

Cycle de vie d'un Logiciel

Cycle de vie d'un logiciel

« La qualité du processus de fabrication est garante de la qualité du produit »

- Ensemble d'activités conduisant à la production d'un logiciel:
processus de développement ou Cycle de vie d'un logiciel
- Buts:
 - ✓ Maitriser le projet,
 - ✓ bien le découper
 - ✓ assigner correctement les tâches
 - ✓ anticiper et gérer les risques.

Cycle de vie d'un logiciel

➤ On peut distinguer deux classes de Cycle de vie logiciel:

✓ Modèles linéaires (classiques)

- Modèle en cascade
- Modèle en V

✓ Modèles incrémentaux et itératifs

- Modèle en prototype
- Modèle en spirale
- Méthodes Incrémentales

Cycle de vie d'un logiciel

Un processus de développement permet de décrire l'enchainement des différentes étapes du développement. L'objectif est de proposer un processus qui permet de contrôler le développement, afin que le logiciel :

- Soit livré dans les délais ;
- Respecte le budget ;
- Soit de qualité.

Modèles de cycle de vie d'un logiciel

- Un cycle de vie : Les modèles du cycle de vie du logiciel sont des « plans de travail » qui permettent de planifier le développement.
- A quoi ça sert un modèles de cycle de vie?
 - Le cycle de vie d'un logiciel indique les étapes par lesquelles doivent passer un logiciel de sa conception jusqu'à sa livraison.
 - Ce cycle de vie permet de détecter les erreurs tout au long du processus de réalisation et ainsi les corriger pour produire un logiciel de qualité

Modèles de cycle de vie d'un logiciel

Classe de Modèle	caractéristiques
Modèles classiques	<ul style="list-style-type: none">• Modèles stricts• Etapes clairement définies• Documentation exigée• Convenable aux grands projets
Modèles incrémentaux et itératifs	<ul style="list-style-type: none">• Petites et fréquentes livraisons• Accent sur le code et moins sur la documentation• Convenable aux projets de petite ou moyenne taille.

Modèles de cycle de vie d'un logiciel : Modèle en cascade

➤ Principes:

- ✓ Dans le modèle en cascade on effectue les différentes étapes du logiciel de façon séquentielle.
- ✓ Chaque étape doit être terminée avant de commencer la suivante
- ✓ À chaque étape, production d'un document base de l'étape suivante

Modèles de cycle de vie d'un logiciel:

Modèle en cascade

Le processus de développement utilisant un cycle en cascade exécute des phases qui ont pour caractéristiques :

- De produire des livrables
- De se terminer à une date précise
- De ne se terminer que lorsque les livrables sont jugés satisfaisants.

Modèles de cycle de vie d'un logiciel:

Modèle en cascade

Avantages

- Hérité des méthodes classiques d'ingénierie
- Facile à utiliser et à comprendre
- Idéal pour la gestion et le suivi des projets
- Choix idéal quand la qualité est plus importante que les délais et le coûts

Inconvénients

- Découverte tardive des erreurs
- Découverte d'une erreur entraîne un retour à la phase origine de l'erreur
 - ❖ nouvelle cascade, avec de nouveaux documents...
- Coût de modification d'une erreur important,
- Le produit n'est visible qu'à la fin
- Très faibles implication du client
- Pas toujours adapté à une production logicielle
 - ❖ les besoins du client changent fréquemment
 - ❖ Les besoins du client sont difficiles à spécifier

Modèles de cycle de vie d'un logiciel:

Modèle en cascade

Quand l'utiliser?

- Quand les besoins sont stables et bien définis
- Quand la technologie à utiliser est maîtrisée
- Lors de la création d'une nouvelle version d'un produit existant
- Lors du portage d'un produit sur une autre plateforme

Modèles de cycle de vie d'un logiciel : Modèle en V

Principe :

- Variante du modèle en cascade
- Accent sur la vérification et la validation
- Mise en évidence de la complémentarité des phases menant à la réalisation et des phases de test permettant de les valider (Test se fait en parallèle)

Modèles de cycle de vie d'un logiciel :

Modèle en V

- Test unitaire : test de chaque unité de programme (méthode, classe, composant), indépendamment du reste du système
- Test d'intégration : test des interactions entre composants (interfaces et composants compatibles)
- Test système : test du système complet par rapport à son cahier des charges
- Test d'acceptation (recette) : fait par le client, validation par rapport aux besoins initiaux

Modèles de cycle de vie d'un logiciel : Modèle en V

Avantages

- Accent sur les tests et la validation
- Chaque livrable doit être testable
- Facile à planifier
- Facile à utiliser

Inconvénients

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

Modèles de cycle de vie d'un logiciel : Modèle en V

Quand l'utiliser?

- Quand le produit à développer à de très hautes exigences de qualité
- Quand les besoins sont connus
- Les technologies à utiliser sont maîtrisées

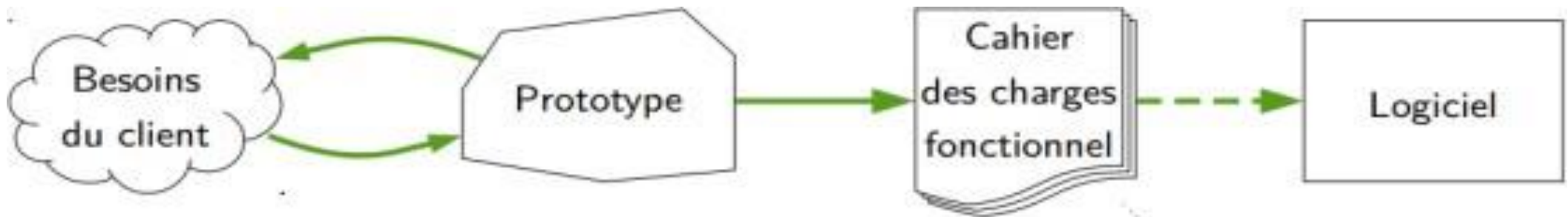
Modèles de cycle de vie d'un logiciel : Cycle de vie itératif et incrémental

- **Itératif** : le processus de développement est appliqué plusieurs fois
- **Incrémental** : chaque itération augmente la quantité d'information
- Une amélioration du modèle en cascade

Modèles de cycle de vie d'un logiciel : Modèle en prototype

➤ Principe :

- ✓ Développement rapide d'un prototype avec le client pour valider ses besoins
- ✓ Le prototype est évalué par le client
- ✓ Écriture de la spécification à partir du prototype, puis processus de développement linéaire
- ✓ Méthode appropriée lorsque le besoin utilisateur n'est pas clair et peut changé fréquemment



Modèles de cycle de vie d'un logiciel : Modèle en prototype

Avantages

- Validation concrète des besoins
- Implication active du client (Le développeur apprend directement du client)
- moins de risques d'erreur de spécification
- S'adapte rapidement aux changements des besoins
- Progrès visible et constant

Inconvénients

- Code faiblement structurer
- Degré très faible de maintenabilité
- Le processus peut ne jamais s'arrêter
- Très difficile d'établir un planning

Modèles de cycle de vie d'un logiciel : Modèle en prototype

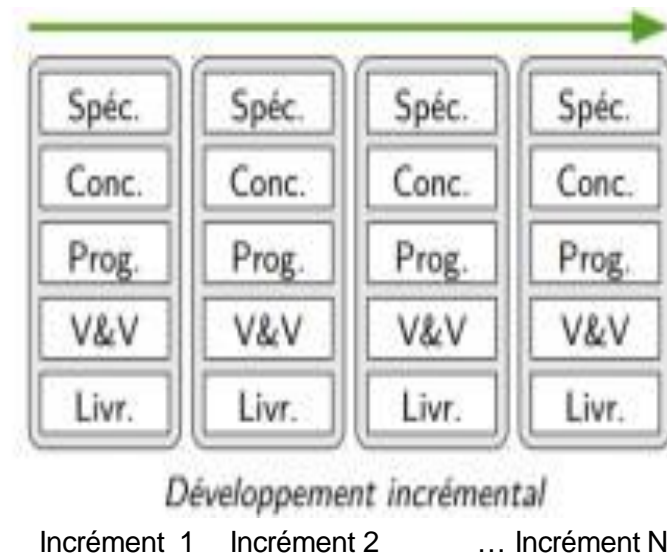
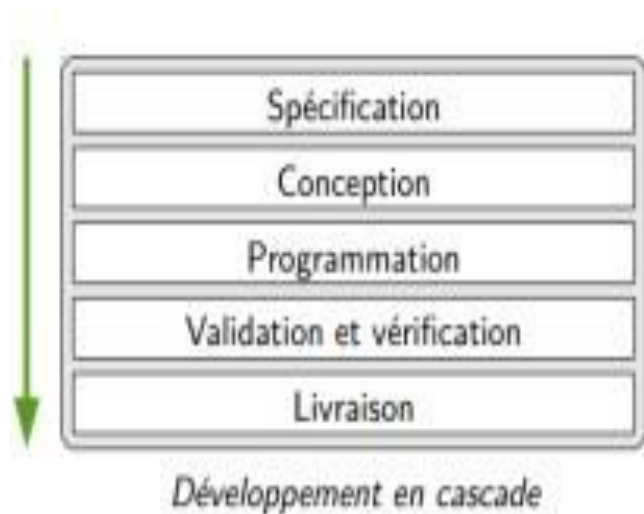
Quand l'utiliser?

- Quand les besoins sont instables et/ou nécessitent des clarifications
- Quand des livraisons rapides sont exigées

Modèles de cycle de vie d'un logiciel : Modèle incrémental

Principe :

- Hiérarchiser les besoins du client: chaque incrément est une construction partielle du logiciel
- Trier les spécifications par priorités et les regrouper dans des sous-ensembles de spécification.
- Chaque incrément conçoit et livre au client un produit implantant un sous-ensemble de spécification selon leur ordre de priorité jusqu'à ce que la totalité du produit soit fini.



Modèles de cycle de vie d'un logiciel : Modèle incrémental

Avantages:

- Minimiser le risque d'inadéquation aux besoins
- Développer les fonctionnalités à risque en premier
- Chaque incrément donne un produit fonctionnel
- S'appuie sur le paradigme « Diviser pour Reigner »
- 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 le diviser en incréments
- Problèmes d'intégration des fonctionnalités secondaires non pensées en amont.

Modèles de cycle de vie d'un logiciel : Modèle incrémental

Quand l'utiliser?

- Quand la plupart de spécifications sont connues à l'avance et peuvent être sujet de faibles évolutions.
- Quand on veut un produit rapidement fonctionnel
- Pour des produits de longues durées
- Pour des projets impliquant de nouvelles technologies

Modèles de cycle de vie d'un logiciel : Modèle en spiral

Principe: Modèle mixte

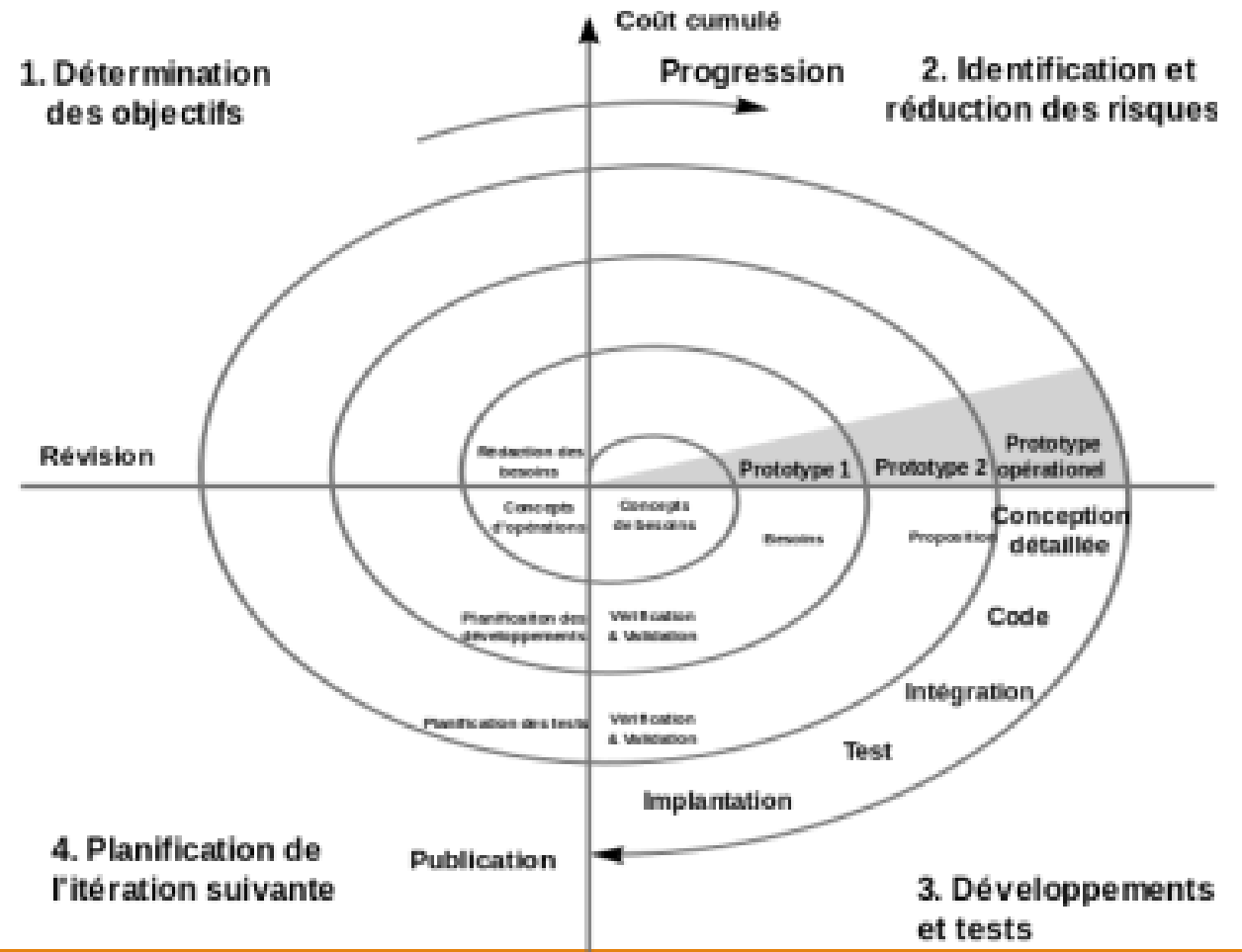
➤ A chaque cycle, recommencer :

1. Consultation du client
2. Analyse des risques
3. Conception
4. Implémentation
5. Tests
6. Planification du prochain cycle

Modèles de cycle de vie d'un logiciel : Modèle en spirale

1. Détermination des objectifs :

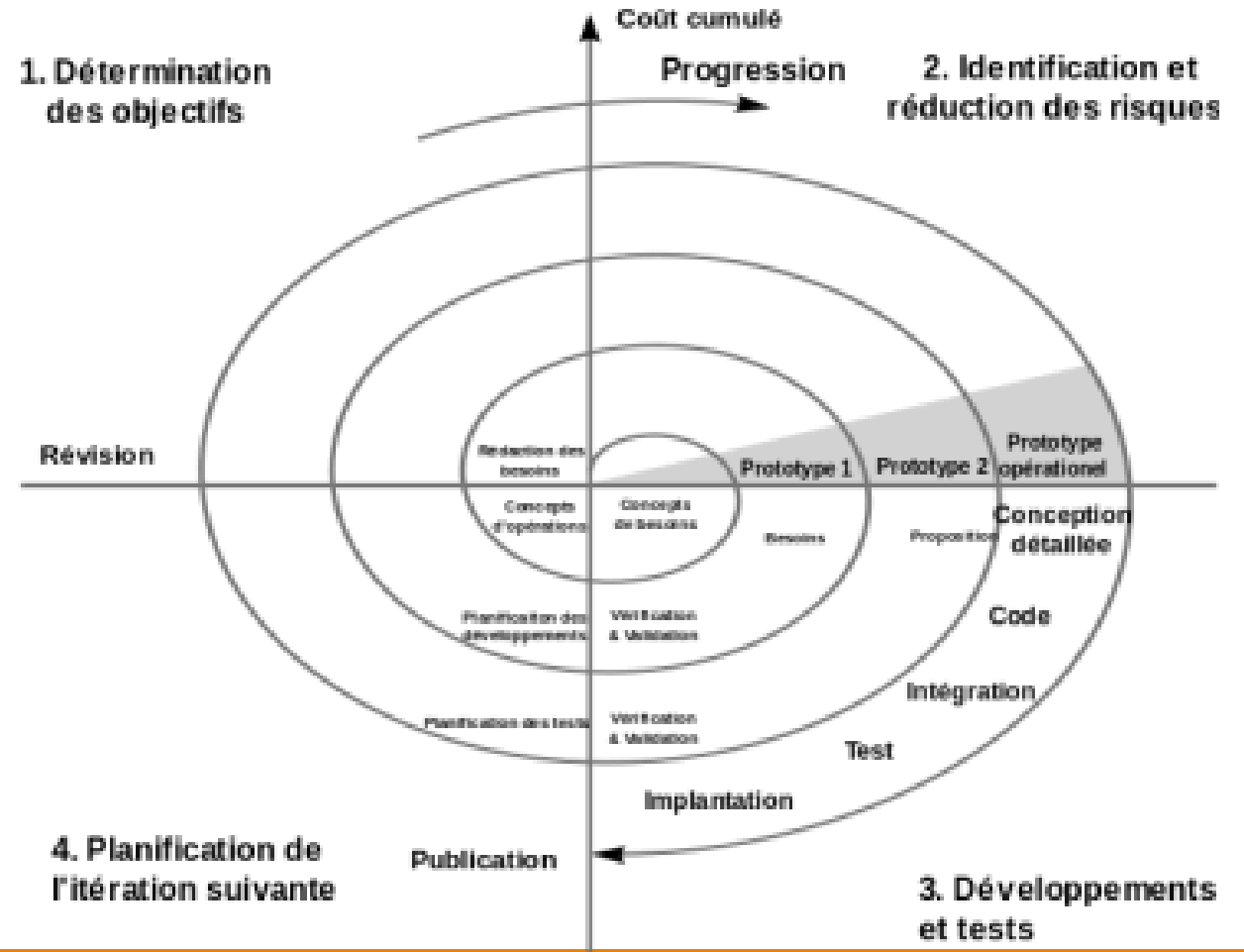
les objectifs sont déterminés conjointement avec le client. Dans le même temps, les alternatives possibles sont discutées et les conditions cadres sont spécifiées



Modèles de cycle de vie d'un logiciel : Modèle en spirale

2. Identification et réduction des risques :

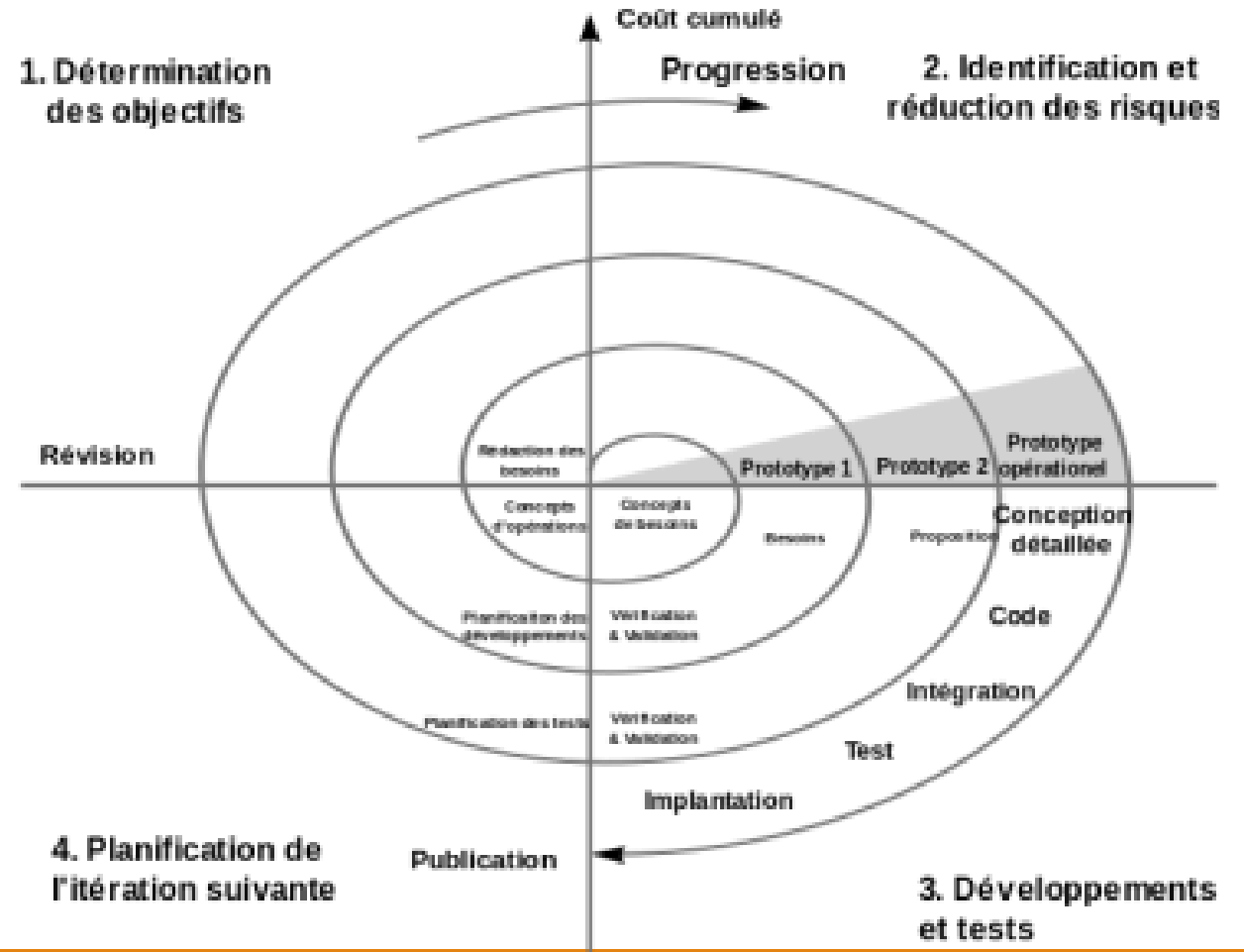
les risques potentiels sont identifiés , évalués. estimés puis réduits à l'aide de prototypes, des simulations et des logiciels d'analyse.



Modèles de cycle de vie d'un logiciel : Modèle en spiral

3. Développement et teste :

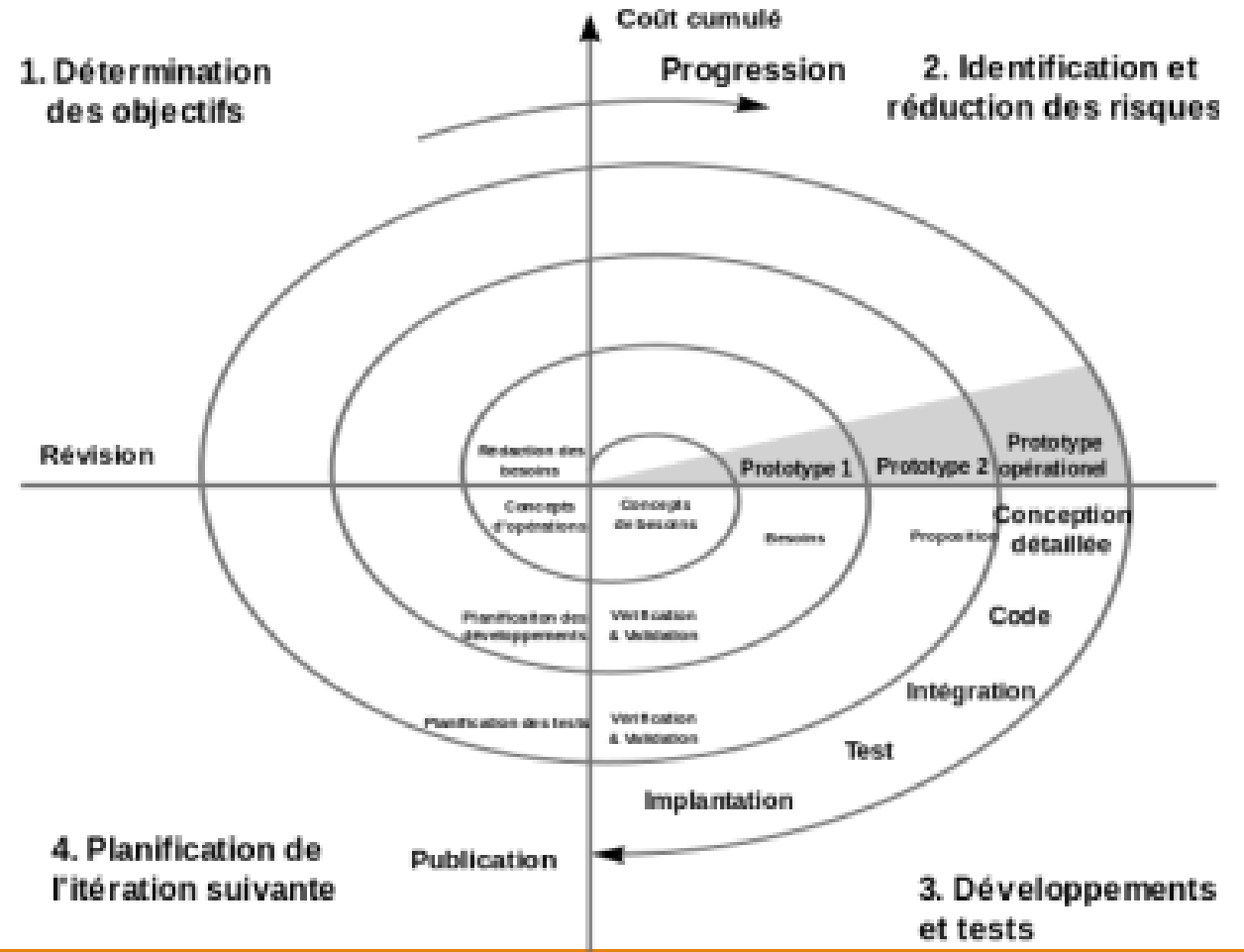
les prototypes sont encore plus étendus et des fonctionnalités sont ajoutées. Le code réel est écrit, testé et migré vers un environnement de test plusieurs fois jusqu'à ce que le logiciel puisse être implémenté dans un environnement productif.



Modèles de cycle de vie d'un logiciel : Modèle en spiral

4. Planification de l'itération suivante :

le cycle à venir est planifié à la fin de chaque cycle. Si des erreurs se produisent, les solutions sont recherchées. Si une meilleure alternative est envisageable, elle sera préférée au sein du cycle suivant.



Modèles de cycle de vie d'un logiciel :

Modèle en spiral

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

Modèles de cycle de vie d'un logiciel : Modèle en spiral

- Détermination des objectifs
 - ✓ En terme de fonctionnalités, performance, coût, etc.
 - ✓ Déterminer les alternatives (développer ou sous traiter,...)
- Identification et réduction des risques
 - ✓ Technologies non maîtrisées, équipe peu expérimentée, planning serré, etc.
- Développement et test
 - ✓ Contient les activités de développement (conception,...)
- Planification de la prochaine itération
 - ✓ Mise en place d'un planning

Modèles de cycle de vie d'un logiciel : Modèle en spirale

Avantages

- Identification rapide des risques
- Impacts minimaux des risques sur le projet
- Fonctions critiques développées en premier
- Feedback rapide du client
- Une évaluation continue

Inconvénients

- Nécessite une grande expérience
- L'évaluation des risques peut prendre beaucoup de temps
- Modèle complexe
- La spirale peut être éternelle!!!
- Les objectifs ne sont pas souvent faciles à formuler.

Modèles de cycle de vie d'un logiciel : Modèle en spiral

Quand l'utiliser?

- Quand le prototypage est exigé
- Quand le risque du projet est considérable
- Quand les spécifications ne sont pas stables
- Quand le projet implique de la recherche et de l'investigation