

CONCEPTION DU SYSTÈME D'INFORMATION

*Cycle de développement, étapes de construction,
processus, performance*

UFR IM²AG

**UNIVERSITÉ
Grenoble
Alpes**

Luc Laurens
Pr Christine Verdier

MIAGE RÉSEAU
DES MIAGE
DE FRANCE
Grenoble

- ☑ Pour concevoir un SI, on s'appuie sur trois grands concepts :
 - Des modèles : de données, de traitement, de fonctionnement → représentent ce qu'on comprend de l'entreprise.
 - Une méthode de conception : ensemble coordonné de règles opératoires qui permet de résoudre un problème en accord avec les concepts des modèles
 - Des outils logiciels : bibliothèques de composantes, frameworks de développement, outils d'intégration continue, etc.

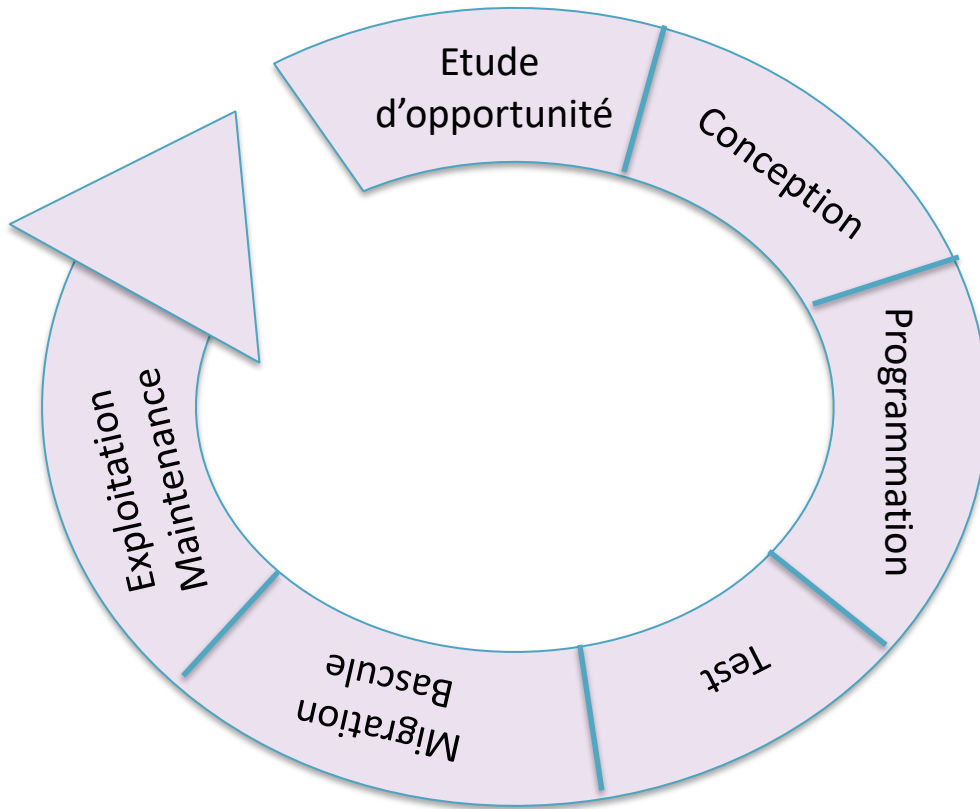
☑ Concevoir un SI :

- c'est utiliser une démarche bien gérée. Ce n'est pas un travail empirique, c'est un projet à construire et à piloter → cf. cycle de vie.
- C'est mettre en œuvre les ressources matérielles, logicielles et humaines selon une organisation bien définie

☑ La méthode répond aux questions : Que faire ? Quand ? Où ? Avec qui ? Comment ?

LES CYCLES DE VIE DU SI

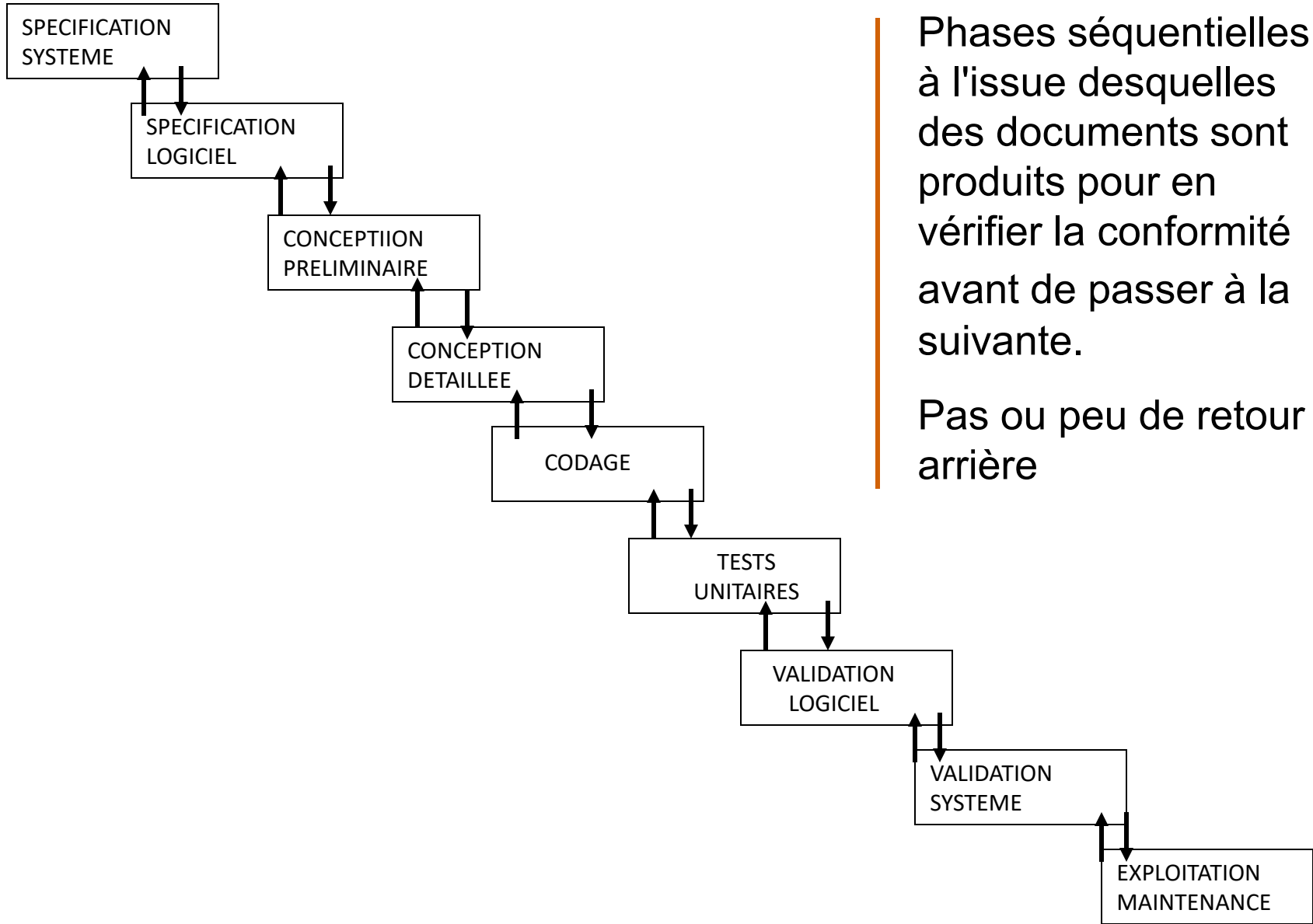
Les grandes phases



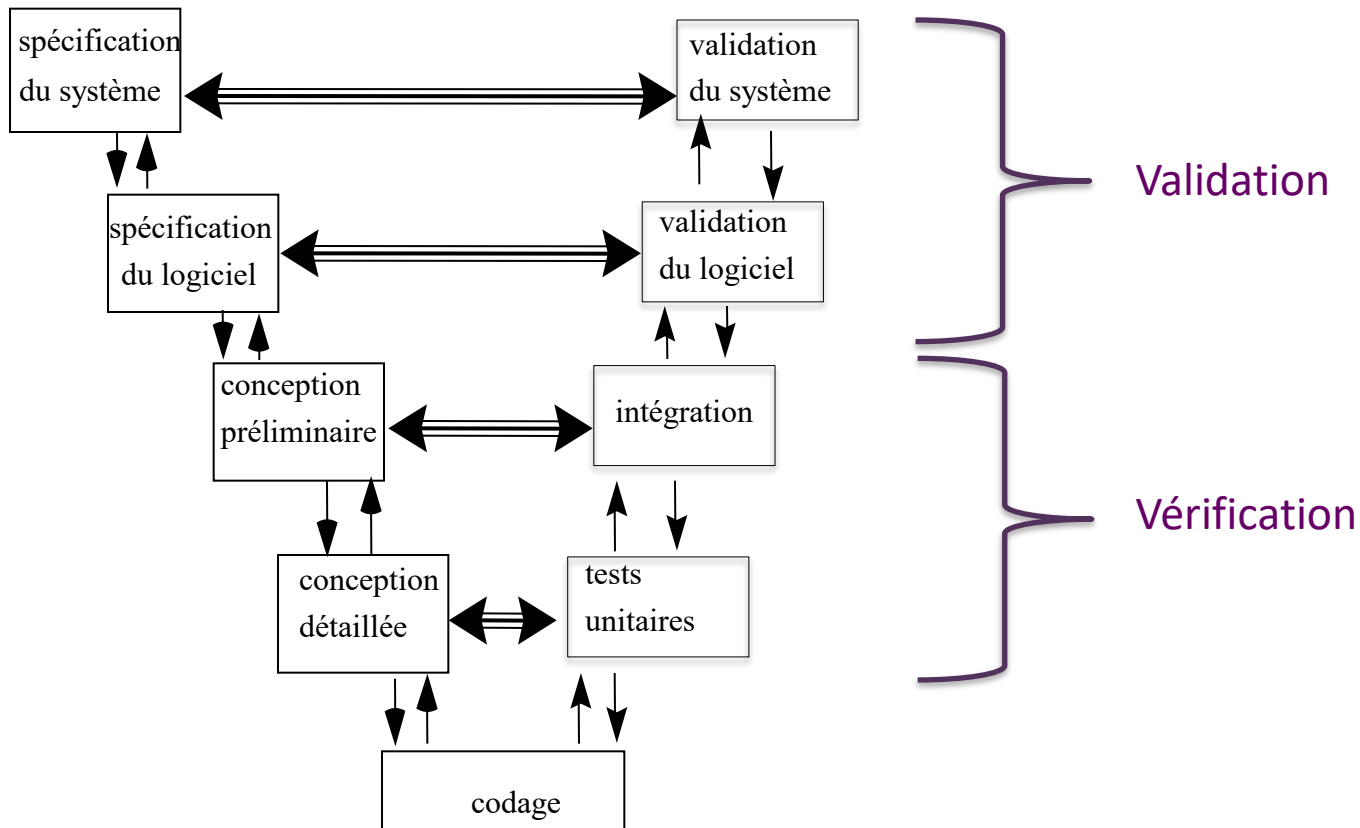
Ces grandes phases sont organisées différemment selon le cycle de développement choisi :

- cycle en V
- cycle en spirale
- cycle agile
- autres....

Cycle en cascade



Cycle en V



Des tests définis à l'issue de chaque phase

Cycle en V et en cascade : avantages vs inconvénients

☑ Avantages

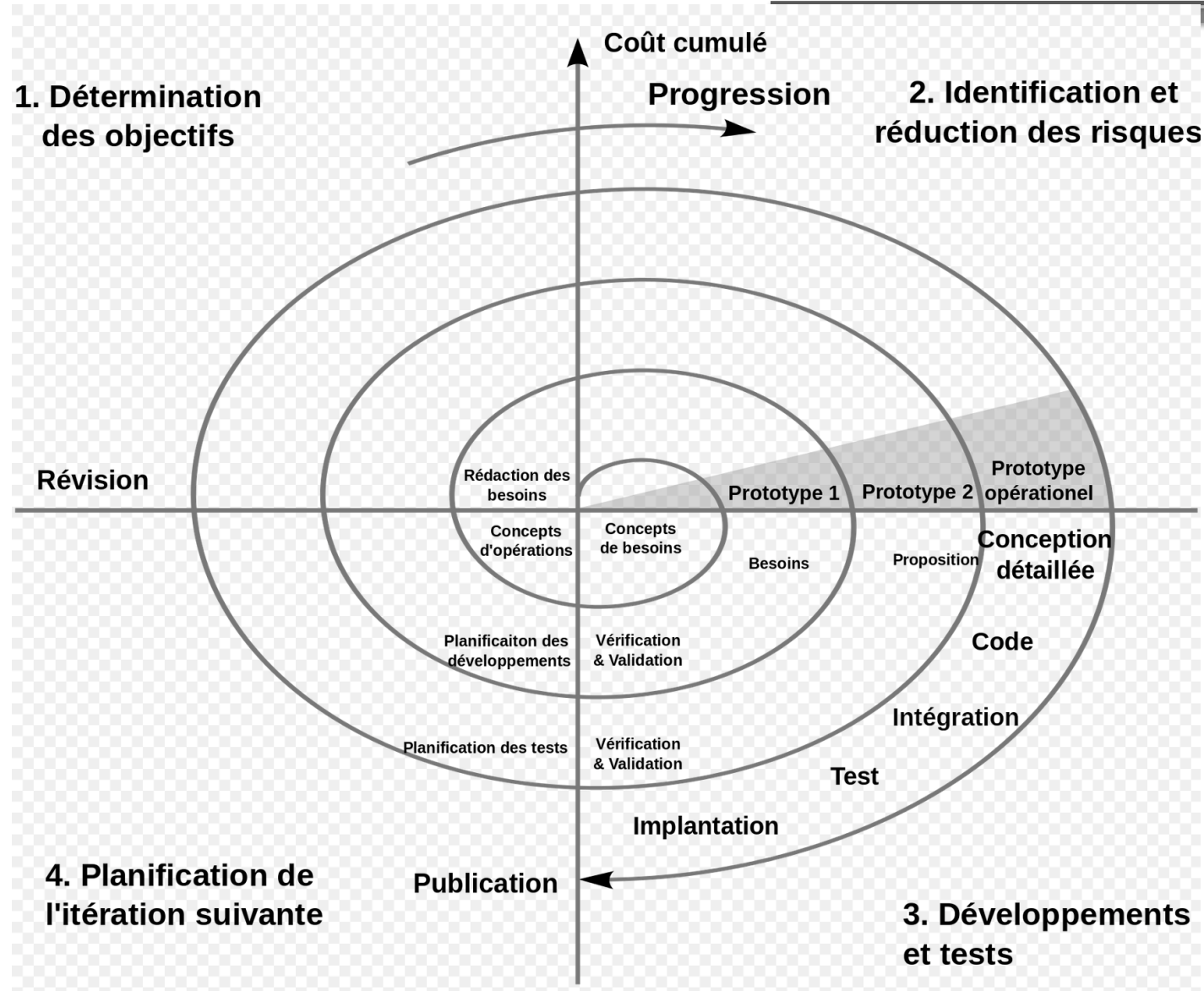
- Simple et facile à comprendre
- Force la documentation : une phase ne peut se terminer avant qu'un document soit validé
- Le test est inhérent à chaque phase
- Les progrès sont tangibles (pour l'équipe de développement)

Cycle en V et en cascade : avantages vs inconvénients

☑ Inconvénients

- Modèle dirigé par les documents.
 - Non compréhensibles par les clients
 - La première chose que voit le client est le produit final. Est-ce vraiment un problème ?
- Fait l'hypothèse de la faisabilité
 - Ne marche que si les exigences sont stables et le problème connu
 - Manque de flexibilité (ne traite pas les évolutions, notamment des exigences)
 - Problèmes peuvent être découverts en phase de validation
 - Irréaliste dans de nombreux cas

Cycle en spirale



analyse de risque ++

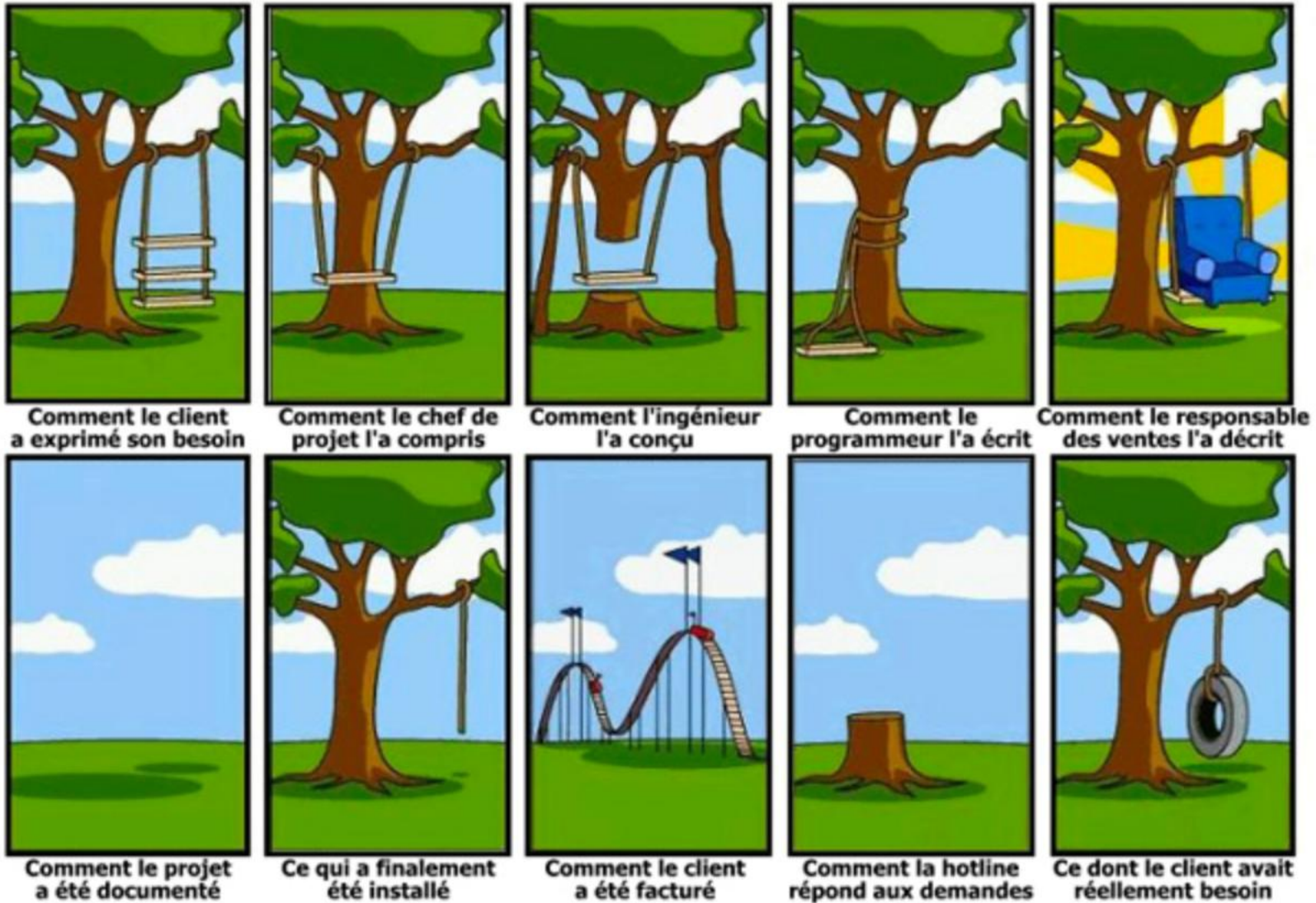
Cycle en spirale

- ☑ Le cycle de vie est représenté à l'aide d'une spirale
 - Chaque boucle représente une phase du développement
 - La boucle la plus interne traite des premières phases (faisabilité).
 - La plus externe traite de la livraison
 - Chaque boucle traverse quatre sections :
 - Définition des objectifs de la phase
 - Evaluation des risques et plan de gestion
 - Développement et validation
 - Planification de la phase suivante
 - Nombre de cycles variable

Cycle en spirale

- ☑ Le modèle en spirale est en fait un méta-modèle
- ☑ Il offre un cadre où chaque boucle doit être instanciée
- ☑ On peut par exemple créer :
 - Une boucle de faisabilité
 - Une boucle de prototypage
 - Des boucles de développement itératif, etc.
- ☑ Il faut alors trouver le bon modèle de processus pour chaque boucle !

Problèmes soulevés par ces cycles de dév



☑ Problème soulevé

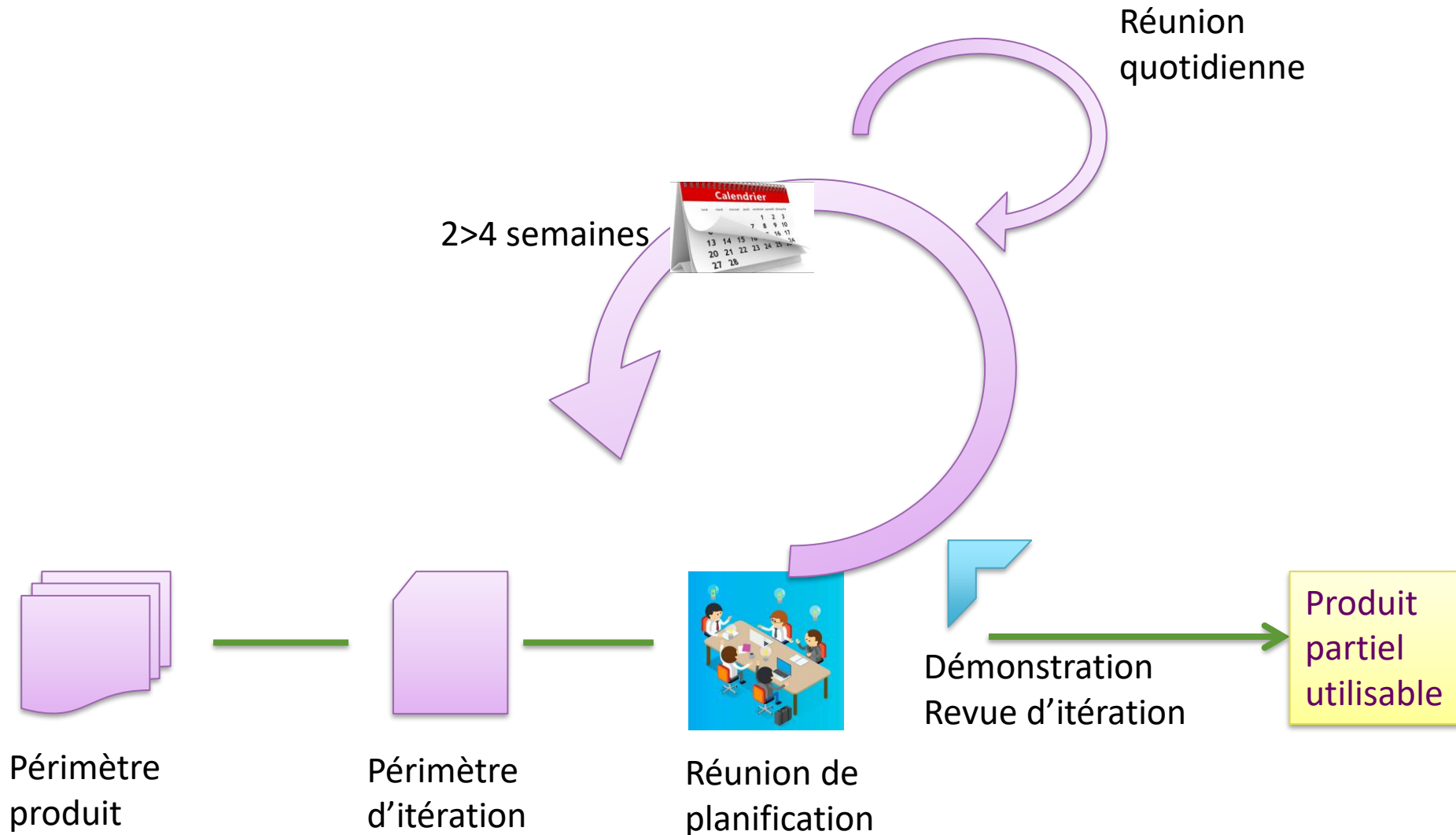
- Recueil des besoins : tâche la plus compliquée
- Constat :
 - La plupart des défauts d'une application est due à une mauvaise définition du besoin
 - 50% des fonctionnalités sont peu ou pas utilisées
- Pourquoi ?
 - Difficulté à exprimer le besoin **clairement** et de manière **exhaustive**
 - Dans une démarche classique :
 - Membres de l'équipe n'interviennent que dans les étapes les concernant
 - Transmission d'informations par documentation : mauvaise interprétation ? Incompréhension ?
 - Levée tardive des facteurs de risque.

- ✓ En 2001, aux EU, 17 experts des méthodes Agile ont créé le Manifeste Agile.
- ✓ Quatre valeurs :
 - Interactions avec les personnes plutôt que les processus et les outils
 - Produit opérationnel plutôt qu'une documentation pléthorique
 - Collaboration avec le client plutôt que négociation par contrat
 - Réactivité face au changement plutôt que suivi d'un plan.

Quelques méthodes agiles

- ☑ L'eXtrême Programming (XP)
 - Accélération des dev, travail en binôme, cycle de dev très courts
- ☑ Scrum (= mêlée)
 - Adapté à la gestion de projets informatiques
 - Modifier la direction prise par le projet au fur et à mesure de son avancement
- ☑ Feature Driven Development (FDD)
 - méthode de gestion de projet basée sur la gestion des risques, tests utilisateurs sur chaque fonctionnalité
- ☑ Lean Software Development
 - Qualité au cœur de la gestion du projet, en optimisant notamment l'ensemble des processus d'apprentissage, de décision, de livraison et de mesure de performances
- ☑ Agile Unified Process (Agile UP ou AUP)
 - Version simplifiée du RUP (Rational UP) : dev piloté par les tests et par les modèles

Cycle agile

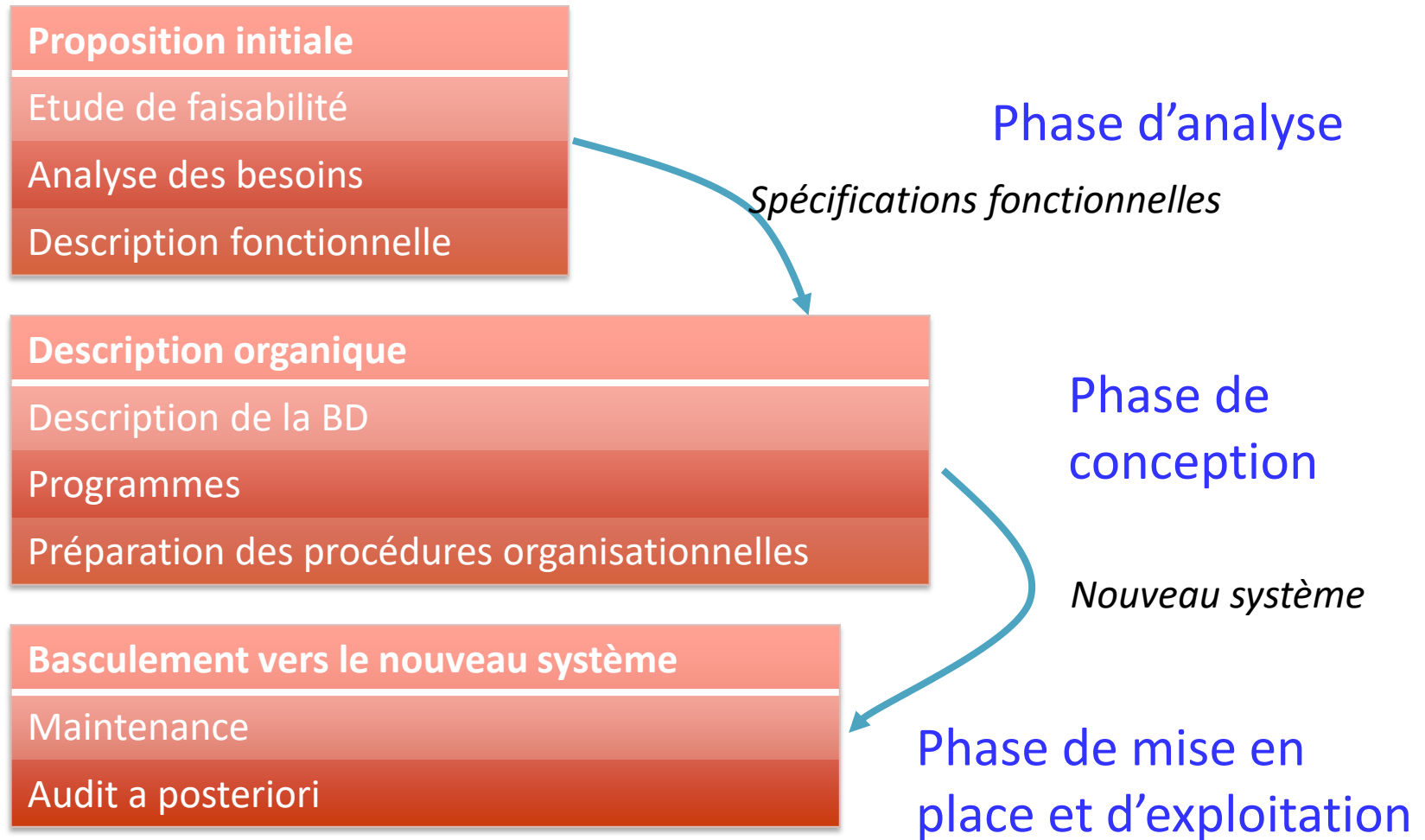


Cycle agile

- ☑ Chaque itération est un mini-projet qui produit une version intermédiaire utilisable du produit final
- ☑ Les itérations se succèdent, pas de parallélisation possible
- ☑ Chaque itération a une durée fixe (mécanisme de timeboxing) : mobilisation des efforts sur des objectifs courts et clairs.
- ☑ On supprime le cahier des charges pour le prototypage : attention à ne pas simplement changer de support !!

LES ÉTAPES DE CONSTRUCTION

Conception du SI



Phase d'analyse

- ☑ Analyse du problème que l'entreprise veut résoudre à l'aide d'un SI. Est réalisée par un analyste.
- ☑ Comprend : la définition du problème, de ses causes, de la solution à mettre en place et des besoins en information.
- ☑ Difficulté : définir les besoins en informations → besoins collectifs vs besoins individuels
- ☑ Informations : structuration, traitements, modalités saisie, etc.
- ☑ Produit une étude de faisabilité (solution réalisable ? Bon investissement ? Avantages vs inconvénients)

Décision sur le développement

- ☑ Plusieurs choix : solution entièrement développée en interne, entièrement confiée à une ESN, solutions mixtes, pas de développement interne mais intégration de solutions.
- ☑ Questions : environnement de développement, outils utilisés, cohérence globale de ces outils (éditeurs différents), paramétrage ?

Conception et réalisation

- ☑ Base de conception = Spécifications fonctionnelles et techniques
 - Entrées et sorties des composants
 - Interface utilisateur
 - Conception des modèles : organisationnels, statiques et dynamiques + modalités de traitement de l'information
 - Procédures : activités, ressources humaines, méthodes
 - Processus métiers
 - Modalités de contrôle : E/S, traitements, procédures
 - Sécurité
 - Documentation ? (selon cycle adopté)

- ☑ Plusieurs choix : solution entièrement développée en interne, entièrement confiée à une ESN, solutions mixtes, pas de développement interne mais intégration de solutions.
- ☑ Questions : environnement de développement, outils utilisés, cohérence globale de ces outils (éditeurs différents), paramétrage ?

Mise en place et exploitation

- ✓ Tests
- ✓ Migration-bascule
- ✓ Maintenance
- ✓ Formation des collaborateurs,
accompagnement au changement : à ne pas
négliger.

- ☑ Un *test* est un ensemble de *cas à tester* (état de l'objet à tester avant exécution du test, actions ou données en entrée, valeurs ou observations attendues, et état de l'objet après exécution), éventuellement accompagné d'une *procédure d'exécution* (séquence d'actions à exécuter). Il est lié à un objectif / *Wikipédia*/
- ☑ Organisation tests : planification, conception, exécution, clôture, suivi.
- ☑ Classification : tests unitaires (des composants), tests d'intégration technique, tests système (d'intégration fonctionnelle), tests d'acceptation (recette). Complété par des tests de non-régression.

Migration-bascule

- ☑ Processus de passage de l'ancien système au nouveau.
 - Stratégie en doublon : exécution des 2 systèmes en parallèle
 - Stratégie du basculement direct : suppose de conserver un « back-up » de l'ancien SI
 - Stratégie de projet pilote : nouveau SI dans un secteur limité → test et si fonctionne → déploiement dans le reste de l'entreprise
 - Stratégie de migration-bascule par étape : introduction du nouveau système graduellement (fonction par fonction par exemple).

Exploitation et maintenance

- ☑ La mise en exploitation correspond à la fin de la migration.
- ☑ Une évaluation régulière sera nécessaire.
- ☑ Maintenance : correspond dans 60% des cas à apporter des améliorations du point de vue des utilisateurs.
- ☑ Pour éviter une maintenance trop coûteuse, il faut soigner l'analyse préalable et la conception.

Attention, créer un SI ne crée pas les usages qui vont avec !!!

✓ Le SI doit :

- s'adapter aux utilisateurs (et non l'inverse)
 - Attention ! Plusieurs utilisateurs ayant des vues différentes sur le SI
- Être modulable et évolutif rapidement
- Être adapté aux nouvelles technologies
- Répondre le plus possible aux besoins

✓ Pour cela, il doit :

- Être co-construit avec les utilisateurs
 - Voire... leur donner les outils permettant de concevoir la partie les concernant
-