

CONDUITE DE PROJET LOGICIEL

**MODELES ET
PROCESSUS DE
DEVELOPPEMENT**

Norme ISO 12207: Définitions

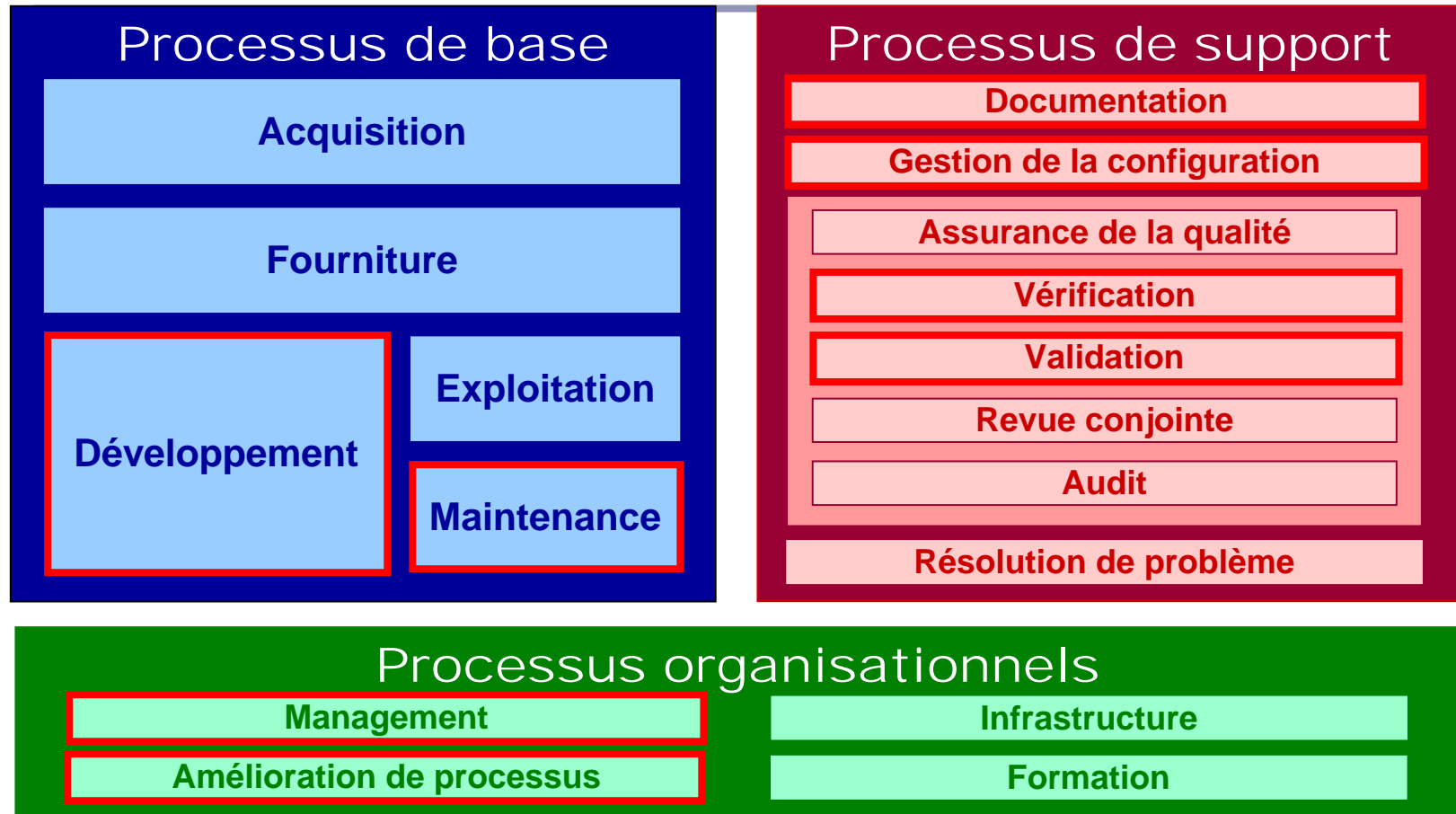
Modèle de cycle de vie du logiciel:

Scénario contenant les processus, les activités et les tâches mis en œuvre pour le développement, l'exploitation et la maintenance d'un logiciel, englobant la totalité de la vie du système depuis l'expression de besoins jusqu'à la fin de son exploitation.

Processus:

Ensemble d'activités qui transforment les éléments entrants en éléments sortants.

Norme ISO 12207: Processus du cycle de vie du logiciel



Vocabulaire

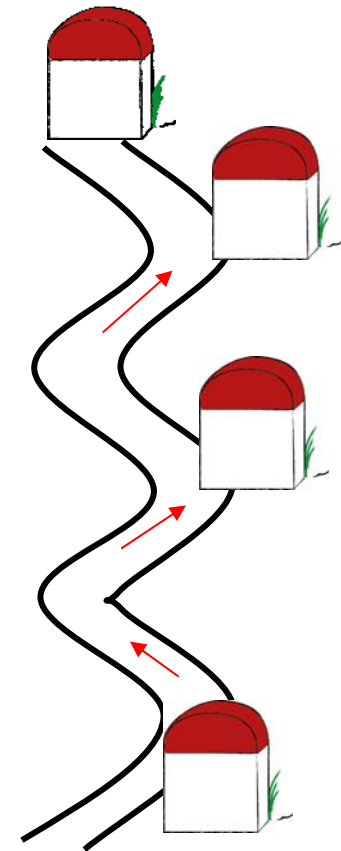
- Le projet repose sur l'exécution de différents processus
- Un processus peut se découper en phases successives, chaque phase correspondant à
 - Un état d'avancement général du projet
 - Un objectif intermédiaire, une étape
 - Une activité dominante
- Pour atteindre un objectif et franchir une phase, différentes tâches doivent être réalisées
- La réalisation d'une tâche nécessite des ressources
- Les tâches peuvent être ponctuées par des livrables

Jalon

- Un jalón (ou, en anglais milestone) correspond à une *étape*, un point de *rendez-vous*, *l'atteinte d'un objectif intermédiaire*, un *événement important* dans le projet.

Exemples: Signature du contrat, fin de définition, fin de conception, entrée en validation, début de déploiement ...

- Franchir un jalón est obligatoire pour passer à la phase suivante.



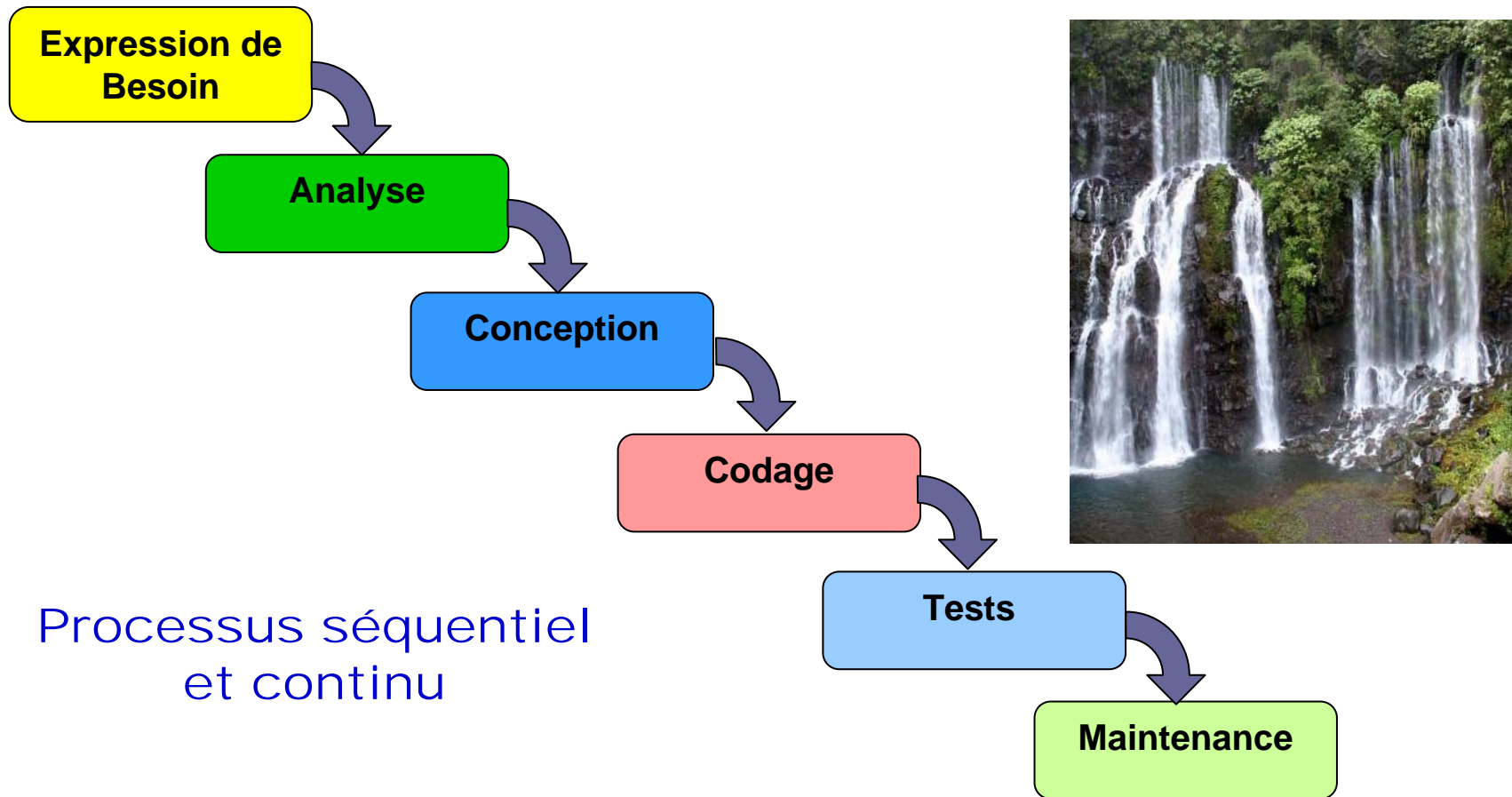
Modèles et processus de développement connus

- MODELE DE LA CASCADE
- CYCLE EN «V»
- CYCLE A PROCESSUS INDEPENDANTS
- MAQUETTAGE / PROTOTYPAGE

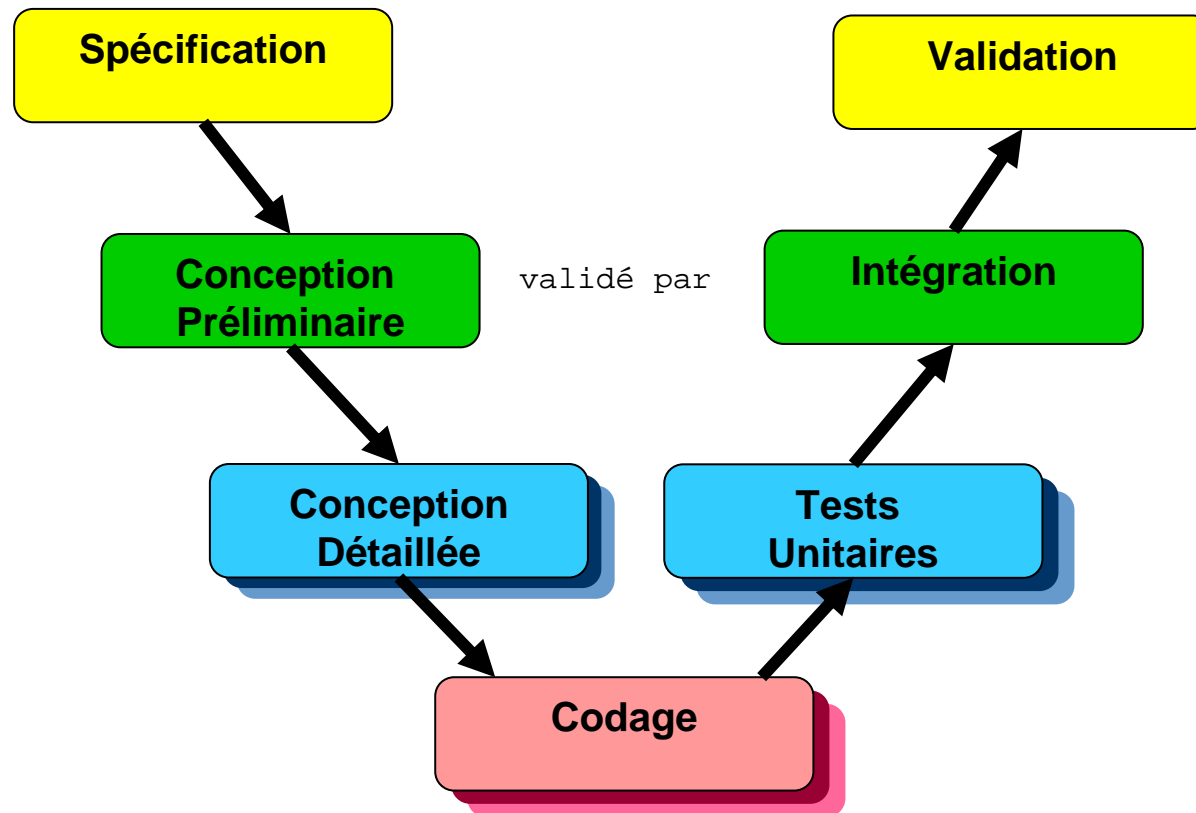
- MODELE DE LA SPIRALE
- PROCESSUS UNIFIE
- RAPID APPLICATION DEVELOPMENT (RAD)
- METHODES AGILES

- CYCLE EN W
- CYCLE EN X
- CYCLE EN Y

Modèle de la cascade

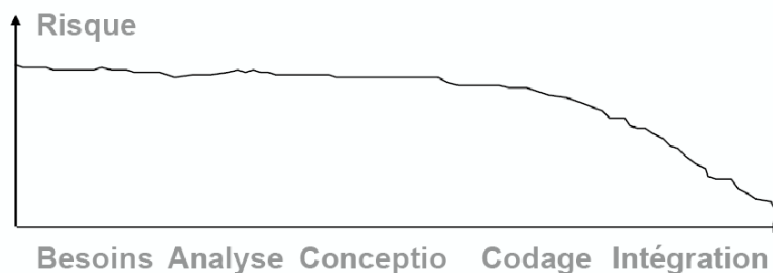


Modèle Général en V



Limites du modèle de la cascade

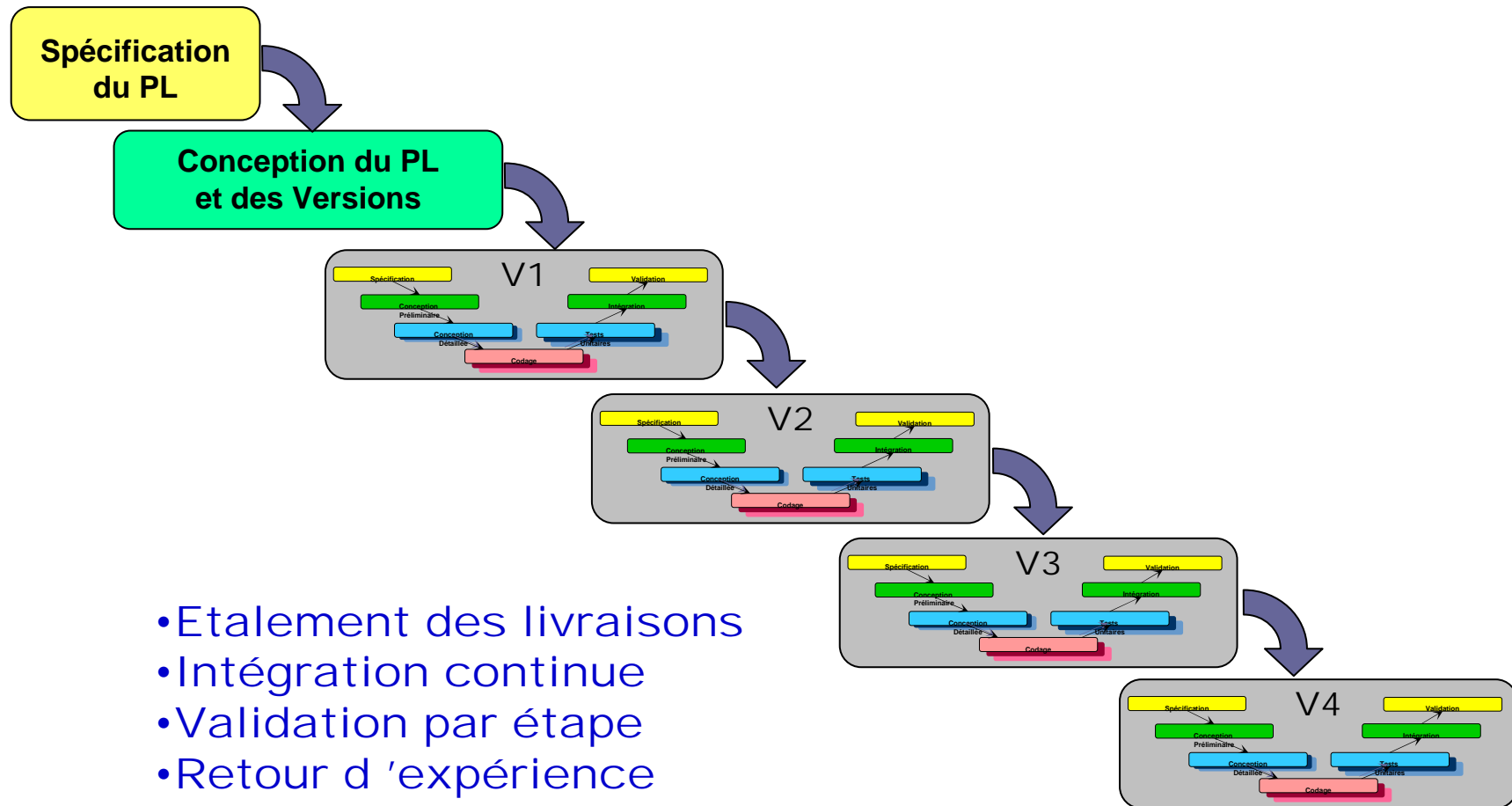
- Le produit n'est visible qu'à l'issue du développement
- Les choix techniques ne sont validés qu'à l'issue de la réalisation
- Les jalons sont basés sur des livraisons documentaires
- Le début d'une phase est conditionné par la terminaison de la phase précédente
- La découverte d'une erreur peut nécessiter un retour en arrière



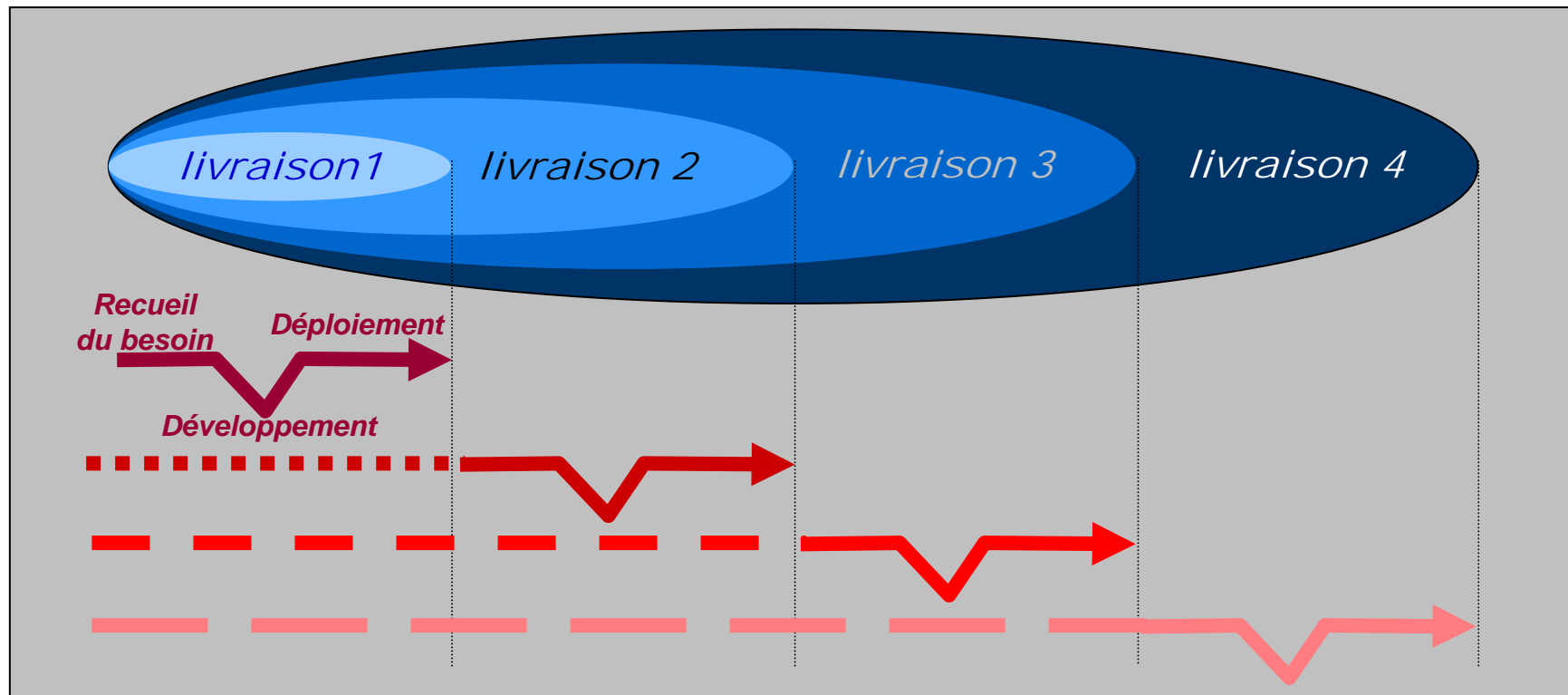
Maquettage et prototypage

- Développement « en avance de phase » d'une partie du produit ou d'un logiciel représentatif de celui-ci
 - Maquette = Logiciel spécifique et « jetable » destiné à appréhender, de façon précoce, le produit final ou l'une de ses caractéristiques (peut être développé avec des outils spécifiques et pour une autre cible)
 - Prototype = Partie ou version préliminaire (réutilisable) du produit final développée, de façon précoce, pour valider une hypothèse technique ou assurer le respect des exigences.
- Activités pouvant être intégrées dans n'importe quelle phase (Micro-processus)

Processus itératif



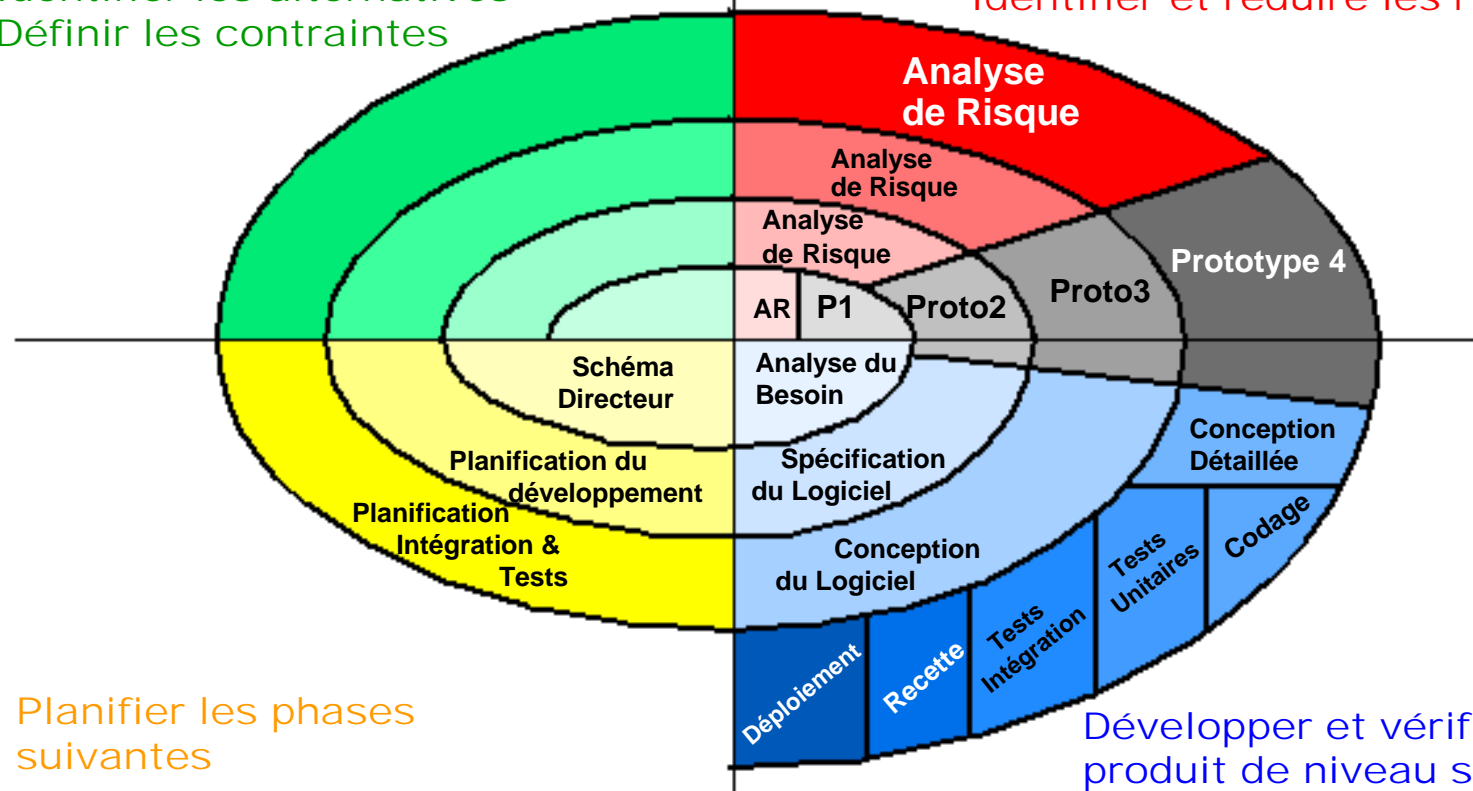
Itérations et incréments



Modèle en spirale

Déterminer les objectifs
Identifier les alternatives
Définir les contraintes

Evaluer les Alternatives
Identifier et réduire les risques



Trois caractéristiques majeures d'une approche incrémentale

- **INTÉGRATION CONTINUE**

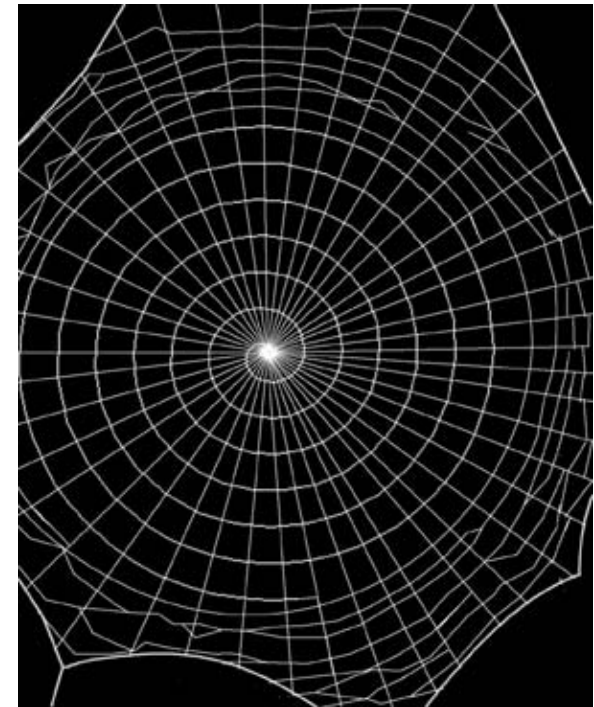
- Pas d'intégration big-bang quelques jours avant la livraison

- **PRODUCTION FRÉQUENTES DE VERSIONS**

- Certaines pour évaluation interne d'autres livrées au client

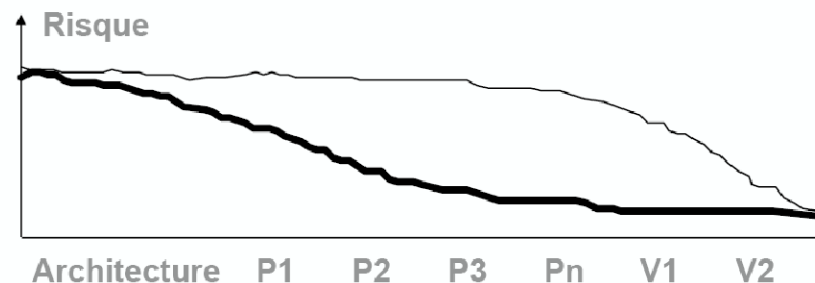
- **MAÎTRISE DES RISQUES ET AVANCEMENT DÉMONSTRABLE**

- L'avancement du développement se mesure à travers le logiciel lui-même et non pas à travers la documentation ou des évaluations des ingénieurs.



Effets et conséquences

- La production de versions oblige l'équipe de développement à stabiliser le logiciel à intervalles réguliers
- Peut permettre d'intégrer les évolutions et corrections d'anomalies dans l'incrément suivant sans risquer de rompre la continuité du développement
- Permet de mieux distribuer la charge sur les ressources utilisées par le projet (développeurs, testeurs, outils, Gestion de configuration, Assurance Qualité, etc.)



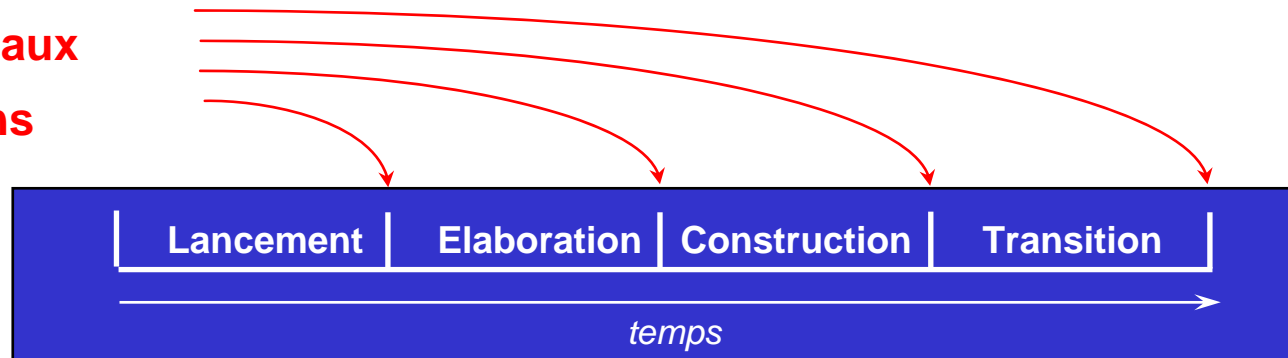
Processus Unifié (UP)

Synthèse des meilleures pratiques de génie logiciel

- Développement itératif
- Traitement et formalisation des exigences
- Architecture à base de composants
- Modélisation visuelle
- Vérification de la qualité du processus et du produit
- Gestion des changements

Processus Unifié (UP)

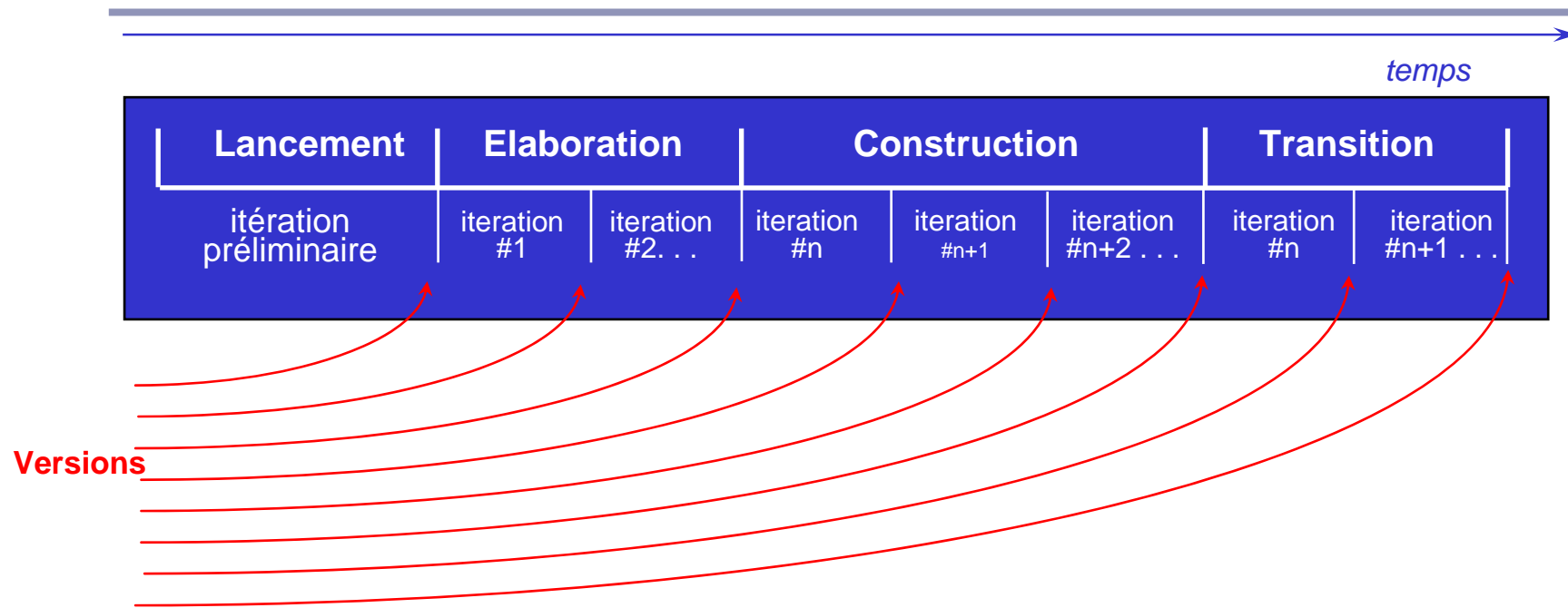
Principaux
Jalons



Unified Process = 4 phases:

- **Création/Lancement** — Définir les objectifs du projet
- **Elaboration** — Planification, spécification des exigences et conception d'architecture
- **Construction** — Développement du produit
- **Transition** — Transfert du produit dans la communauté des utilisateurs

Processus incrémental et itératif



Une itération est un cycle de développement complet se terminant par une version exécutable qui correspond à un sous-ensemble du produit à développer.

Distribution des activités

Modélisation Métier

Spécification

Analyse et conception

Réalisation

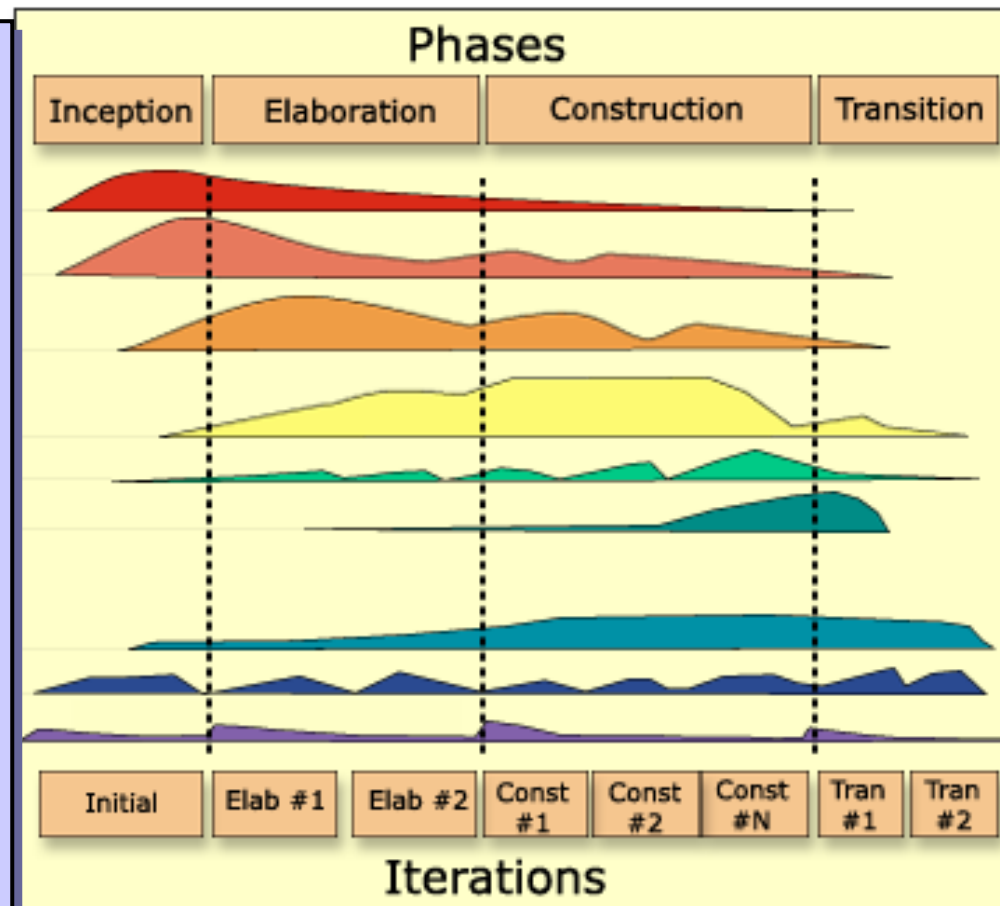
Tests

Déploiement

Gestion de la configuration
et du changement

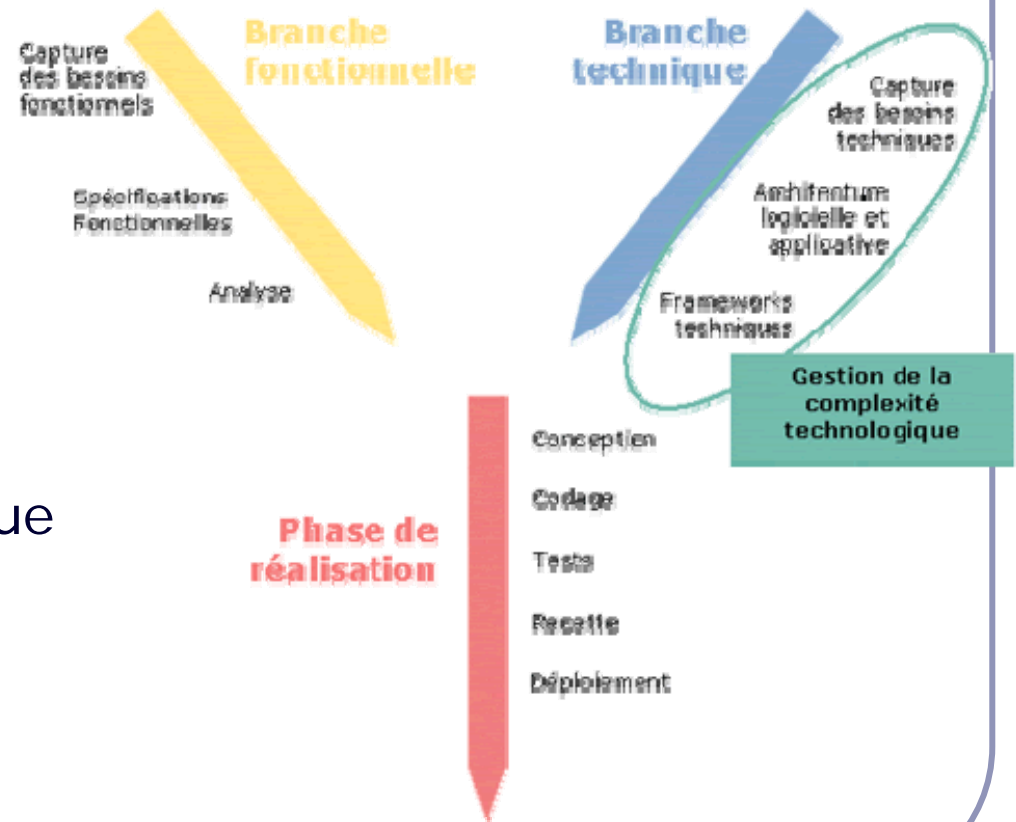
Conduite de projet

Environnement



2TUP: Two Track Unified Process

- Dissocie les aspects techniques des aspects fonctionnels.
- 3 axes de travail
 - une branche technique
 - une branche fonctionnelle
 - une phase de réalisation



Agilité

AGILITE: Aptitude à s'adapter pour une entreprise ou une organisation

Méthodes agiles: Ensemble de principes et de valeurs pour le développement de logiciel basé sur des techniques de production légères pour produire de la valeur rapidement et fréquemment



Bref historique

L'agilité est à la mode, mais les méthodes de développement logiciel dites « itératives et incrémentales » existent depuis longtemps:

- **1986 : Modèle de la spirale de B.W. Boehm**
- **1991: Méthode RAD de James Martin**
- **1995: SCRUM**
- **1996: eXtrem Programming.**
- **2011: Manifeste Agile = Réunion de 17 figures éminentes du développement logiciel**

"We are uncovering better ways of developing software by doing it and helping others do it. These are our values and principles."

Processus agile

En plus d'être itératif et incrémental, un processus agile est caractérisé par

- des itérations courtes et régulières
- un client qui fait partie de l'équipe
- une capacité au changement
- une forte implication des membres de l'équipe dans la conduite du projet
- des pratiques d'ingénierie permettant de garantir la qualité du code

→ Prochain chapitre du cours

Planification d' une itération

- Avant de commencer la nouvelle itération, les objectifs généraux doivent être établis
 - en tenant compte des résultats des précédentes itérations
 - après avoir ré-actualisé l'évaluation des risques
- Des critères objectifs doivent être définis pour évaluer le résultat de l'itération
- Le planning détaillé de l'itération doit être élaboré puis intégré dans le planning général
 - en insérant des jalons pour mesurer l'avancement
 - en prévoyant des inspections et des revues
- Le travail à réaliser pendant l'itération se détermine en fonction
 - des (nouvelles) fonctionnalités à implémenter
 - de la reprise des fonctionnalités existantes

Bilan d' itération

- **Evaluer les résultats de l'itération en regard des objectifs fixés lors de la phase de planification**
 - Fonctionnalités
 - Performance
 - Capacité
 - Mesures de qualité
- **Identifier et analyser les évolutions externes du produit qui sont intervenues pendant l'itération**
 - Par exemple, modifications de spécification, nouveaux besoins du client, progrès des concurrents, etc.
- **Identifier ce qui doit être repris et/ou corrigé, ce qui doit évoluer et l'allouer à une future itération**

Définition des itérations

- **Combien d 'itération sont nécessaires?**
 - Pour un projet de 18 mois ou moins, 3 à 6 itérations sont généralement planifiées
- **Est-ce que les durées des itérations sont identiques?**
 - C 'est souhaitable pour réguler la charge
 - La durée des itérations peut varier en fonction de la phase. Par exemple, les itérations d élaboration peuvent être plus courtes que les itérations de construction

La première itération

- **Généralement la plus difficile**

- car elle nécessite
 - la mise à disposition de l'intégralité de l'environnement de développement
 - la présence de la majeure partie de l'équipe
- et elle est souvent marquée par
 - de nombreux problèmes d'intégration d'outils,
 - des difficultés de montée en charge,
 - des erreurs de débutants , etc.

- **Les équipes non aguerries à une approche de développement itératif sont généralement trop optimistes ...**

- ⇒ Ne pas être trop ambitieux en ce qui concerne les fonctionnalités à délivrer à l'issue de la première itération
- Faute de quoi,
 - la finalisation de la première itération devra être différée,
 - le nombre total d'itérations sera réduit, et
 - les effets bénéfiques de l'approche itérative seront limités

Difficultés d' un processus itératif et incrémental

Un développement ITERATIF et INCREMENTAL

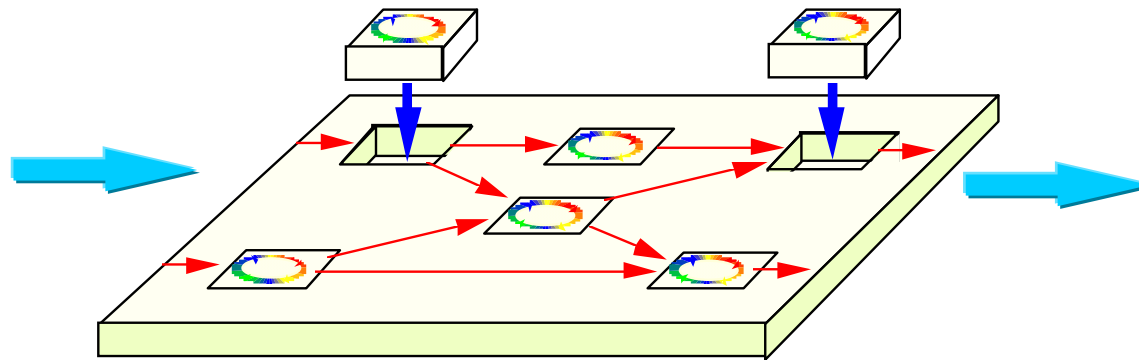
- nécessite l'implication et la participation du client, du management et des développeurs
- ne justifie pas l'absence de planning et de suivi d'avancement
- peut engendrer des régressions ou des remises en cause techniques en cours de développement
- ne doit pas consister à faire et refaire la conception jusqu'à l'atteinte de la perfection
- laisse place aux opportunités mais ne doit pas devenir un développement imprévisible

Objectifs d' un processus itératif et incrémental

Un développement ITERATIF doit

- s'appuyer sur un processus planifié et maîtrisé
- être découpé en une suite d'étapes prévisibles
- offrir la possibilité de prendre en compte les évolutions de spécification
- être balisé par des jalons formels et des livraisons intermédiaires (pas uniquement documentaires)
- impliquer le client/utilisateur durant toute la durée du développement
- être conduit par la gestion des risques

Il n' existe pas de processus universel !



- **Pour chaque projet un processus spécifique doit être établi (sur la base d'un modèle générique)**
- **Chaque itération doit être définie et planifiée en fonction du plan de versions et des risques identifiés**