



**TD 1**  
**Cycle de vie**  
**Solutions proposées**

# TD 1 : Cycle de vie en logiciel

**Objectifs : Connaître les modèles de cycle de vie d'un logiciel**

1. Pourquoi parle t-on de "cycle de vie" en logiciel ?

- Le « **cycle de vie** d'un **logiciel** », désigne toutes les étapes du développement d'un **logiciel**, de sa conception à sa disparition. ...
- Le **cycle de vie** permet de détecter les erreurs au plus tôt et ainsi de maîtriser la qualité du **logiciel**, les délais de sa réalisation et les coûts associés.

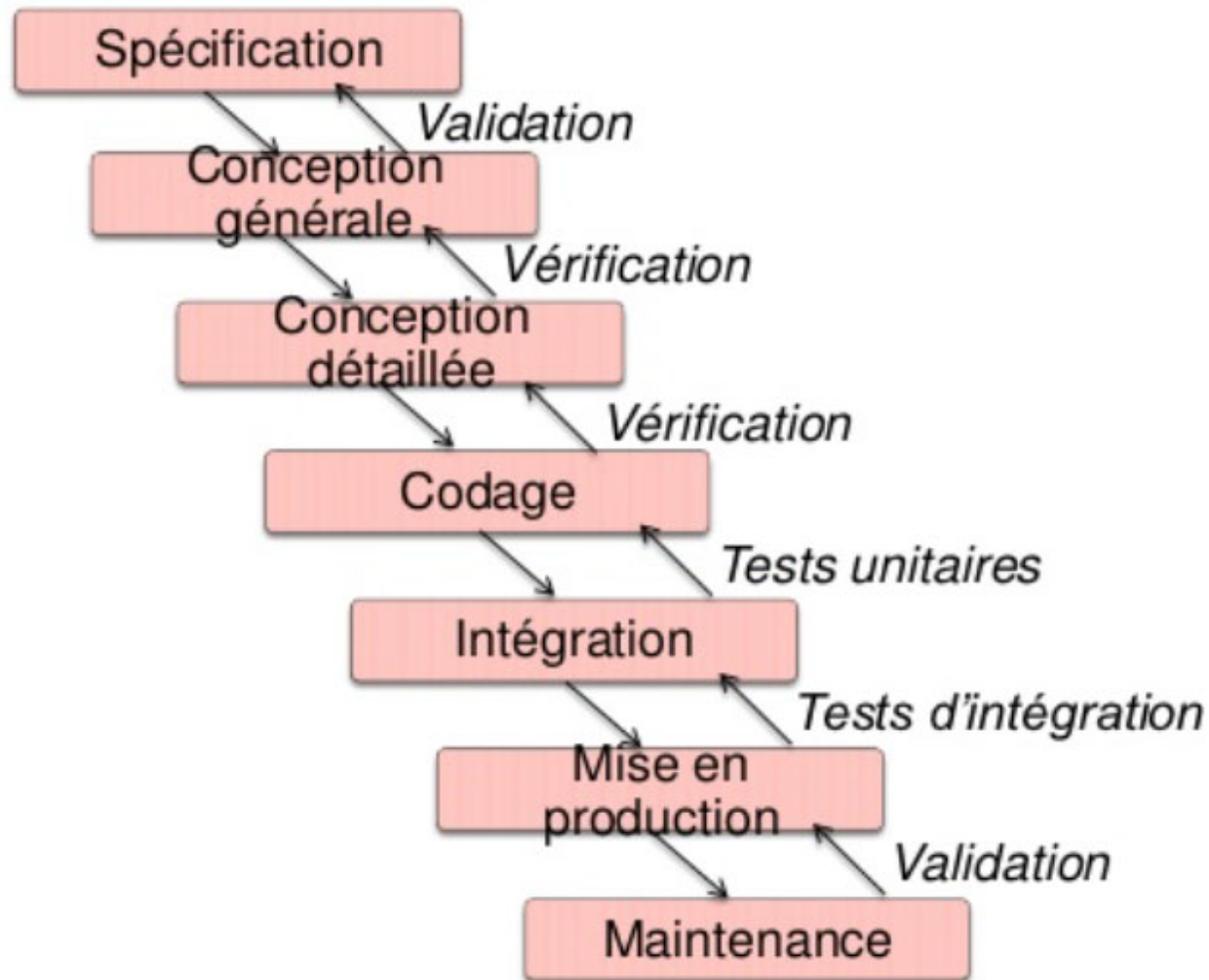
# TD 1 : Cycle de vie en logiciel

**Objectifs : Connaître les modèles de cycle de vie d'un logiciel**

2. Décrivez les principaux cycles de vie du logiciel, montrez leurs principales forces et faiblesses.



# Cycles de vie classiques, modèles en CASCADE



# Cycles de vie classiques, modèles en CASCADE

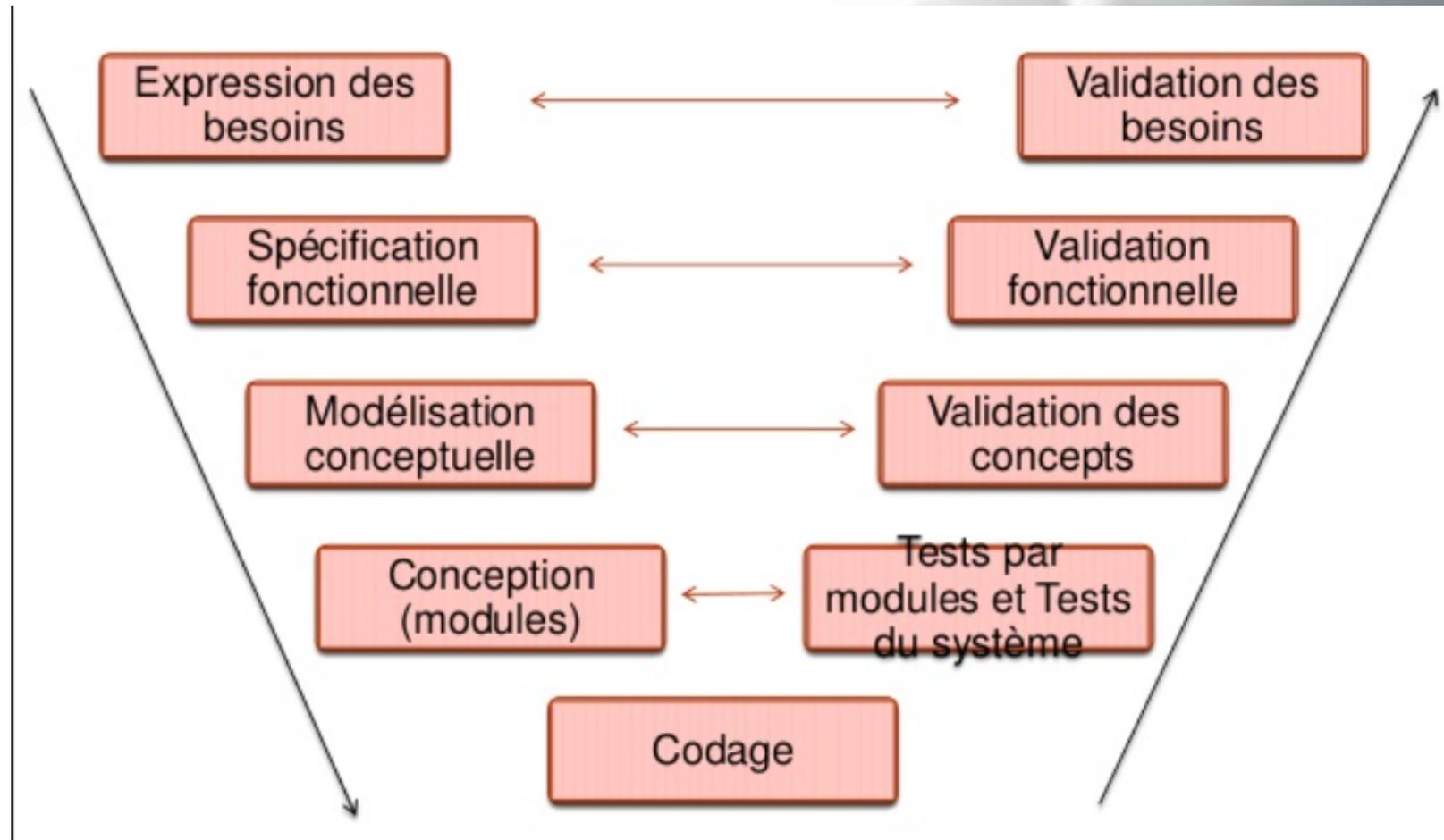
## Avantages

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

# Cycles de vie classiques, modèles en V



## Cycles de vie classiques, modèles en V

### Avantages

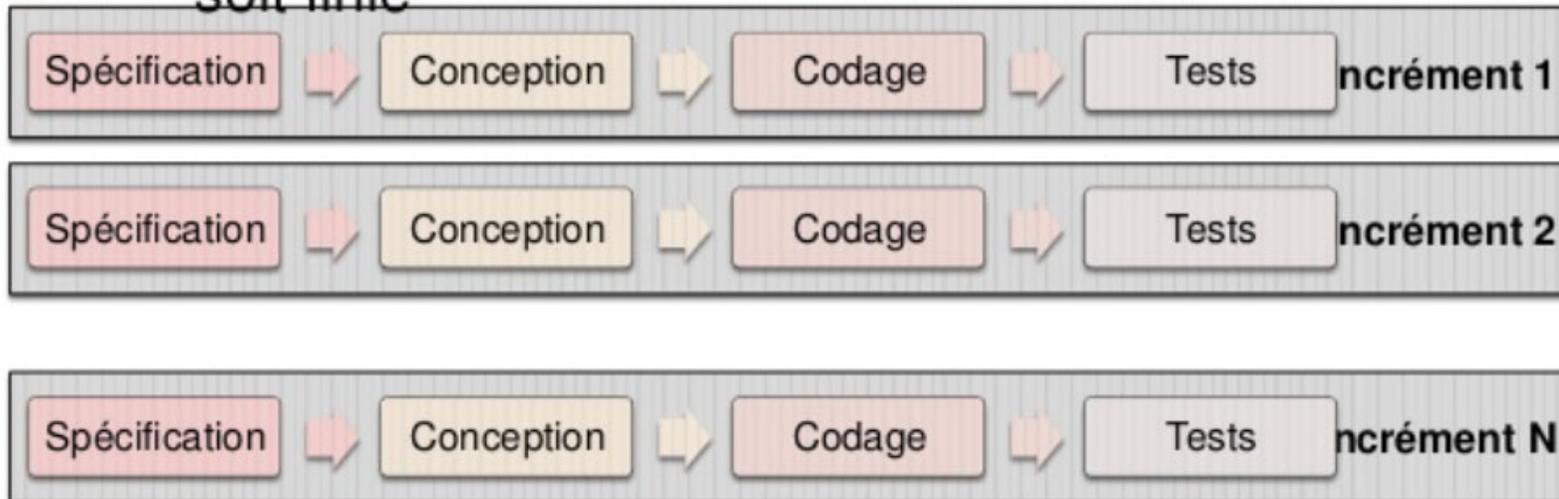
- Met l'accent sur les tests et la validation et donc accroît la qualité
- Chaque livrable doit être testable
- Facile à utiliser et planifier

### Inconvénients

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

# Cycles de vie classiques, modèle INCREMENTAL

- Chaque incrément 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



## Cycles de vie classiques, modèle INCREMENTAL

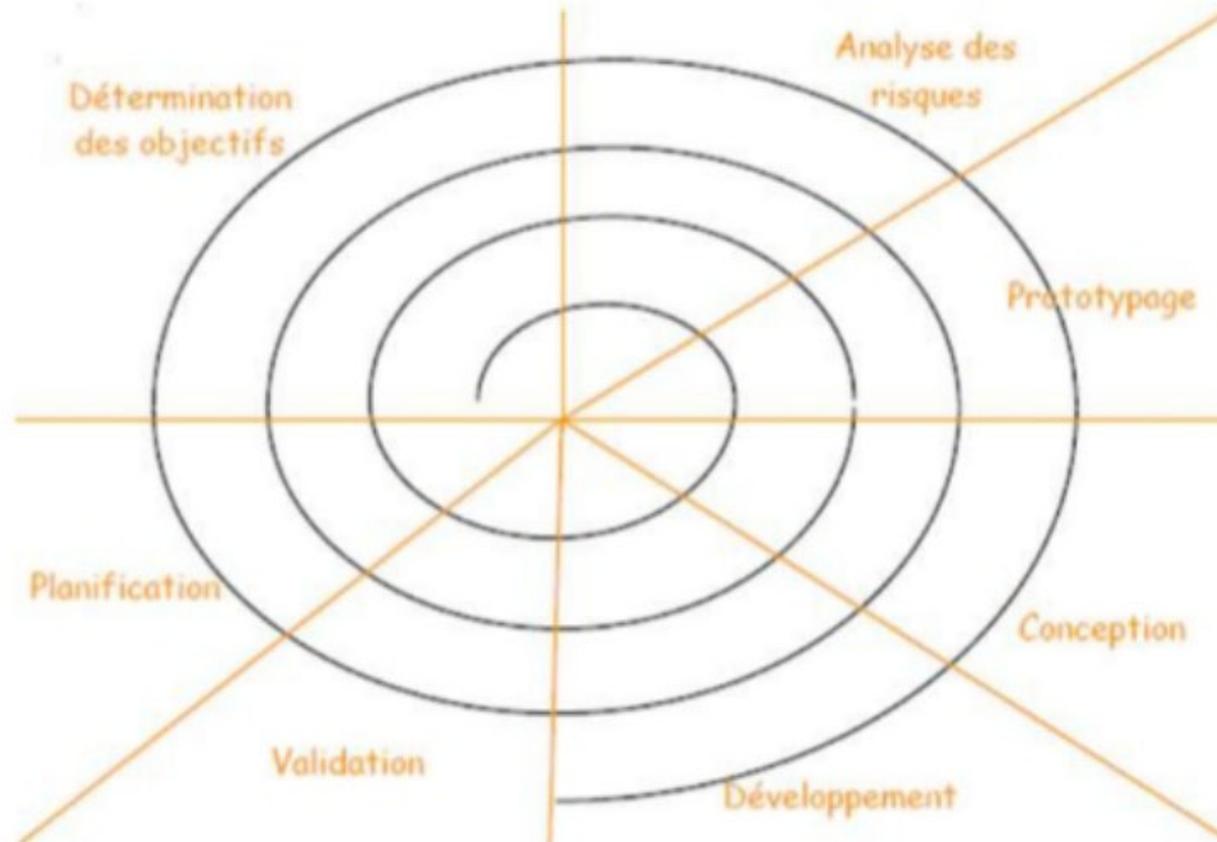
### Avantages

- 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

# Cycles de vie classiques, modèle en SPIRALE



## Cycles de vie classiques, modèle en SPIRALE

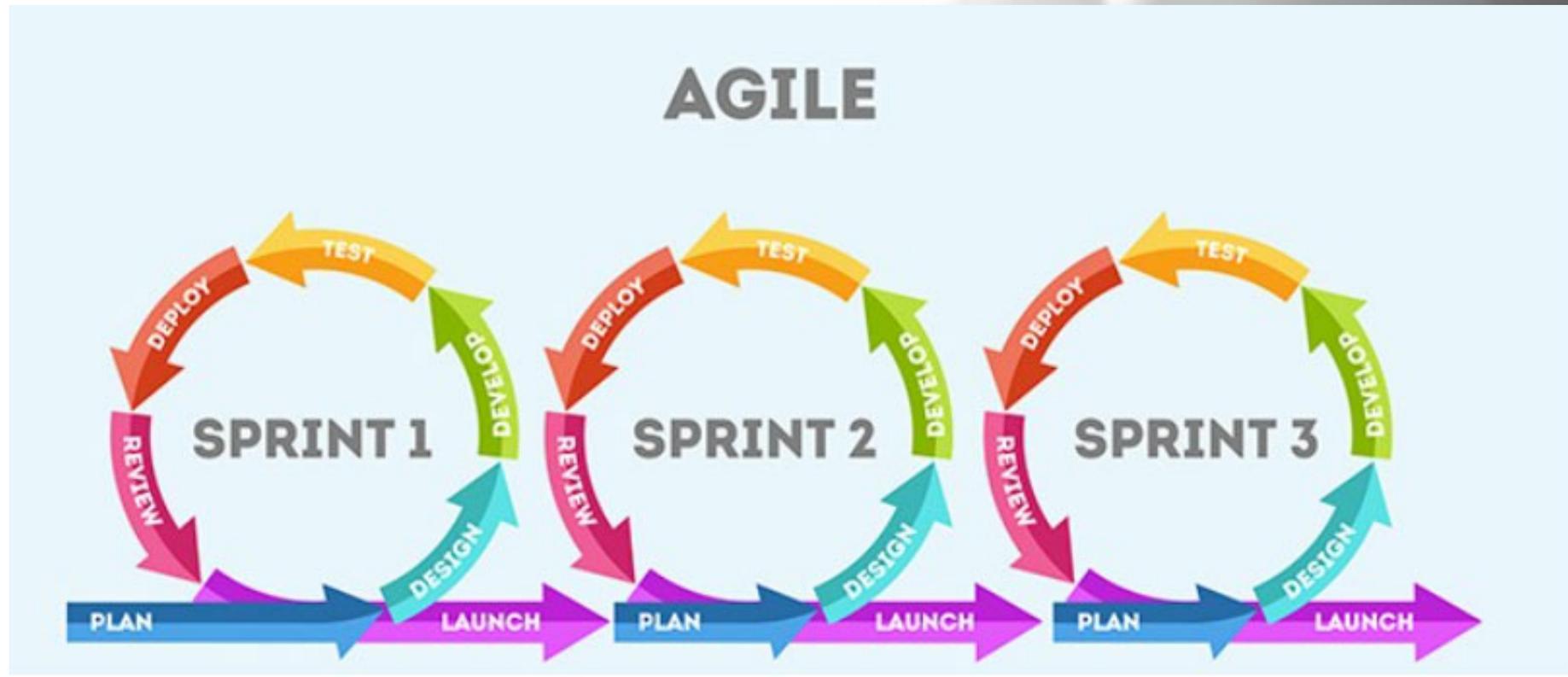
### Avantages

- 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

# TD 1 : Cycle de vie : Méthodes AGILE



En général, les exploitants des méthodes Agile n'utilisent pas le mot " terminer ". En effet, les produits sont toujours perfectibles même s'ils sont sur le marché.

# TD 1 : Avantages Méthodes AGILE

- Les méthodes Agile se distinguent par leur **flexibilité**. Elles impliquent la **participation en amont des clients pour des tests**. En fonction des retours, l'équipe de développement a la possibilité de **réagir rapidement** en effectuant des modifications par exemple.
- **La collaboration étroite entre les développeurs et les clients** permettent l'établissement d'une relation de confiance très utile lors de la commercialisation du produit. Le client final dispose de **meilleures visibilité sur l'avancement du projet**. Il saura à l'avance si le produit lui conviendra.
- **Les coûts du projet sont mieux contrôlés** puisque la réalisation d'un mini-projet permet de déterminer le budget nécessaire à l'atteinte de l'objectif final. Ainsi, le porteur du projet peut décider de poursuivre ou de stopper le projet en fonction de ses fonds.
- **L'agilité permet d'avoir un rythme soutenu et prévisible des livrables. Avec des sprints qui durent de 1 à 2 semaines, vous pouvez lancer de nouvelles fonctionnalités dès qu'elles sont opérationnelles**

# TD 1 : Les Défis des Méthodes AGILE

- Se battre contre l'industrie **agile** complexe et sa tendance à imposer les processus aux équipes ;
- Insister sur l'importance de **l'excellence technique** ;
- **Organiser** les équipes **autour d'un produit et non d'un projet.**
- Modifier la culture historique des **projets** au sein des entreprises

# TD 1 : Fonctionnement des méthodes AGILE

- Le principe des méthodes Agile est **d'éviter les planifications dans le développement d'un produit.**
- Cette habitude est contre-productive selon les adeptes de ces méthodes.
- Etapes à suivre:
  - ✓ **Lancement du projet: création d'un backlog.**  
Il s'agit de définir des objectifs à court terme. Le projet est subdivisé en plusieurs sous-projets. Dès qu'un objectif est atteint, on passe au suivant jusqu'à l'atteinte de la finalité recherchée.
  - ✓ Après avoir créé le backlog, l'initiateur du projet et ses collaborateurs peuvent commencer la **planification du sprint**. Ce dernier désigne la période pendant laquelle les développeurs vont créer un ensemble de fonctionnalités.
  - ✓ Généralement, le sprint s'effectue pendant **10 jours ouvrés**.
  - ✓ Ensuite, toute l'équipe procède à la **revue du sprint**. Il s'agit d'analyser les livrables réalisés par les développeurs. C'est à ce stade que les clients entrent en jeu pour tester les fonctionnalités. Les étapes précédentes sont à refaire en fonction des retours et jusqu'à l'obtention d'un produit commercialisable.

# TD 1 : Cycle de vie SPIRALE

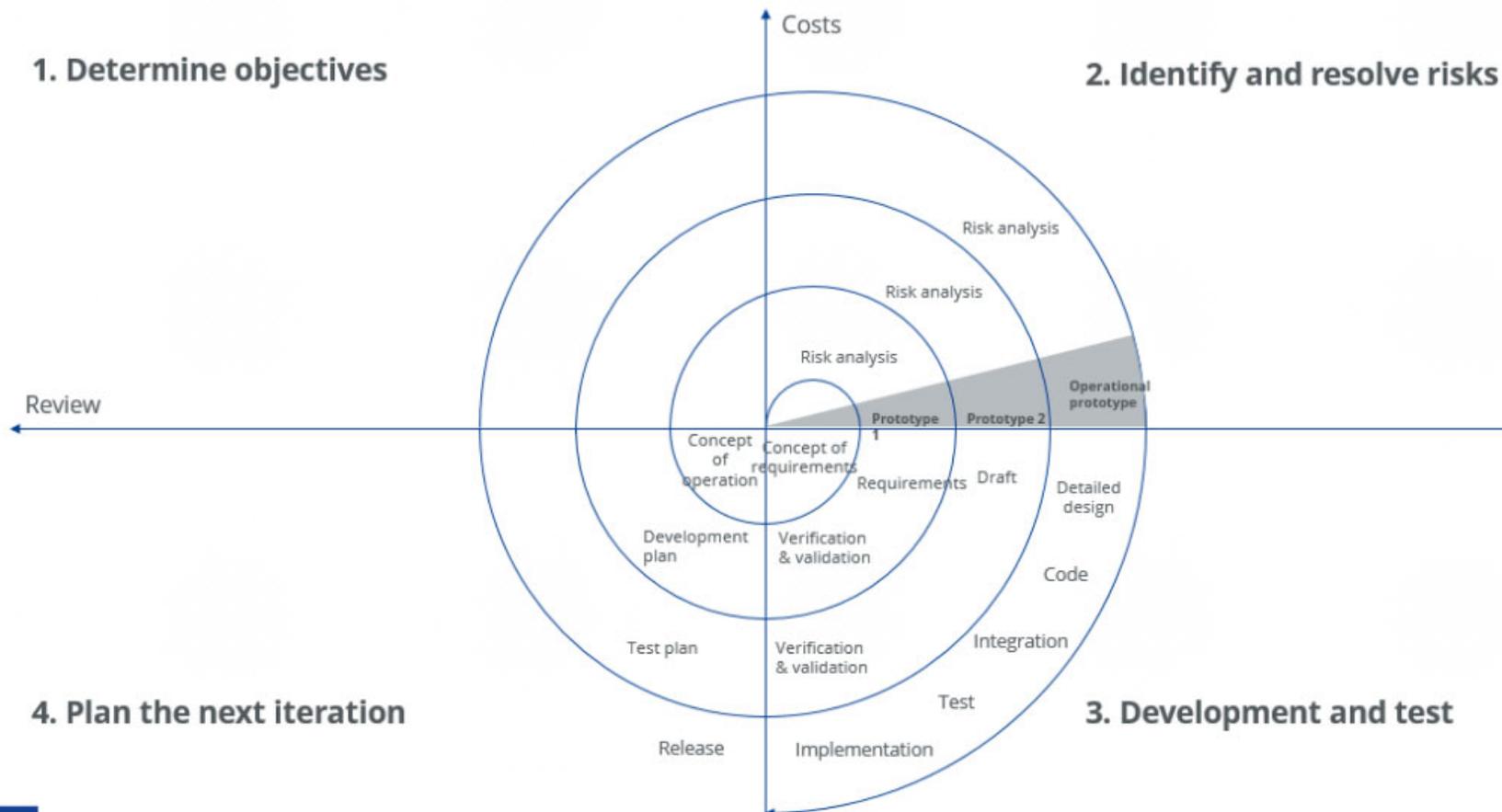
**Objectifs : Connaître les modèles de cycle de vie d'un logiciel**

3. Expliquer les principales caractéristiques du cycle de développement en spirale, par un petit exemple montrer de quelle manière on peut l'utiliser dans un projet.

Faites un schéma du cycle en illustrant l'enchaînement des différentes étapes pour un développement de quelques versions successives (v1.0, v2.0,...) d'un logiciel.

- Le modèle en **spirale** est un modèle de **Cycle de développement logiciel** qui reprend les différentes étapes du **cycle** en **V**.
- Par l'implémentation de **versions successives**, le **cycle** recommence en proposant un produit de plus en plus complet .
- Il représente un cycle itératif, qui doit être répété jusqu'à ce que le but fixé soit atteint. Par une **analyse régulière des risques et des contrôles réguliers du produit intermédiaire**, le modèle en spirale diminue considérablement le risque d'échec lors des projets logiciels de grande taille.

# TD 1 : Cycle de vie SPIRALE



# TD 1 : Cycle de vie SPIRALE

- Le modèle en spirale intègre l'approche par étapes du modèle classique en cascade.
  - ✓ Par exemple, une **spirale à boucle unique** représente en fait le **modèle itératif en cascade**
- Le modèle en spirale utilise l'approche du **modèle de prototypage** en construisant un prototype au début de chaque phase comme technique de gestion des risques.
  - En outre, le modèle en spirale peut être considéré comme supportant le modèle évolutif
    - ✓ les itérations le long de la spirale peuvent être considérées comme des niveaux évolutifs à travers lesquels le système complet est construit.

## TD 1 : Cycle de vie en logiciel

4. Une entreprise de génie logiciel spécialisée en objet souhaite réaliser un petit logiciel de jeux sur Internet, cette demande est inhabituelle pour cette société.

Vous maîtrisez très bien la technologie nécessaire au développement de ce projet qui ne comporte pas de risques techniques.

Un cahier des charges précis est donné par le client.

- Que proposez vous comme cycle de vie de développement. Argumentez votre proposition, montrez les avantages et inconvénients de votre proposition par rapport à d'autres possibles.

## TD 1 : CASCADE MIXTE

- Cahier des charges précis = risques maîtrisés mais non nuls, donc se laisser la possibilité de revenir sur une étape précédente :

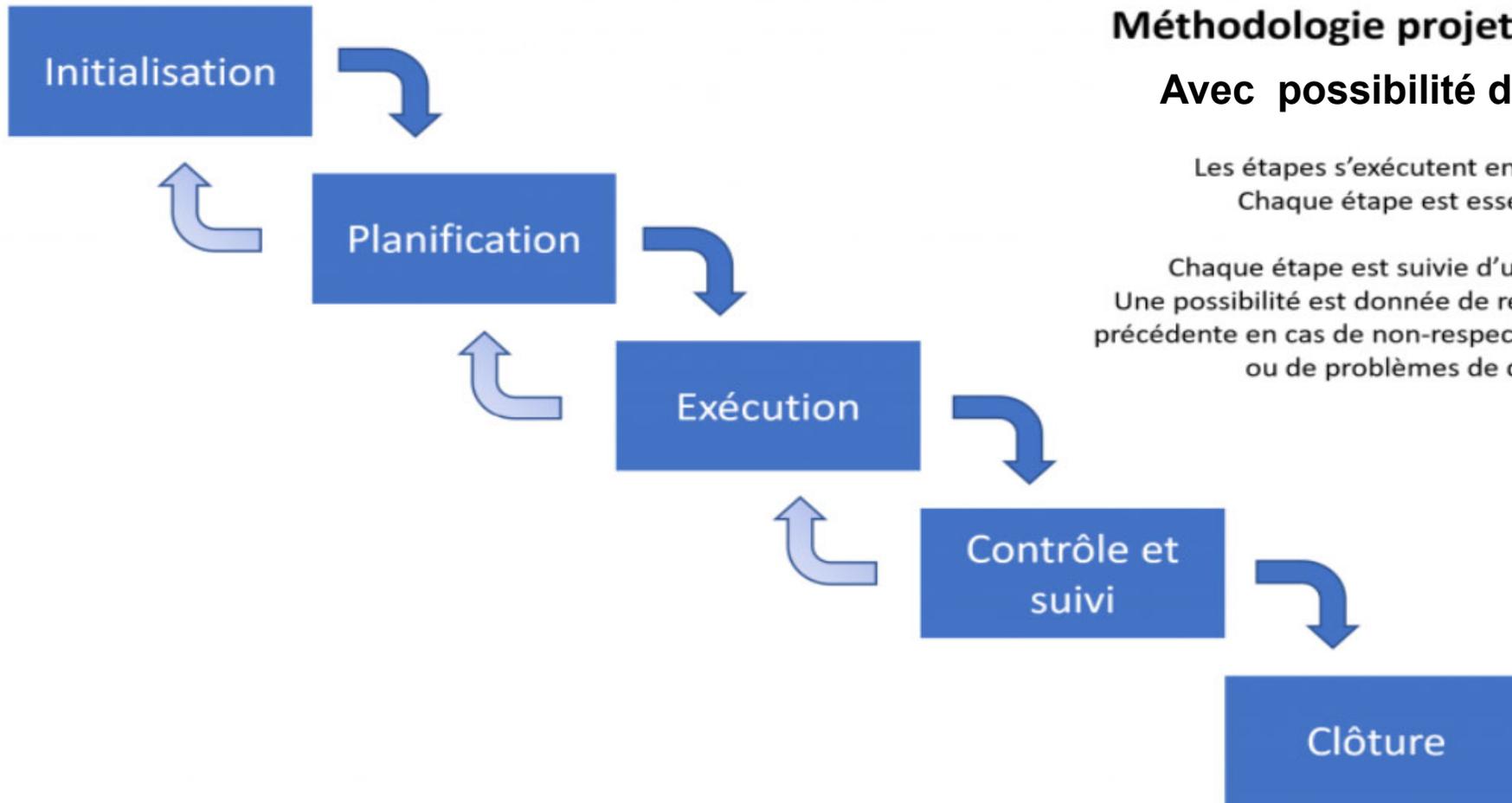
Elle consiste à ajouter une « **porte** » après chaque étape.

Cette porte va permettre de valider la qualité et les travaux de l'étape qui vient de s'achever.

Et aussi permettre de corriger certaines erreurs ou apporter des modifications en décidant de revenir à l'étape précédente si cela est nécessaire pour le bien du projet.

- Simplicité de mise en œuvre, n'implique pas de participation importante du client sur toutes les phases comme les méthodes AGILE

# TD 1 : Cycle de vie CASCADE MIXTE



## Méthodologie projet en cascade Avec possibilité de « retour »

Les étapes s'exécutent en séquence.  
Chaque étape est essentielle.

Chaque étape est suivie d'une « porte ».  
Une possibilité est donnée de retourner à l'étape  
précédente en cas de non-respect des engagements  
ou de problèmes de qualité.



**Fin de document**

**MERCI**