



NFA007

Méthodes pour l'informatisation

CYCLE DE VIE

Cycle de vie d'une application

Le cycle de vie du logiciel comprend généralement les activités suivantes :

- **Définition des objectifs** : définition de la finalité par le demandeur.
- **Analyse des besoins** : formalisation des besoins du demandeur et de l'ensemble des contraintes.
- **Conception générale** : spécifications de l'architecture générale du logiciel.
- **Conception détaillée** : spécifications précises de chaque partie du logiciel.
- **Implémentation** : programmation des fonctionnalités définies lors des phases de conception.

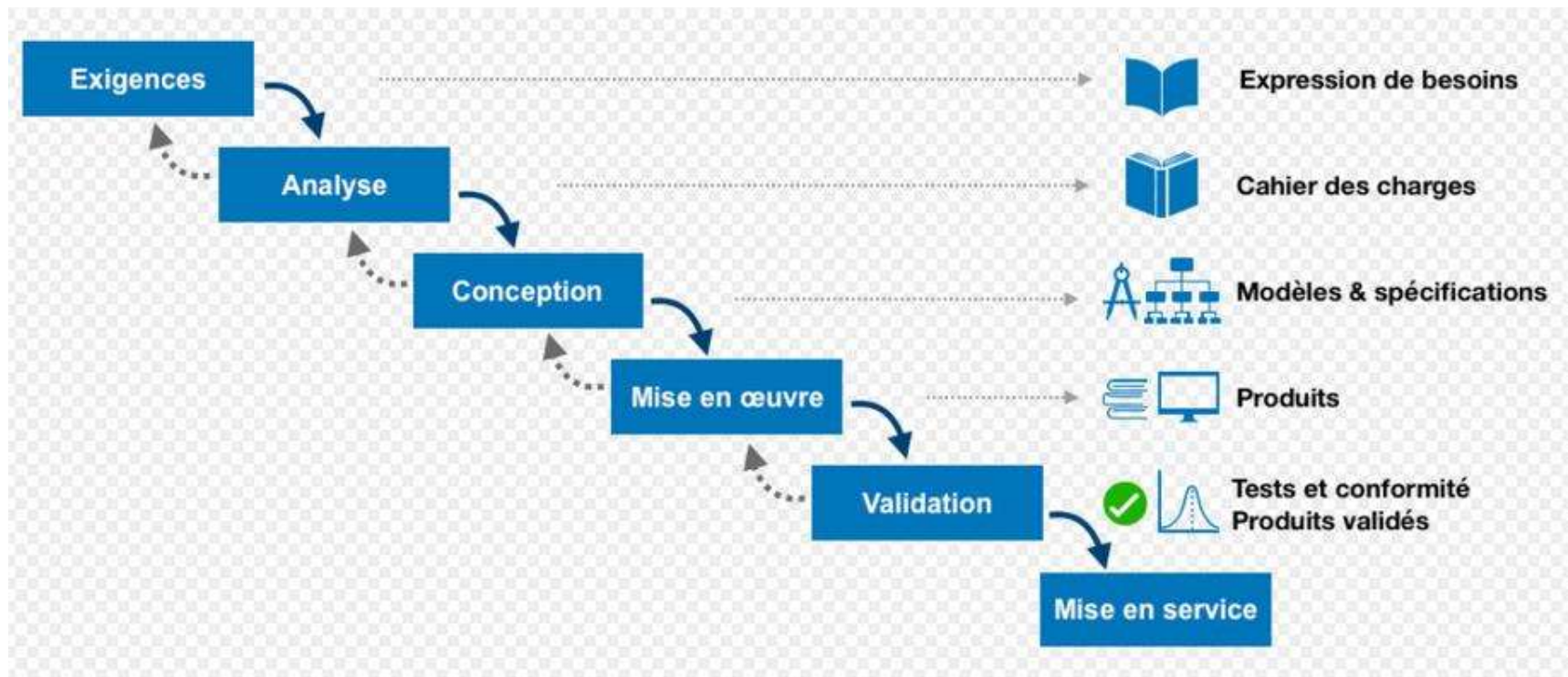
Cycle de vie d'une application

Le cycle de vie du logiciel comprend généralement les activités suivantes :

- **Tests unitaires** : vérification que la partie du logiciel est implémenté conformément aux spécifications.
- **Intégration** : vérification que la partie du logiciel programmée s'interface bien avec différents éléments (modules) du logiciel.
- **Qualification / Recette** : vérification de la conformité du logiciel aux spécifications initiales.
- **Documentation** : production des informations nécessaires pour l'utilisation du logiciel et pour des développements ultérieurs.
- **Mise en production.**
- **Maintenance** : actions correctives (maintenance corrective) et évolutives (maintenance évolutive) sur le logiciel.

Cycle de vie d'une application

Modèle en cascade (waterfall)



(référence : https://fr.wikipedia.org/wiki/Mod%C3%A8le_en_cascade)

Cycle de vie d'une application

Modèle en cascade (waterfall)

Principe : A chaque étape, il est produit un “livrable” (deliverable) qui est nécessaire pour l'étape suivante.

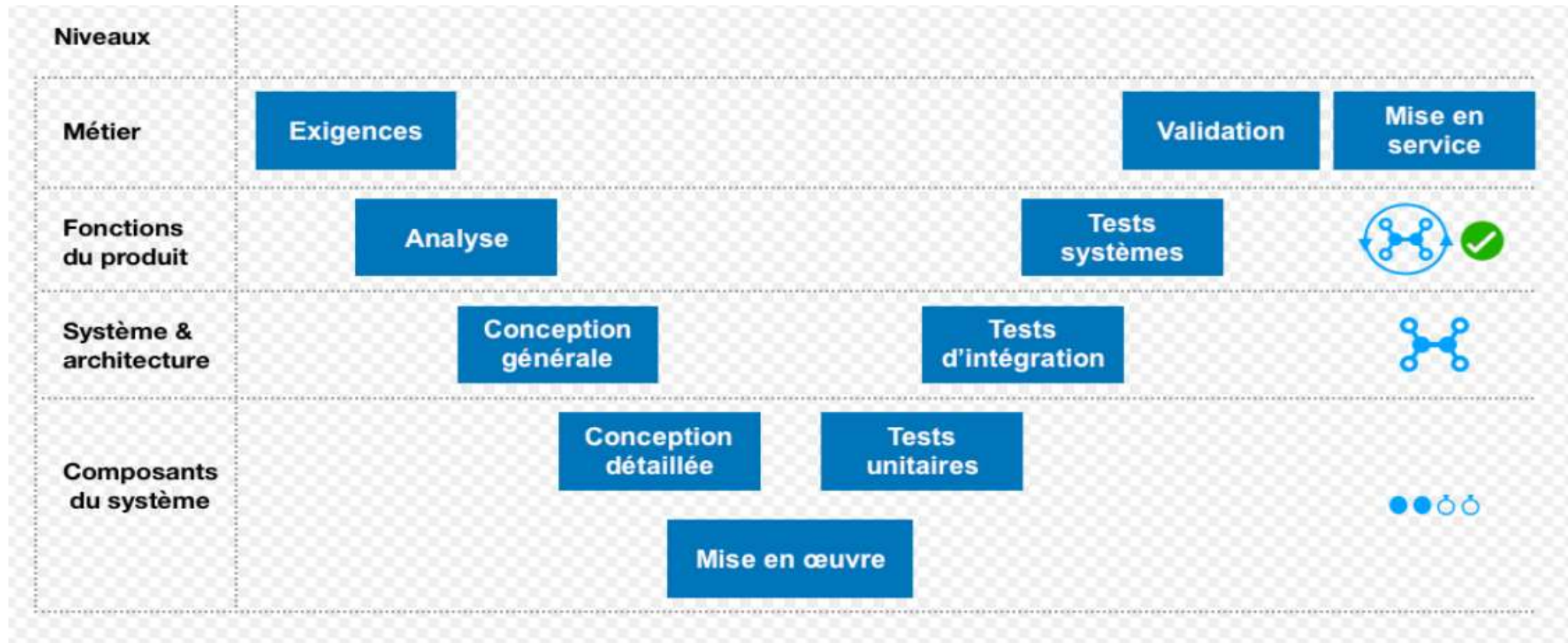
Avantages : Démarche simple à comprendre et du coup à mettre en place.

Risques :

- Coûts importants de correction : Si une erreur est rencontrée, on retourne à la phase à l'origine de cette erreur et on recommence un cycle.
- Boîte noire pour les utilisateurs: les utilisateurs n'interviennent qu'en début du processus (expression des besoins) et qu'à la fin du processus (validation).

Cycle de vie d'une application

Modèle en V



(référence : https://fr.wikipedia.org/wiki/Cycle_en_V)
CC BY-SA 4.0

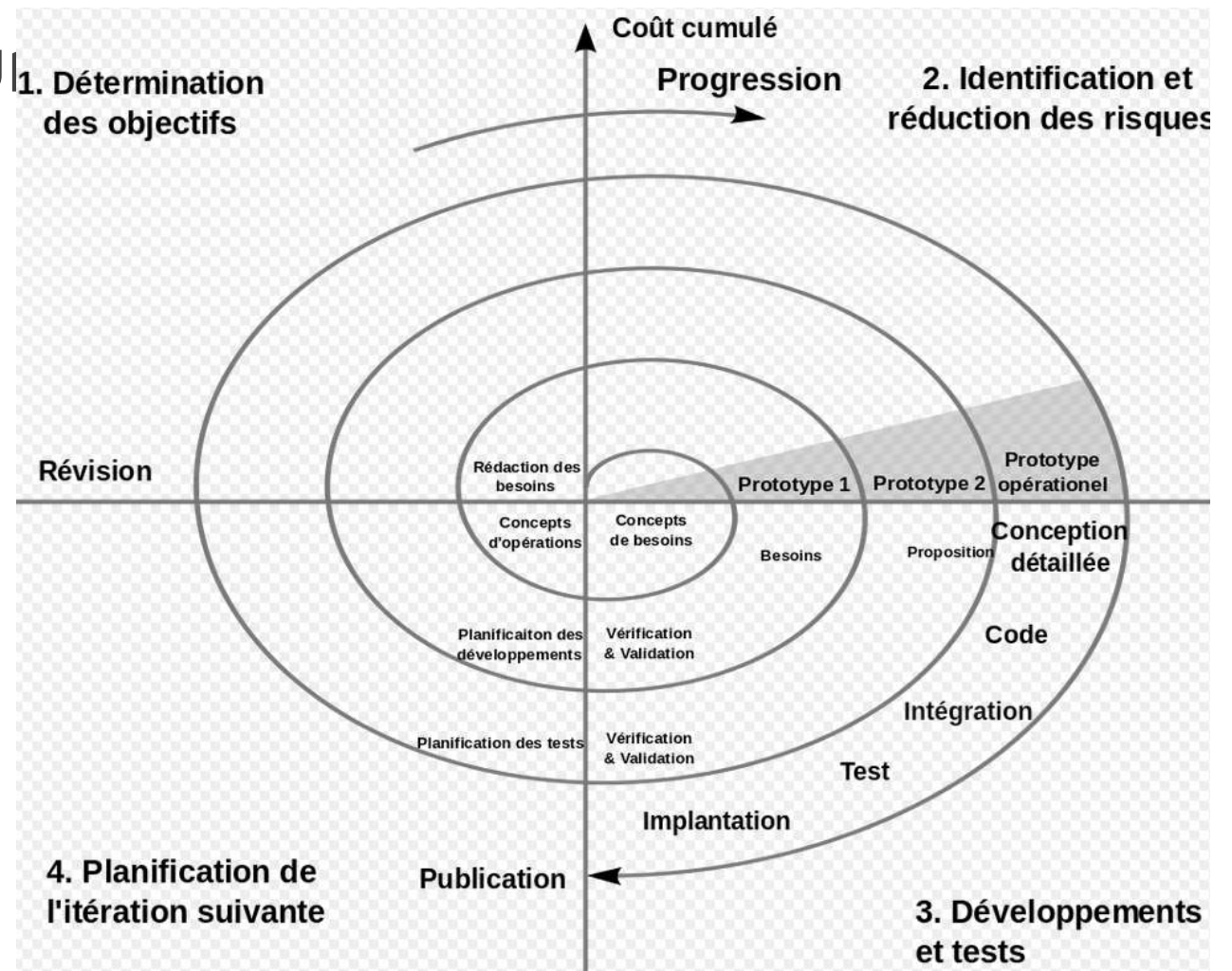
Cycle de vie d'une application

Modèle en V

Principe : Variante du modèle en cascade, qui met en relation les étapes de réalisation et les étapes de tests. Les tests sont conçus pendant les phases de réalisations et sont exécutés pendant les phases de tests.

- **“Tests unitaires”** : Test isolé de chaque composant de l'application.
- **“Tests d'intégrations”** : Test de bonnes liaisons entre les composants de l'application.
- **“Tests de validations”** ou **“Tests Système”**: Test de l'application complète par les concepteurs.
- **“Validation”** ou **“Recette”** : Test de l'application par le client

Cycle de vie d'un Modèle en Spirale



(référence : https://fr.wikipedia.org/wiki/Mod%C3%A8le_en_spirale)
CC BY-SA 3.0

Cycle de vie d'une application

Modèle en Spirale

Principe : Approche itérative où on répète des mini-processus de développements.

Avantage :

- Développer en premier les fonctions primordiales pour l'application et de les faire valider rapidement au client.
- Recueil des besoins continu et évolutif.
- Détection plus rapide des anomalies.

Risque :

- Nécessité à bien cadrer les besoins pour avoir des itérations cohérentes.

La méthode AGILE

<https://manifesteagile.fr/>

Principe : Valoriser

- Les individus et leurs interactions, de préférence aux processus et aux outils.
- Des solutions opérationnelles, de préférence à une documentation exhaustive.
- La collaboration avec les clients, de préférence aux négociations contractuelles.
- La réponse au changement, de préférence au respect d'un plan.

La méthode agile SCRUM

[SCRUM La méthode agile en 10 minutes \(Projet agile\) - YouTube](#)



Cette approche permet d'obtenir un cycle de vie de livraison des logiciels beaucoup plus flexible, collaboratif, efficace et itératif.