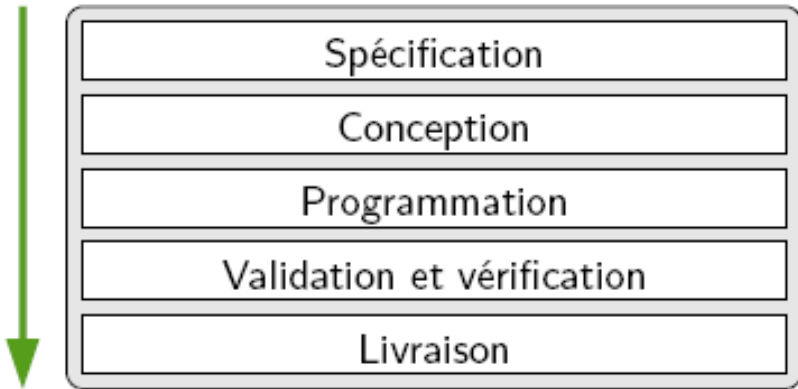
Quand l'utiliser ?

- Quand les besoins sont instables et/ou nécessitent des clarifications
- Peut être utilisé avec le modèle en cascade pour la clarification des besoins
- Quand des livraisons rapides sont exigées

Prototypage

Principe :

- **Hiérarchiser les besoins** du client
- Concevoir et livrer au client un produit implantant un sous ensemble de fonctionnalités par ordre de priorité



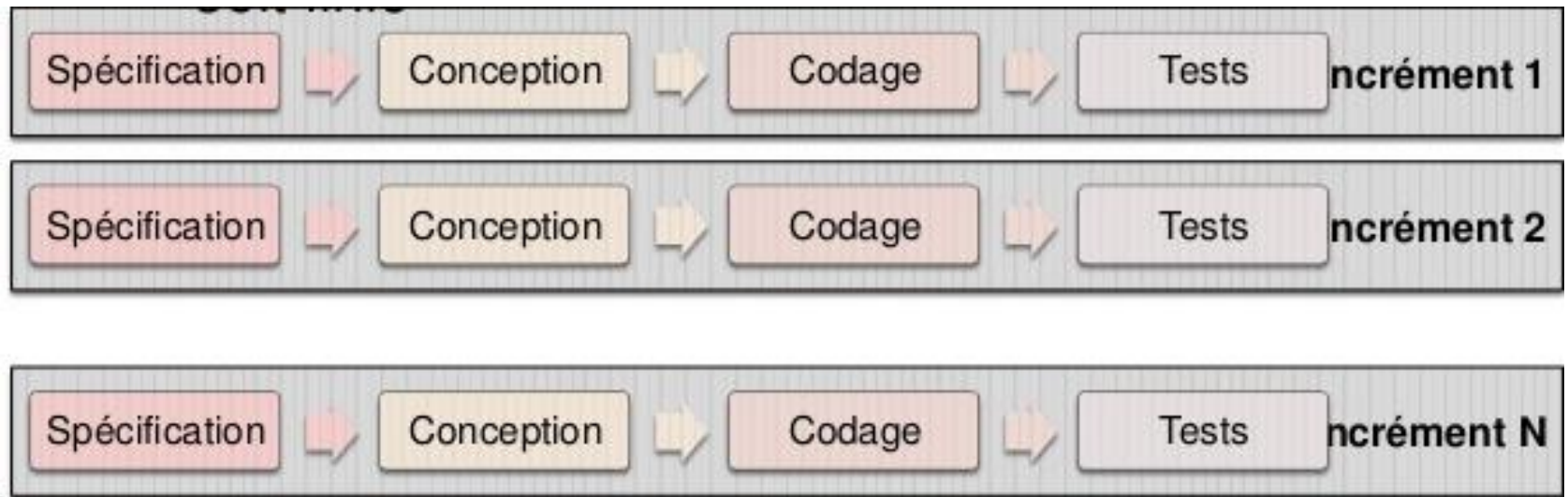
Développement en cascade



Développement incrémental

Le modèle incrémental

- ✓ Chaque incrémentation est une construction partielle du logiciel
- ✓ Trie les spécifications par priorités
- ✓ Chaque incrément implémente un ou plusieurs spécifications jusqu'à ce que la totalité du produit soit finie



Le modèle incrémental

Avantages

- Développement de fonctionnalités à risque en premier
- Chaque incrément donne un produit fonctionnel
- Le client intervient à la fin de chaque incrément
- Utiliser l'approche « diviser pour régner »
- Le client entre en relation avec le produit très tôt

Inconvénients

- Exige une bonne planification et une bonne conception
- Exige une vision sur le produit fini pour pouvoir bien le diviser en incréments
- Le coût total du système peut être cher

Le modèle incrémental

Quand l'utiliser ?

- Quand la plupart des spécifications sont connues à l'avances et vont être sujettes à de faibles évolutions
- Quand on veut rapidement un produit fonctionnel

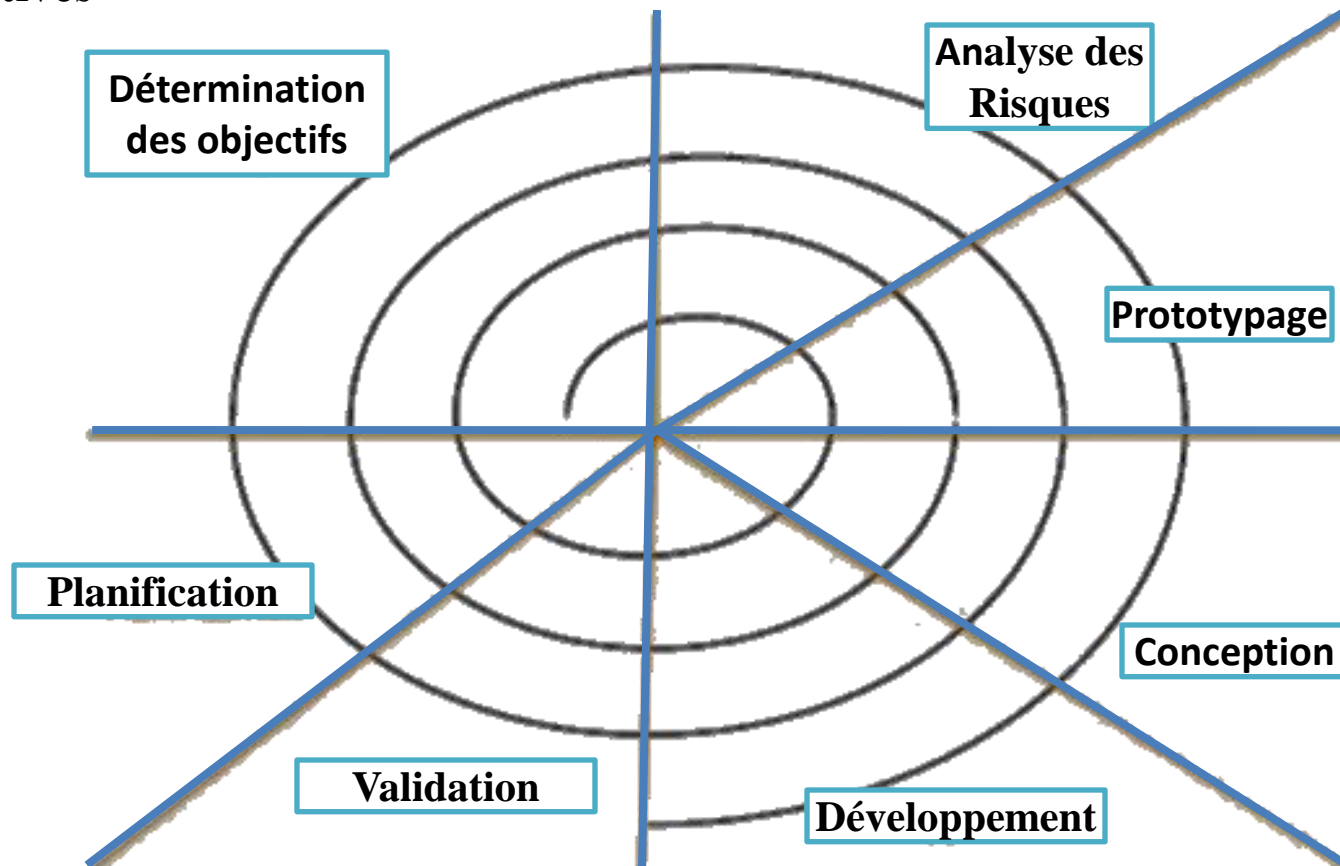
Le modèle en spirale (BOEHM 1988)

- Modèle itératif
- Des incréments sous forme de cycle
- À la fin de chaque cycle on détermine les objectifs du cycle suivant
- Chaque cycle est composé des même activités que du modèle en cascade
- Inclut l'analyse de risque et le prototypage

Le modèle en spirale

Phase 1:
**Détermination des objectifs
et d'alternatives**

Phase 2: identification
et analyse de Risques



Phase 4: Planification de la
prochaine itération

Phase 3: Développement et test

Le modèle en spirale

Détermination des objectifs

- En terme de fonctionnalité, de performance, de coût,...etc.
- Déterminer les alternatives : développer, réutiliser, acheter, sous-traiter...etc.
- Contraintes : coûts, plannings, ... etc.

Le modèle en spirale

Identification et évaluation de risques

- Etudier les alternatives de développement
- Identification des risques : technologie non maîtrisées, équipe peu expérimentée, planning trop serré, ...etc.
- Evaluation des risques : voir si les risques peuvent impacter le projet et à quel degré

Le modèle en spirale

Développement et test

- Contient pratiquement la plupart des activités : conception, codage, test, ... etc.

Planification de la prochaine itération

- Un planning de l'itération
- Un plan de tests

- Si l'**objectif du cycle a été atteint** donc l'**objectif suivant** doit être **défini**.
- Si l'**étape de développement précédente ne s'est pas déroulée comme prévu**.
 - **ex**: la stratégie suivie jusqu'alors peut être **remplacée** par l'une des **alternatives déjà définies** au préalable **ou bien par une nouvelle alternative**.

Le modèle en spirale

Avantages

- Inclut l'analyse de risque et le prototypage
- Fonctions critiques développées en premier
- Feedback rapide du client
- Une évaluation continue du procédé
- Chaque cycle est composé des mêmes activités que du modèle en cascade

Inconvénients

- L'évaluation des risques peut prendre beaucoup de temps
- Le modèle est très complexe

Le modèle en spirale

Quand est-ce que l'utiliser ?

- Quand le prototypage est exigé
- Quand le risque du projet est considérable
- Quand les spécifications ne sont pas stables
- Pour les nouveaux produits