

**Qu'est-ce qu'un projet ?**

# Sommaire

- Définition
- Typologie
- Les acteurs et les rôles
- Le déroulement

# Sommaire

- **Définition**
- Typologie
- Les acteurs et les rôles
- Le déroulement

# Projet : définition, exemple

- Un projet représente une intention, un plan à réaliser
- Exemples :
  - Projet immobilier ;
  - Projet de voyage ;
  - **Projet informatique.**

# Un projet : ça consiste en quoi ?

- De la réalisation bien sûr !  
Du « code » en informatique !
- Mais aussi :
  - De la préparation :
    - Elaboration
    - Faisabilité
    - Confrontation avec des tiers
    - Conception
  - Du suivi :
    - Délais
    - Budget
  - Un bilan !

# Un projet s'inscrit dans le temps

- Date de début et date de fin (prévue / effective)
- Jalons pour suivre l'avancement
- Même après la livraison, le projet continue :
  - Suivi des utilisateurs
  - Exploitation
  - Maintenance, corrections
  - Evolutions
  - Nouvelles versions
  - Réutilisation pour d'autres projets

Finalement dans un projet ...

**70 % des efforts sont  
hors réalisation !**

# Pourquoi faire un projet

- Pour avoir un résultat
- Balance Efforts / Résultats
- Suivi et rupture

Le projet est réussi si le résultat est positif

# Financement et contrat

- Un projet doit être financé, il est lié à un budget
- Le budget fait partie d'un engagement contractuel, à valeur juridique
- Réalisation d'un plan budgétaire avec évaluation des écarts et du pourcentage de risque : connaître ses marges de manoeuvre
- La maîtrise du budget est fondamentale :
  - Définition et suivi des lignes budgétaires
  - Négociations

# Sommaire

- Définition
- **Typologie**
- Les acteurs et les rôles
- Le déroulement

# Les projets informatiques

- Dans un contexte de recherche, les projets de logiciels informatiques diffèrent notamment selon :
  - leur organisation et leur finalité
  - les personnes impliquées
  - l'architecture du logiciel mis en oeuvre

# Organisation et finalité

- Facteurs de différenciation :
  - Taille et durée
  - Equipes : internes ou externes
  - Statut du projet
  - Finalité :
    - Valider un travail de recherche
    - Proposer une application finalisée

# Cas types

- Chercheur/Ingénieur qui a l'intention de tout faire lui-même
- Chercheur/Ingénieur qui travaille avec son informaticien alloué (permanent, affecté une partie de son temps sur le projet)
- Chercheur/Ingénieur qui fait appel à une SSII ou a recours à un poste informatique temporaire

# Architecture du logiciel

- Application mono-utilisateur :  
Chaque utilisateur :
  - installe le logiciel sur sa machine
  - dispose de ses propres données
  - gère ses changements de version
- Application multi-utilisateur :
  - une seule base de données centralisée
  - accès sur un réseau local ou sur Internet
  - nécessite un hébergeur, un exploitant

# Application mono-utilisateur

- Avantages :
  - Fonctionnement en mode déconnecté et itinérant
  - Pas de moyens à allouer à la sauvegarde et à l'exploitation
- Inconvénients :
  - Hétérogénéité des plate-formes techniques, d'où tests et support importants
  - Perte possible d'informations pour les utilisateurs (défauts de sauvegardes)
  - Mises à jour compliquées
  - Perte de la maîtrise du logiciel : propriété, diffusion

# Application multi-utilisateurs

- Avantages :
  - Hébergements et maintenances centralisées
  - Moins d'adhérences avec le contexte de l'utilisateur (surtout avec les architectures web)
  - Maîtrise totale du logiciel et des utilisateurs
- Inconvénients
  - Nécessite une connexion
  - Ergonomie moindre
  - Moins d'intégration avec les logiciels de l'utilisateur (suite bureautique par exemple)
  - Temps de réponse plus longs

# $\beta$ : Le Projet BETHA

- Outil de simulation des rendements et des marges de cultures de blé
- $\beta$  est représentatif d'un cas de projet informatique pouvant être mis en oeuvre à l'INRA pour valider et mettre en application des modèles de recherche agronomique
- $\beta$  a déjà servi de support pour un projet d'ingénieur à AgroParisTech
- Cet exemple sert de « fil rouge » tout au long de la formation : symbole  $\beta$  et cadre rouge

# $\beta$ : Caractéristiques

- Organisation, acteurs et finalité :
  - Projet chapeauté par un chercheur
  - Spécifications et prototype réalisés par un groupe d'étudiants
  - Tester et valider un modèle agronomique
  - Diffusion envisageable à terme
- Architecture :
  - Web dans le cas du prototype
  - Classique au final

# Sommaire

- Définition
- Typologie
- **Les acteurs et les rôles**
- Le déroulement

# Les principaux acteurs

- Le client ou maître d'ouvrage : MOA
- Le maître d'oeuvre : MOE
- L'intégrateur
- L'hébergeur
- L'exploitant

# Le client ou maître d'ouvrage

- C'est le commanditaire du projet
- Celui qui paye
- Celui qui exprime le besoin
- Celui qui fournit les éléments, les « inputs »
- Celui qui arbitre
- Signe le contrat

La maîtrise d'ouvrage n'est pas un informaticien !

# Le maître d'oeuvre

- Conçoit le projet avec le client
- Négocie le budget et les avenants
- Dirige la réalisation
- Coordonne, pilote
- Rend des comptes régulièrement au client
- Dans les petits projets, joue le rôle d'intégrateur également

# L'intégrateur

- Réalise la partie :
  - programmation
  - assemblage de composants
  - paramétrage
- Est suivi par la MOE

# L'hébergeur

- Fournit les infrastructures techniques nécessaires au fonctionnement du logiciel :
  - Serveurs
  - Espace disque
  - Réseau
- Rôle prépondérant dans le cas des applications de type web, c'est-à-dire installées sur un serveur et pas sur les postes des utilisateurs
- Peut être interne ou externe
- Doit anticiper la montée en charge de l'utilisation du logiciel

# Hébergeur : incidence

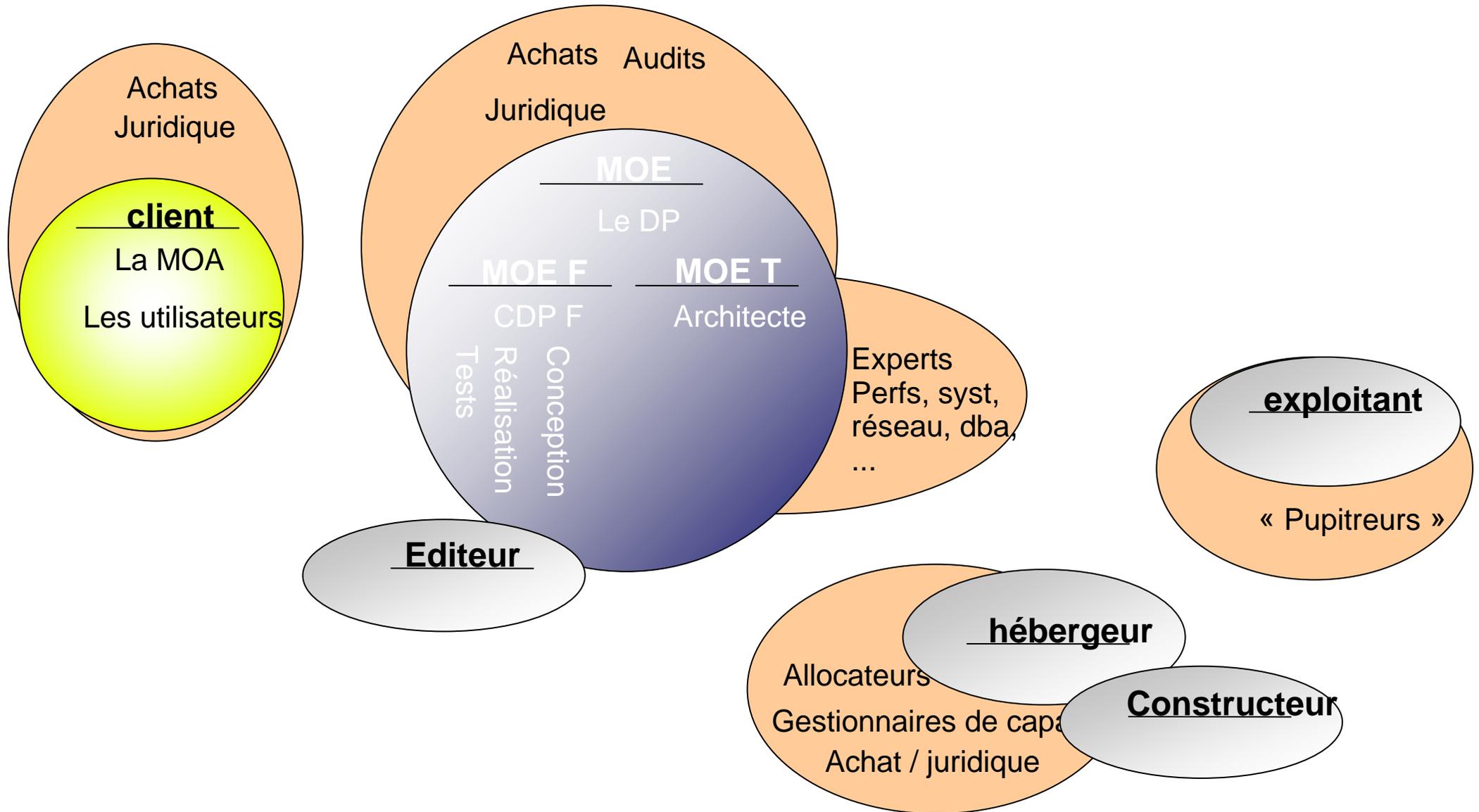
- Rôle prépondérant dans le cas des logiciels multi-utilisateurs
- L'hébergeur doit être consulté préalablement à tout développement car il définit des normes :
  - Système d'exploitation
  - Composants logiciels
  - Bases de données
  - etc.

# L'exploitant

- Assure le suivi du fonctionnement de l'application
- Assure le contrat de service (QOS)
- Gère les incidents
- Gère les sauvegardes et restaurations
- Garantit la sécurité
- Rôle souvent oublié mais primordial

L'exploitant assure la vie du projet et la continuité de service après sa mise en production !

# Positionnement des acteurs



# Responsabilité des acteurs

Il est important de connaître  
le rôle et les obligations de  
chacun pour que le projet  
se déroule bien !

# $\beta$ : Les acteurs

- MOA : Chercheurs INRA
- MOE : étudiants de la spé MISI
- Hébergeur/Exploitant : service informatique de l'INRA

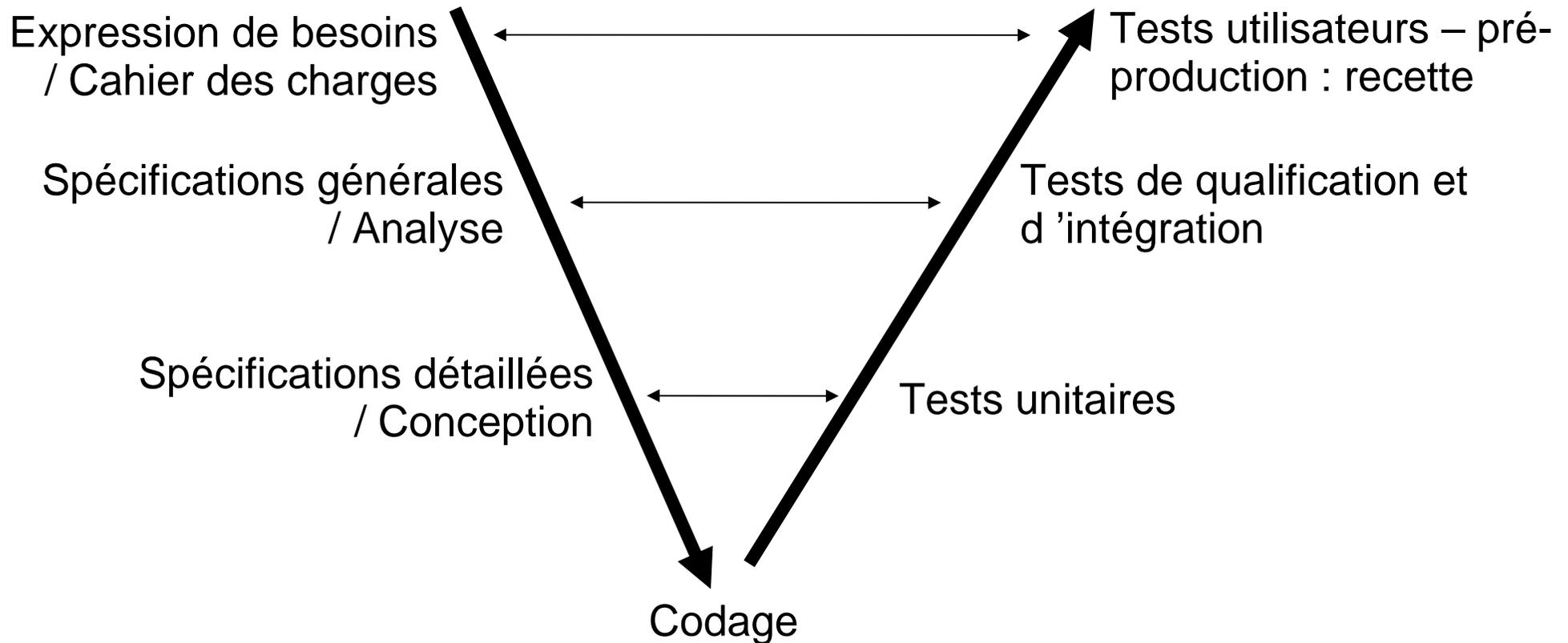
# Sommaire

- Définition
- Les acteurs et les rôles
- Typologie
- **Le déroulement**

# Découpage d'un projet

- Un projet est découpé en plusieurs phases
- Le découpage est essentiel pour :
  - Planifier
  - Assurer le suivi
- Le projet peut être réalisé en un ou plusieurs cycles de ces différentes phases
- Il est important de revalider toutes les phases en cas de modification, afin d'avoir une documentation à jour par rapport au code

# Le cycle en V



# Le cycle en V : décryptage

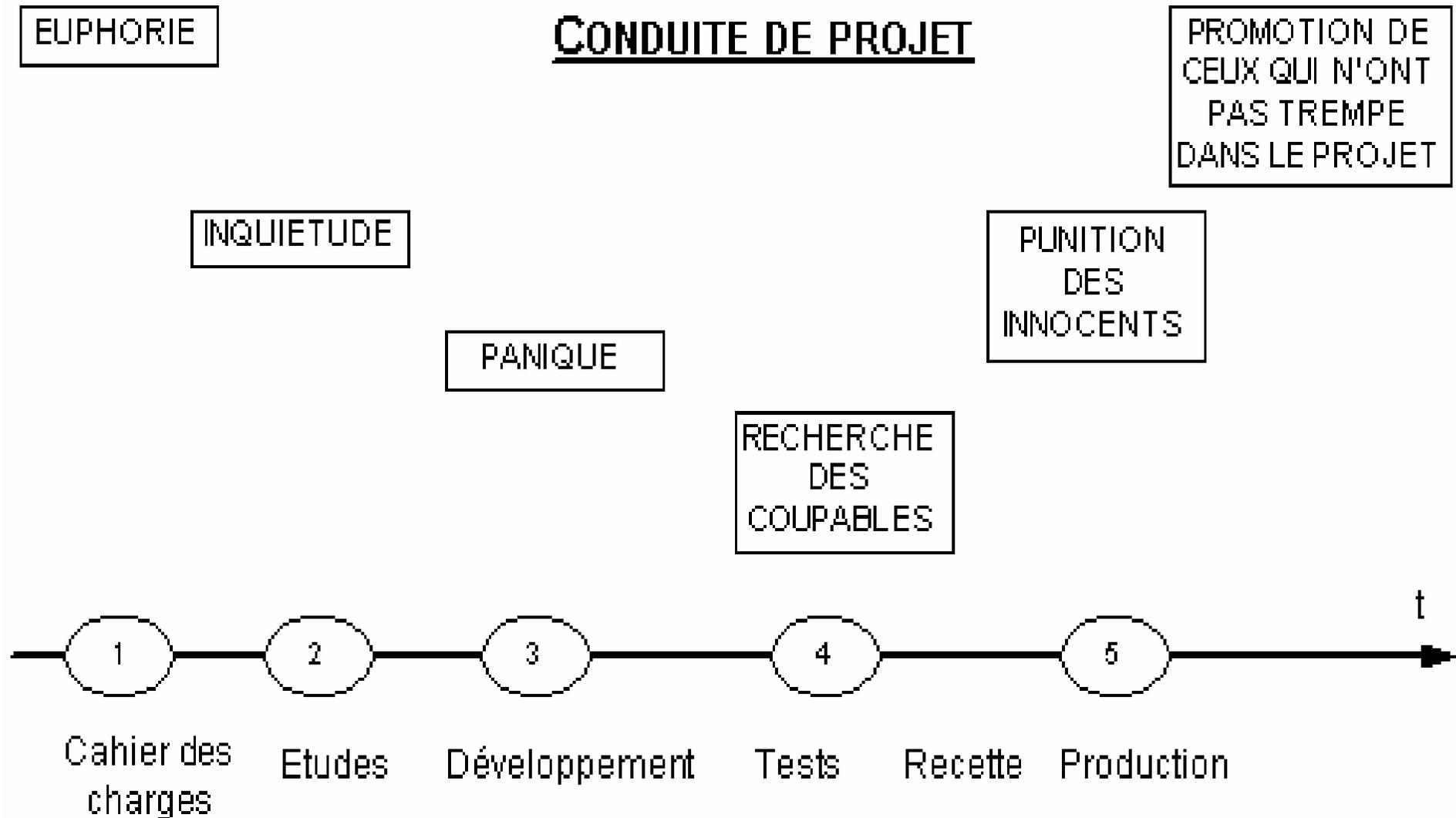
- De haut en bas :
  - du besoin utilisateur vers le code
  - du fonctionnel vers le technique
- Partage horizontal :
  - à gauche : avant le code
  - à droite : après le code
- Correspondance horizontale entre les phases :  
par exemple les tests unitaires permettent de valider les spécifications détaillées
- Les « phases de droite » sont préparées à partir des « phases de gauche » avant la livraison du code

# Cycle en V : rôle des acteurs

- Les documents sont produits par la MOE
- Mais le rôle de la MOA est essentiel :
  - Fournit les informations : interviews
  - Exige les bons livrables
  - Exige un planning des livrables
  - Valide les livrables produits à chaque phase  
==> Permet de passer à la phase suivante

La MOA doit donc connaître le contenu des livrables produits à chaque phase !

# Parodie



# Implication des acteurs

- Mis à part la phase de codage, chaque phase implique les différents acteurs
- Chaque phase donne lieu à la rédaction et à la validation de documents
- Itération du cycle car :
  - division du projet en lots distincts ;
  - évolution de la demande au cours du projet ;
  - complexification du projet.
- On gère une traçabilité sur les différentes phases

# L'approche agile

- Apparue en réaction à certains échecs de projets notamment explicables par :
  - La lourdeur des procédures
  - L'effet « cascade » : chaque retard dans une phase se répercute dans les autres
  - Le déphasage entre les spécifications d'origine et le code
- Permise par les évolutions technologiques :
  - Intégration et livraison continues
  - Connexion permanente avec le client

# Responsabilité du client

A tout moment, le client doit  
« maîtriser le timing »,  
savoir à quelle phase on  
en est, si on est en retard  
ou pas !