



Gestion Projet

Cours 3

Le cycle de vie



Sommaire

■ Généralités	3
■ Séquentiel	7
■ Itératif/Incrémental	17
■ Extreme Programming	22
■ Que choisir ?	29
■ Etats Transverse	33



Généralités

Notion et but

- But :
 - Maîtriser les risques
 - Maîtriser les délais, les coûts
 - Maîtriser la qualité

- Qu'est qu'un cycle de vie
 - Découpage en un ensemble de processus
 - Un processus : objectif, étapes, entrée/sortie

Les phases classiques (1)

- Analyse et spécification
 - Description du dossier qualité et du cahier des charges
- Conception
 - Définition de l'architecture du logiciel (globale)
 - Découpage en modules (détaillée)
- Réalisation
 - les modules codés
 - Tests unitaires

Les phases classiques (2)

- Tests
 - intégration
 - qualification technique et fonctionnelle
- Mise en production et maintenance
 - correction des anomalies résiduelles
 - vérification du respect du contrat de service

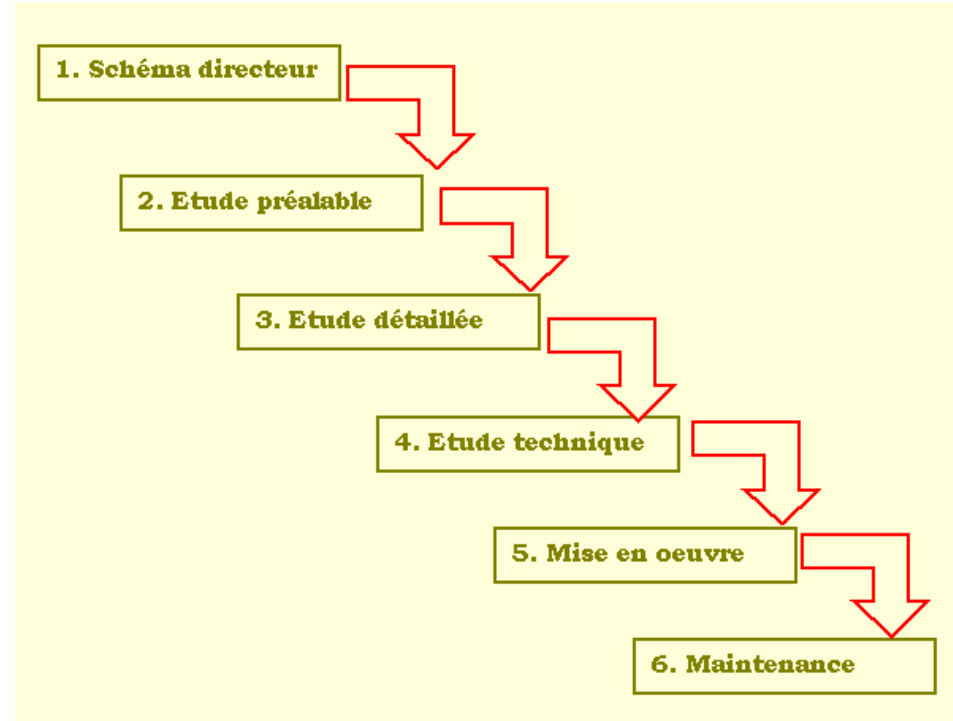
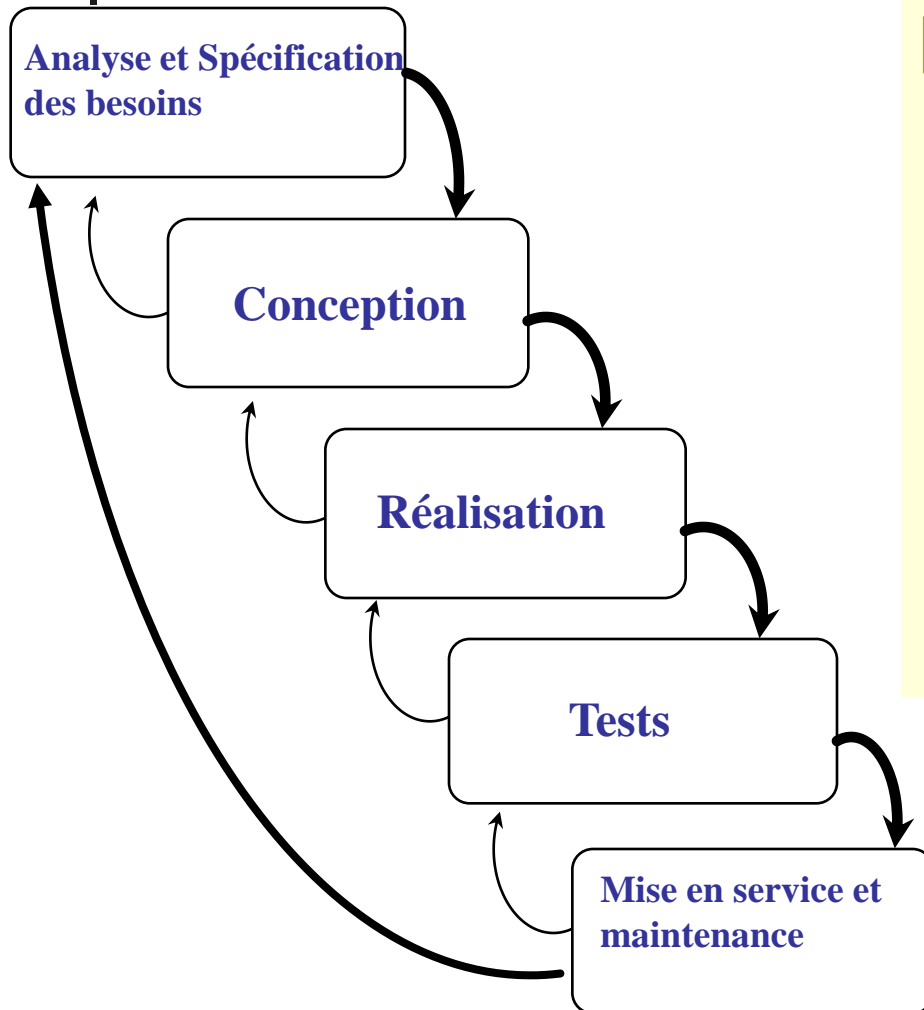


Séquentiel

Le cycle en cascade (1)

- Cycle classique (Bohem 1988)
 - Base et référence des autres modèles
- Développement modulaire
- Une étape doit être finie pour que la suivante commence
- Chaque phase se termine à une date précise
- A l'origine pas de retour arrière
 - Modification : une étape ne remet en cause que l'étape précédente

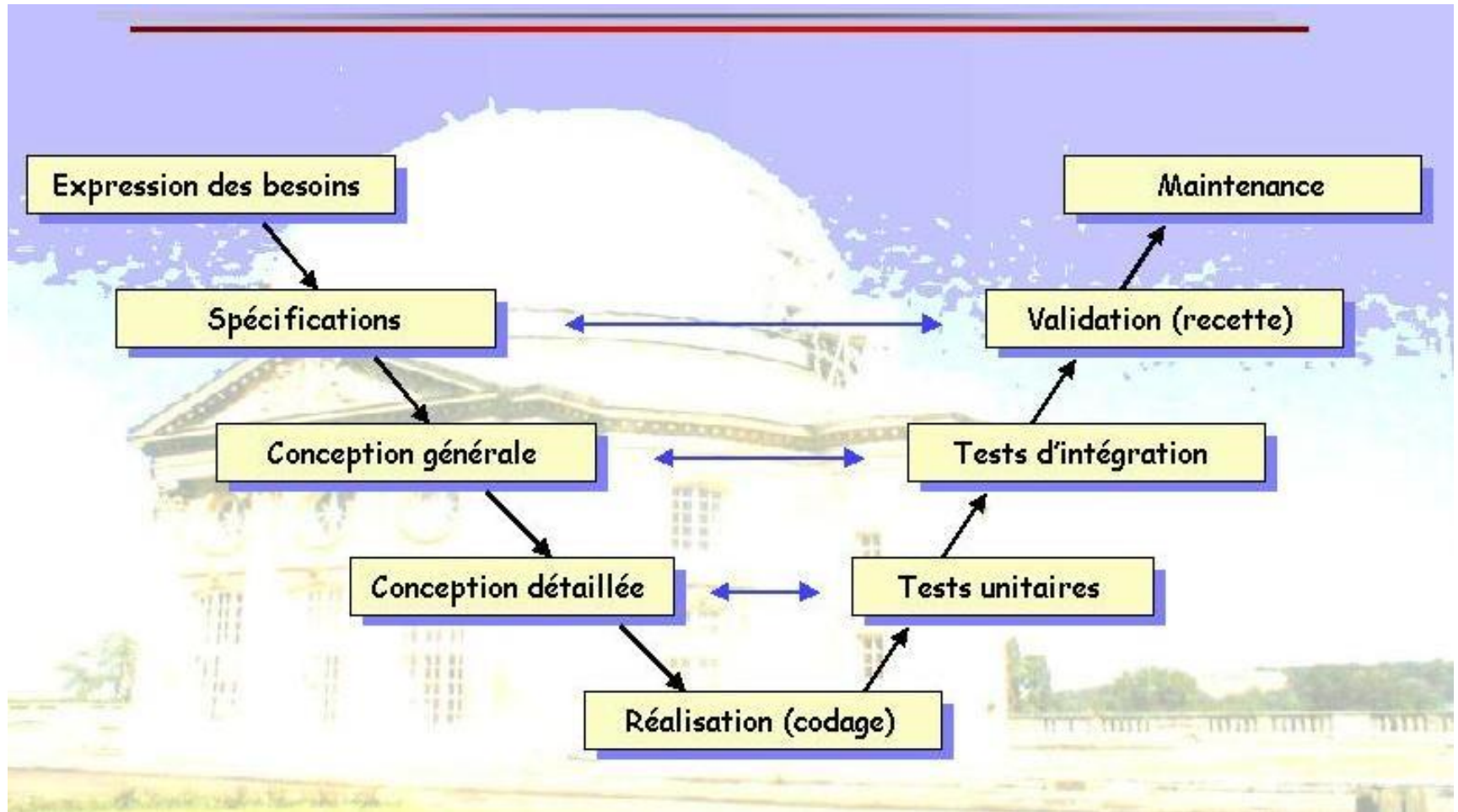
Le cycle en cascade (2)



Le cycle en V (1)

- Evolution du cycle en cascade
- Mise en évidence des phases de vérification
 - plier la ligne afin de différencier les phases de production et de validation
- Faire le parallèle entre les phases de conception/réalisation et les phases de tests
 - une erreur découverte lors d'une phase de test remet en cause la phase de conception associée

Le cycle en V (2)



Cycle en V (3)

EUPHORIE

CONDUITE DE PROJET

PROMOTION DE
CEUX QUI N'ONT
PAS TREMPÉ
DANS LE PROJET

INQUIETUDE

PANIQUE

PUNITION
DES
INNOCENTS

RECHERCHE
DES
COUPABLES

1

2

3

4

5

Cahier des
charges

Etudes

Développement

Tests

Recette

Production

t

Avantages

- Permet de discipliner le processus de développement
- Fait apparaître des points de contrôle
- Notion de validation forte (cycle en V)
- Base de tout cycle

Inconvénients

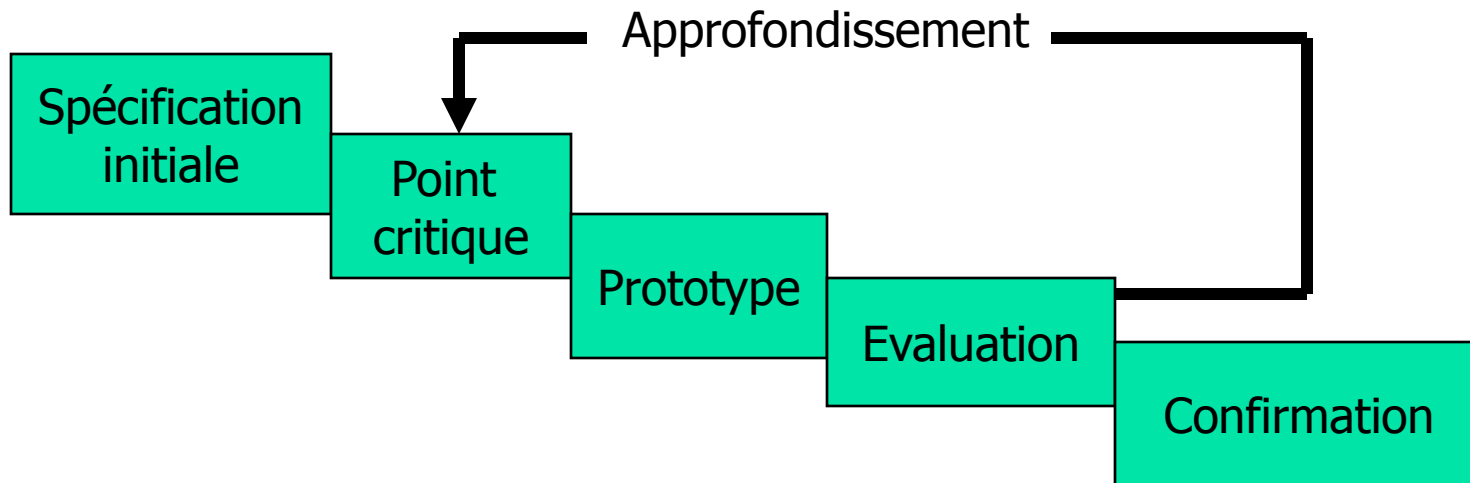
- Rigidité des phases « linéaires »
- «industrie manufacturière » <> créativité
- Projet monolithique : un produit global livré en bloc
- Estimation du coût difficile à faire
- Difficulté d'intégrer et d'anticiper le changement

Prototypage/maquettage

- Utilisé lors des phases amonts du projet
 - Analyse des besoins, Spécifications fonctionnelles
- Validation par expérimentation
- Phase rapide
- S'intègre dans un cycle en cascade ou en V

Prototypage expérimental

- Utilisé au niveau de la conception pour :
 - s'assurer de la faisabilité de parties critiques
 - valider des options de conception
- Souvent jeté après développement





Itératif/Incrémental

Prototypage évolutif

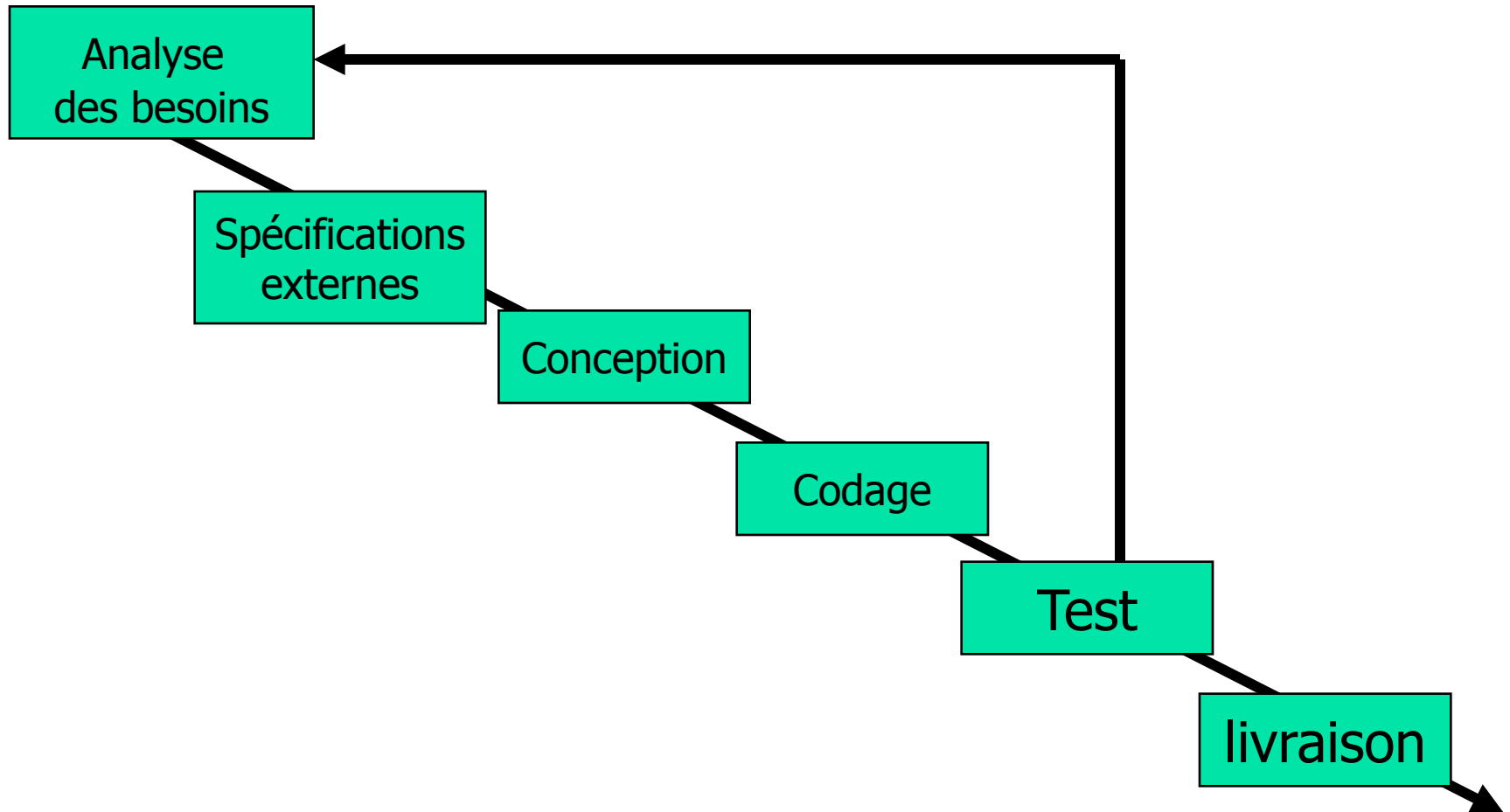
- La première version du prototype: embryon
- Développement par incrément
 - chaque prototype fait l'objet d'un cycle spécification-conception-implantation-test.
- Mise en place de la notion de cycle incrémental



Modèle en spirale

- Basé sur le prototypage évolutif
- Basé sur la gestion des risques
- Démarche :
 - Identifier les risques et commencer par les plus importants
 - Chaque cycle peut est basé sur une démarche en V ou en cascade

Modèle en spirale (2)





Avantages/inconvénients

- Avantages :
 - Meilleure maîtrise des risques
 - Les utilisateurs acceptent progressivement le produit
 - Meilleure réactivité
- Inconvénients :
 - Le nombre d'incrémentations doit être assez important
 - Marche si le projet est géré en interne (client=fournisseur)



XP : Extreme Programming

C'est quoi ?

- Naissance : année 2000
- Cycle itératif
- Une méthode appelé « agile »
- Mise en avant du rôle du développeur
- Réduction de la documentation
- Renforcement du lien client/intégrateur

Les valeurs

- La communication :
 - Client intégré dans le projet
 - Groupe soudé
- Feedback (commentaire)
 - Travail en binôme
 - Retour du client
- La simplicité :
 - Supprimer ce qui est inutile
- Le courage
 - Jeter le code mauvais
 - Le client doit donner des priorités

Le fonctionnement

- Pratiques de programmation :
 - Une conception simple, Refactoring, Tests unitaires, Tests fonctionnels
- Les acteurs :
 - Coach, manager, tracker, développeur, testeur, client
- Règles :
 - programmation en binôme, responsabilité collective du code, intégration continue

Les avantages

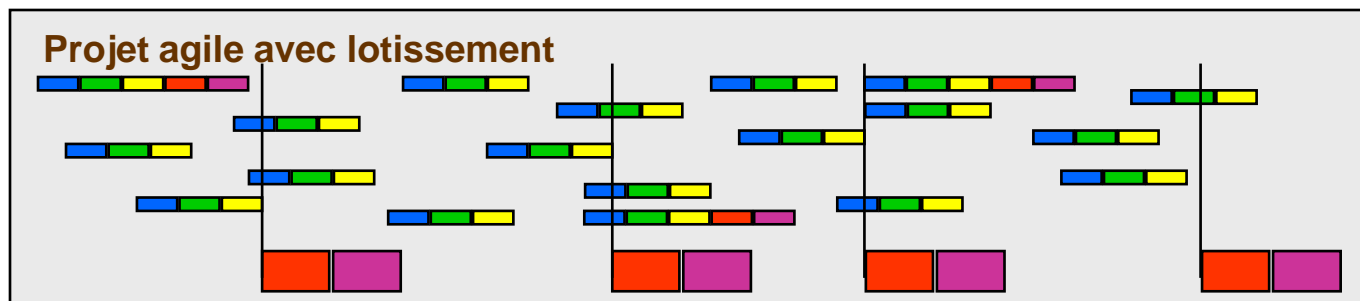
- Qualité
 - maintenabilité,
 - Efficacité
 - Adaptabilité
- Transparence
 - client intégré
- Respect des objectifs
- Ambiance de travail

Les inconvénients

- Langages : adaptés pour des langages objets
- Projet de petits envergures
- Equipe réduite, et devant être très compétente et soudée
- Travail en binôme
- Clients omniprésents

A base d'XP ... exemple Celerial

- La notion de besoin remplacé par la notion de service
- Découpage en Lots et en îlots
- Reprise des valeurs et de la méthodologie XP pour chacun de ces lots.
- Découpage du projet en îlots indépendants





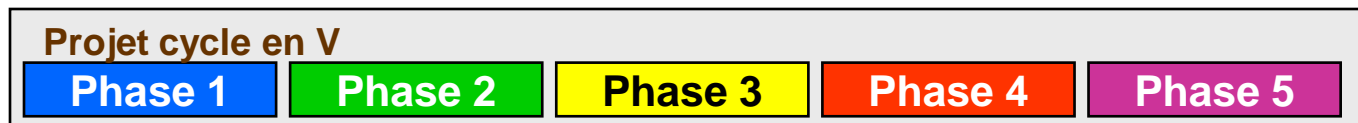
Que choisir ?

Questions

- Sommes nous d'accord sur le besoin ?
- Quel est le degré de modification ?
- Quel est le degré d'implication du client ?
- Quel est le degré de maîtrise du client ?
- Quel est le délai ? Le budget ? La qualité ?

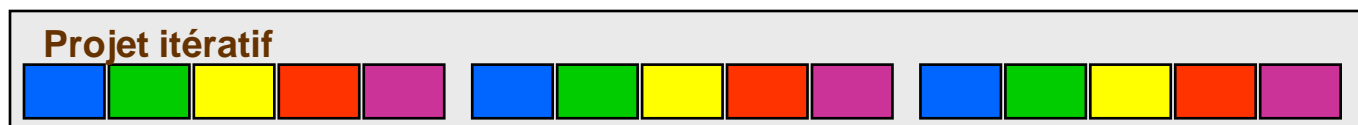
Cycles séquentiels

- Le projet est important, peu d'implication du client, vieilles technologies maîtrisées :
 - Le cycle en V est très approprié
 - Si la technologie n'est pas totalement maîtrisée on peut rajouter une phase de prototypage maquettage
 - Si les tests sont réduits → simple cascade



Cycles orientés itérations

- Besoins pouvant évoluer dans le temps, technologie objet utilisée, client très impliqué
 - Mode spirale afin de garantir une intégration performante des besoins
 - Equipe expérimentée et limitée à une dizaine de personnes → XP





Les étapes parallèles



Le transverse

- Gestion des environnements
 - Données de tests
 - Administration des serveurs
 - Mise en place du réseau informatique
 - Installation/Masterisation du parc informatique
- Le contrôle qualité
- capacity planning
 - Suivi du ou du futur système développé ou en cours de développement

La formation

- Formation des futurs utilisateurs de l'application
 - Mise en place d'un environnement de formation
 - Adaptation du projet par rapport aux premières remontés
- Formation des futurs administrateurs de l'application
 - Prise de connaissance de l'application
 - Formation aux nouveaux outils pouvant être mis en place
 - ...