Gestion de Projet Informatique

Partie 1 : Introduction

Cycles de vie du projet : classiques et agiles

CV - Entretien

Licence d'Informatique 3e Année
Tianxiao Liu
CY Cergy Paris Université

Sommaire

- Objectif du cours
- Modalité de contrôle de connaissances (MCC)
- Les différents rôles
- Les étapes d'un projet informatique
- Objectif d’un cycle de vie de gestion de projet
- Les modèles séquentiels: Cascade, V
- Les termes et notions agiles
- Les modèles agiles : ASD, FDD, Crystal
- CV : éléments clés
- Entretien : l’essentiel

Objectif du cours

- Un ***pont*** vers le monde professionnel
- Simulation d'une petite expérience en entreprise
- **Des activités pédagogiques collectives**
- Méthodologies pour gérer un projet informatique
- Clé de réussite : **esprit d'équipe + jeu de rôle**

Modalité de contrôle de connaissances

- **Note CC : 50%**
  - 5 séances de TD
    - Productions liées aux activités pédagogiques
  - Documents écrits
  - Présentations orales
  - La plupart de tâches sont à finir en séance (idéal)
- **Note d'examen : 50%**
  - Documents autorisés (sauf appareils électroniques et livres)
  - **Mise en situation**

Les rôles à jouer

- Chef de projet (CHEF)
- Maître d'Œuvre (MOE)
- Maître d'ouvrage (MOA)
- Responsable de qualité (QA)
- Responsable de validation / recette (RVR)

Chef de projet (CHEF)

- **Profil / compétences**
  - Etre ouvert et social, expressions orales
  - Vision globale, capacité de modélisation
  - Etre sensible à la gestion du temps
  - Une **large** connaissance sur les techniques (programmation, intégration, etc.)

- **Responsabilités**
  - Orientation de l'évolution du produit, initialisation de conception
  - **Gestion de l'équipe, organisation du travail pour chaque séance → communications internes**
  - En cas de problèmes, prendre la décision (après avoir discuté avec ses collègues)

Maître d'Œuvre (MOE)

- **Profil / compétences**
  - Etre très bon en programmation, connaissances techniques approfondies
  - S'adapter facilement aux nouvelles technologies
  - Gestion globale du développement
  - Etre capable de simplifier les problèmes complexes
  - Convention de codage et esprit de programmation orientée objet

- **Responsabilités**
  - Architecture et conception du logiciel, organisation du code source et de la structure du projet
  - Aider aux autres membres de l'équipe à configurer leur environnement de travail, pour assurer le bon déroulement du développement
  - Evaluation des compétences/efficacités de développement des membres de l'équipe → décisions

Maître d'Ouvrage (MOA)

- **Profil / compétences**
  - Etre très bon en expression orale et écrite
  - Etre ouvert aux idées
  - Etre bon en synthèse/résumé des points clés
  - Savoir trouver l'équilibre entre la satisfaction du client et l'avancement du projet dans les délais

- **Responsabilités**
  - Assurer la communication entre le client et l'équipe
  - Formuler les besoins
  - Prise en compte des feedbacks
  - Aide à la planification du projet avec CHEF

Responsable d'assurance qualité (QA)

- **Profil / compétences**
  - Etre très rigoureux sur les tests effectués
  - Etre très bon en outils de développement et de tests
  - Etre bon en communication interne

- **Responsabilités**
  - Assurer la qualité du produit fabriqué (du point de vue des utilisateurs et des développeurs)
  - Couverture des fonctionnalités par rapport au cahier des charges
  - Assurer les tests d'intégration, tests de performance ainsi que les tests fonctionnels
  - Construction du plan de tests
  - Mise à jour des tests avec l'avancement des tâches

Responsable de validation / recette (RVR)

- **Profil / compétences**
  - Etre à la fois rigoureux, créatif et flexible
  - Etre très bon en rédaction
  - Etre capable de bien structurer les documents
  - Etre pragmatique et efficace

- **Responsabilités**
  - **Concevoir et valider des rendus (document, programme, etc.) pour chaque séance**
  - Rassurer que l'équipe soit sur le bon chemin pour les rendus définis

Définitions des termes de base

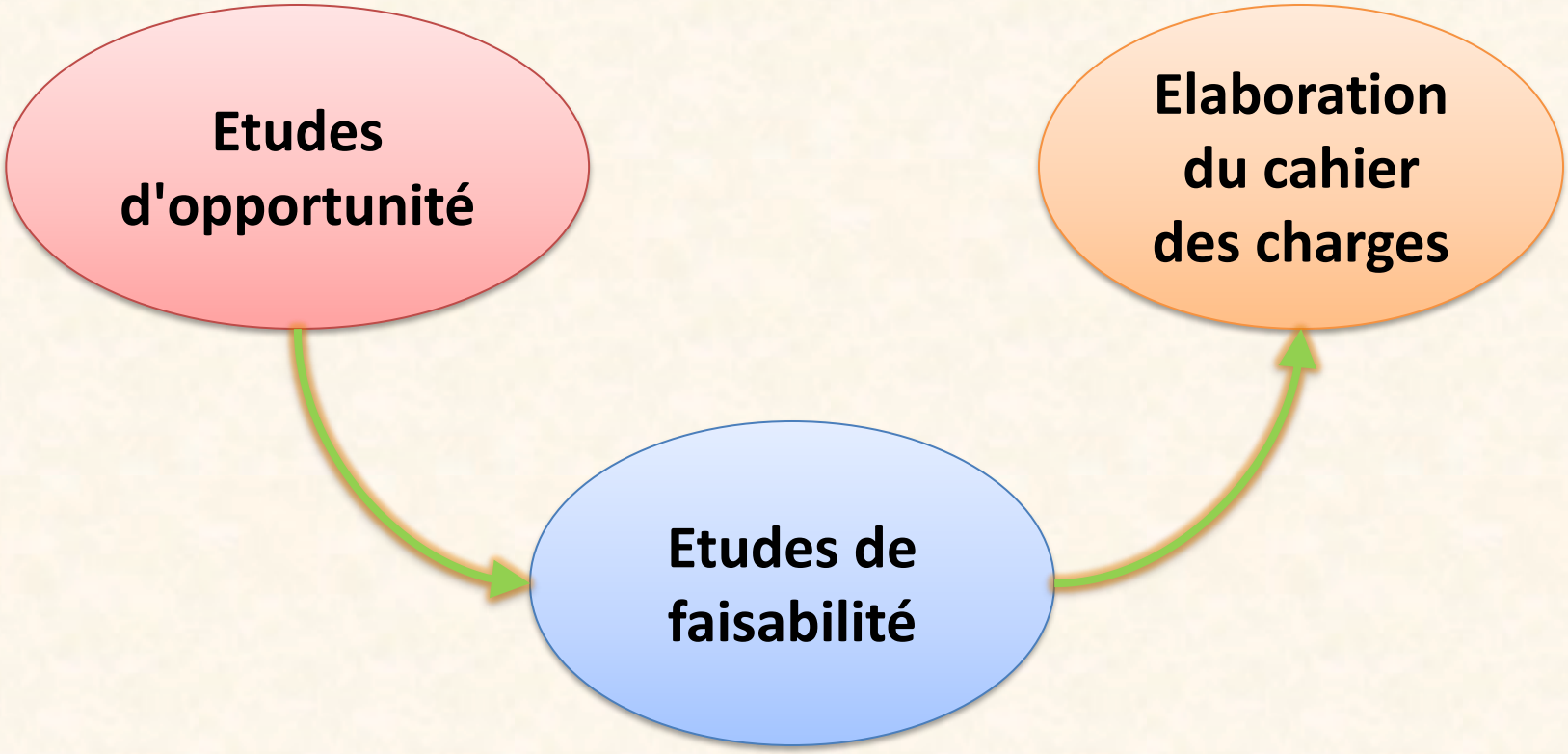
- **Un projet informatique**
  - Un ensemble d'**activités** prises en charges
  - Dans un **délai** donné
  - Dans les limites de **ressources** imparties
  - Par des **personnes** y sont affectées
  - Dans le but d'atteindre des **objectifs** définis

- **Une étape de projet**
  - Une thématique regroupant un ensemble d'**actions** à entreprendre
  - Répondre à un besoin défini dans les délais fixés

Les étapes générales d'un projet

- L'avant-projet
- Conception et réalisation
- Validation
- Recette et réception
- Livraison
- Bilan de fin de projet

L’avant-projet



L'avant-projet

- **Etude d'opportunité**
  - Etudier la demande de projet: concept → viable?
  - Valider la demande des utilisateurs
    - Périmètre du projet: contexte
    - Utilisateurs finaux: qui utiliseront le produit ?
  - **Conseils**
    - **En raison d'efficacité, associez les utilisateurs à la réflexion globale → Réunion**
    - **N'hésitez pas trop sur les points faibles!**

L'avant-projet

- **Etude de faisabilité**
  - 3 dimensions: économique, organisationnelle et technique
  - Présentation des solutions
    - Aspect fonctionnel, technique et impact
    - Avantages et inconvénients de la solution
    - Conditions de mise en œuvre: moyens humains et matériels, documentations, etc.
  - **Conseil**
    - **Restez clair et synthétique, sans trop détailler !**

L'avant-projet

- **Cahier des charges** : objectifs
  - Garantir que les livrables seront conformes à ce qui est écrit
  - Tout doit être défini dès le début? → cas idéal mais pas réaliste ! → **projet agile**
  - Un document de référence: un outil de dialogue modifiable en cours de projet

L'avant-projet

- **Cahier des charges** : contenu général (rappel GLP)
  - Présentation du projet
    - Contexte du projet
    - Objet : Brève définition
    - Organisation : équipe de projet → rôle de chacun
    - Environnement : existants fonctionnels et techniques
  - **Objectif: description de la future solution**
    - Points clés sur les besoins fonctionnels
    - Limites du projet: licences, budget, etc.
    - **Description détaillée de l'ensemble des fonctionnalités attendues : scénarios d'utilisation**

L'avant-projet

- **Cahier des charges** : contenu (suite)
  - Livraison attendue (liste non exhaustive)
    - Date à laquelle le projet devra impérativement être terminé
    - Programmes → code source
    - Documents à remettre
    - Base de données
  - **Conseil : Attention au vocabulaire !!**
    - **Projets échoués → manque de communication, de culture et de vocabulaire commun**

Conception et réalisation

- **Conception**
  - Période primordiale → approchée avec attention!
  - **Conseil: Un document central aide !**
  - Analyse conceptuelle mais aussi fonctionnelle
    - Niveau conceptuel: MCD (Modèle Conceptuel de Données) et MCT (Modèle Conceptuel de Traitements)
    - Niveau organisationnel: MLD (Modèle Logique de Données) et MLT (Modèle Logique de Traitements)
    - Niveau physique: MPD (Modèle Physique de Données) et MOT (Modèle Opérationnel des Traitements)

Conception et réalisation

- **Conception (points clés)**
  - Concevoir du plus général au plus précis
    - Pièce par pièce mais aussi ensemble!
  - Ne pas hésiter à recommencer
    - On peut très bien se tromper !
    - Concevoir avec rigueur → Eviter de grands risques en réalisation
  - Savoir mettre fin à la conception
    - Trouver le juste équilibre entre conception et temps de développement

Conception et réalisation

- **Réalisation**
  - Phase de développement (codage)
  - Nécessité d'une très bonne planification
  - Utilisation du **prototype**

- Comment découper le projet?
  - Ce qui doit être fait (tâches)
  - Par qui (ressources)
  - Comment les résultats (livrables) doivent être présentés
  - Comment les valider et à quelle étape → cycle de vie et intégration continue (à traiter plus tard en cours)

Validation

- S’assurer que le produit réalisé répond bien au cahier des charges

- Tests unitaires (automatisé)
  - Fonctionnement correct d’une partie ou d’un module
  - Réalisés majoritairement par les développeurs

- Tests d’intégration
  - Bon fonctionnement de l’ensemble du projet
  - Réalisés et rassurés par QA
  - Attention: **au fur et à mesure de l'avancement du projet**

Validation

- Tests fonctionnels
  - Fonctionnalités d'une application → sans se soucier des détails d'implémentation
  - Attention: Ce ne sont pas des tests d'intégration !
  - Assurés par QA dès le début : intégration continue
- Test de performance
  - Problèmes de performance ?
- Plan de test
  - Périmètre dans lequel s'effectueront les tests
  - Différents scénarios et jeux d'essais nécessaires
  - Bilan des tests, observation, etc.

Recette et réception

- **Les choses à faire pour la recette**
  - Prévoir l'installation du produit
  - Explication de l'IHM
  - Prévoir les interfaces avec d'autres applications / systèmes
  - Présentation de la performance du produit
  - Mesures de sécurité
  - Robustesse (support mémoire, erreurs internes, changement de mode)
  - Aspect réseaux
  - Exploitation (sauvegarde, restauration…)

Recette et réception

- **Document de réception**
  - Objectifs de la réception : conformité fonctionnelle du produit réalisé
  - Présentation des livrables à réceptionner
  - Limites connues
  - Vocabulaire ou abréviations
  - Responsabilité → **droits informatiques**
  - Procédure de réception
  - Ajournement de la réception (en cas de retard)

Livraison

- Mise à disposition
  - Sites pilotes
    - Présentation du produit, forum pour identifier les bugs
    - Mises à jour du produit achevées, en cours ou dans le futur
    - Formations possibles pour mieux se servir du produit
    - Une entraide
    - Une assistance (téléphonique, en ligne, etc.)
    - Une maintenance
  - Suivi de la livraison
    - Feedbacks des clients
    - Mises à jour correctives ou évolutives

Bilan de fin de projet

- Constituer une base de connaissances, d'expériences et de pratiques
  - **Un enrichissement pour les futurs projets !**

- Les rubriques d'un bilan de projet
  - Bilan de la charge et du coût des achats
  - Bilan de l'organisation
  - Bilan technique
  - Anomalies et évolutions
  - Bilan qualité et méthodes
  - Estimation des coûts d'obtention de la qualité
  - Appréciation globale → Enquête de satisfaction

Objectif d’un cycle de vie 1/2

- Définir **la démarche** à appliquer à un projet

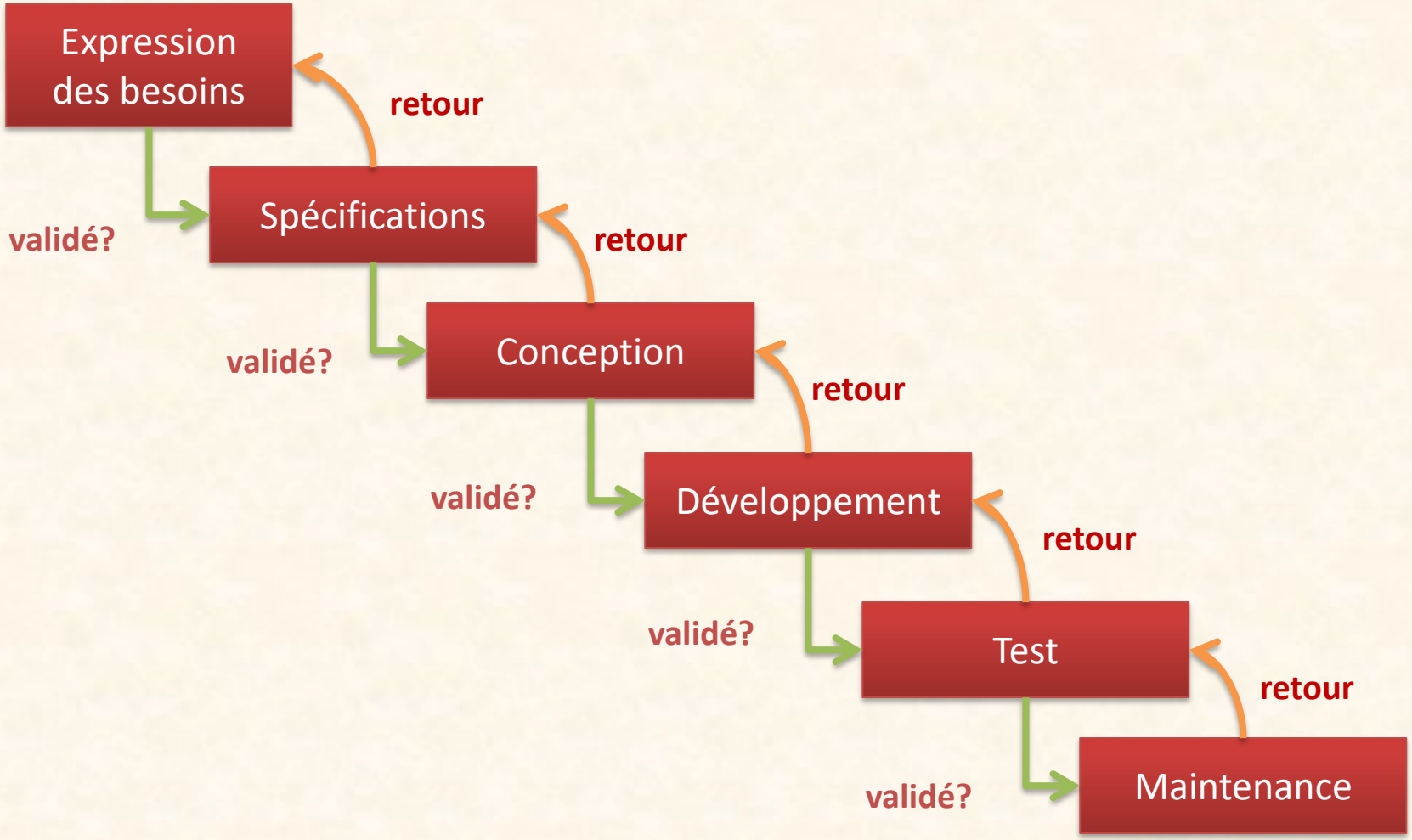
- **Attention :** on parle ici des cycles de **développement** de projet, pas de notions marketing

- **Choix : 4 facteurs majeurs à prendre en compte**:
  - Minimiser les évolutions fonctionnelles en cours de projet
  - Soutenir la démarche d’assurance qualité (QA)
  - Maîtriser les coûts et les risques
  - Garantir sa conformité aux règles contractuelles et juridiques

Objectif d’un cycle de vie 2/2

- D’autres éléments pouvant avoir une influence sur le choix de cycle de vie
  - La culture de l’entreprise
  - Les équipes de projet
  - Le niveau de connaissance des utilisateurs
  - Le contexte du projet
  - Les acteurs intervenant
  - Le type du projet

Modèle en cascade

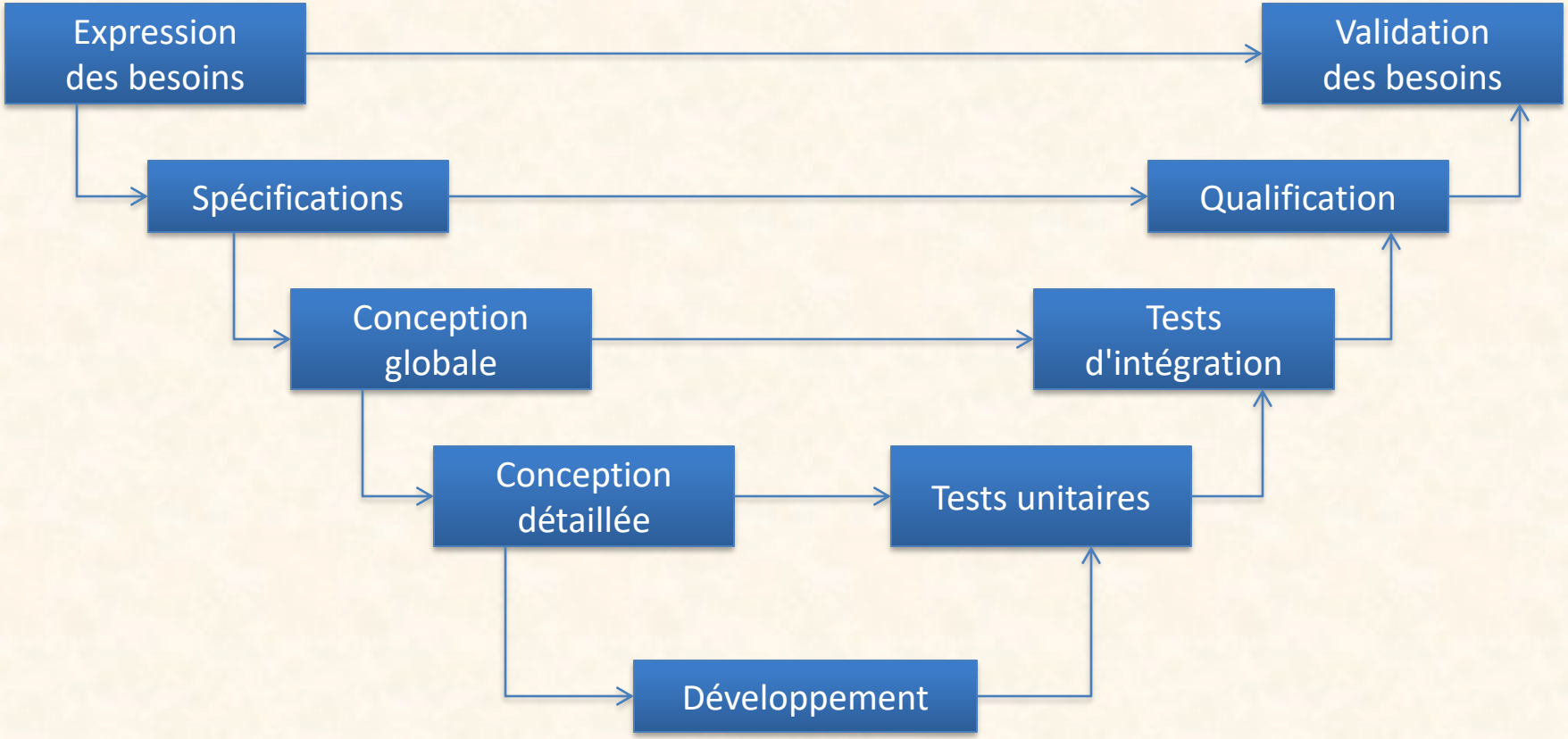


Modèle en cascade

- **Principe**
  - Une suite de phases dans un déroulement **linéaire**
  - Validé → avancer, non validé→ retourner

- **Avantages et inconvénients**
  - Bonne réduction de risques en minimisant l'impact des incertitudes
  - Bonne solution pour les projets peu complexes
  - Problèmes non découverts avant les tests
  - Pas de prise en compte des évolutions
  - Difficulté d'amélioration des performances
  - Durée du projet inférieure à **un an**

Modèle en V


Expression des besoins
Validation des besoins
Spécifications
Qualification
Conception globale
Tests d'intégration
Conception détaillée
Tests unitaires
Développement

Modèle en V

- **Principe**
  - Modèle orienté tests
  - Décomposition et recomposition

- **Avantages et inconvénients**
  - Proposer au fur et à mesure une démarche de réduction des risques, en minimisant progressivement l'impact des incertitudes
  - Exclusion de l'utilisateur dès la phase de conception car trop technique
  - Contrôle qualité significatif seulement en fin de projet
  - Dans un contexte compétitif, risque éventuel de ne pas respecter les délais

Définitions des termes agiles

- **Prototype**
  - L'embryon du produit initial : **"Je saurai ce que je veux quand je le verrai"**
  - Viser à livrer rapidement une maquette de la solution à développer avec un minimum de fonctions **viables**
  - Clarifier les besoins afin d'y arriver à une meilleure définition des spécifications fonctionnelles et techniques
  - Eviter l'écart entre les besoins réels, ceux exprimés et ceux interprétés

Définitions des termes agiles

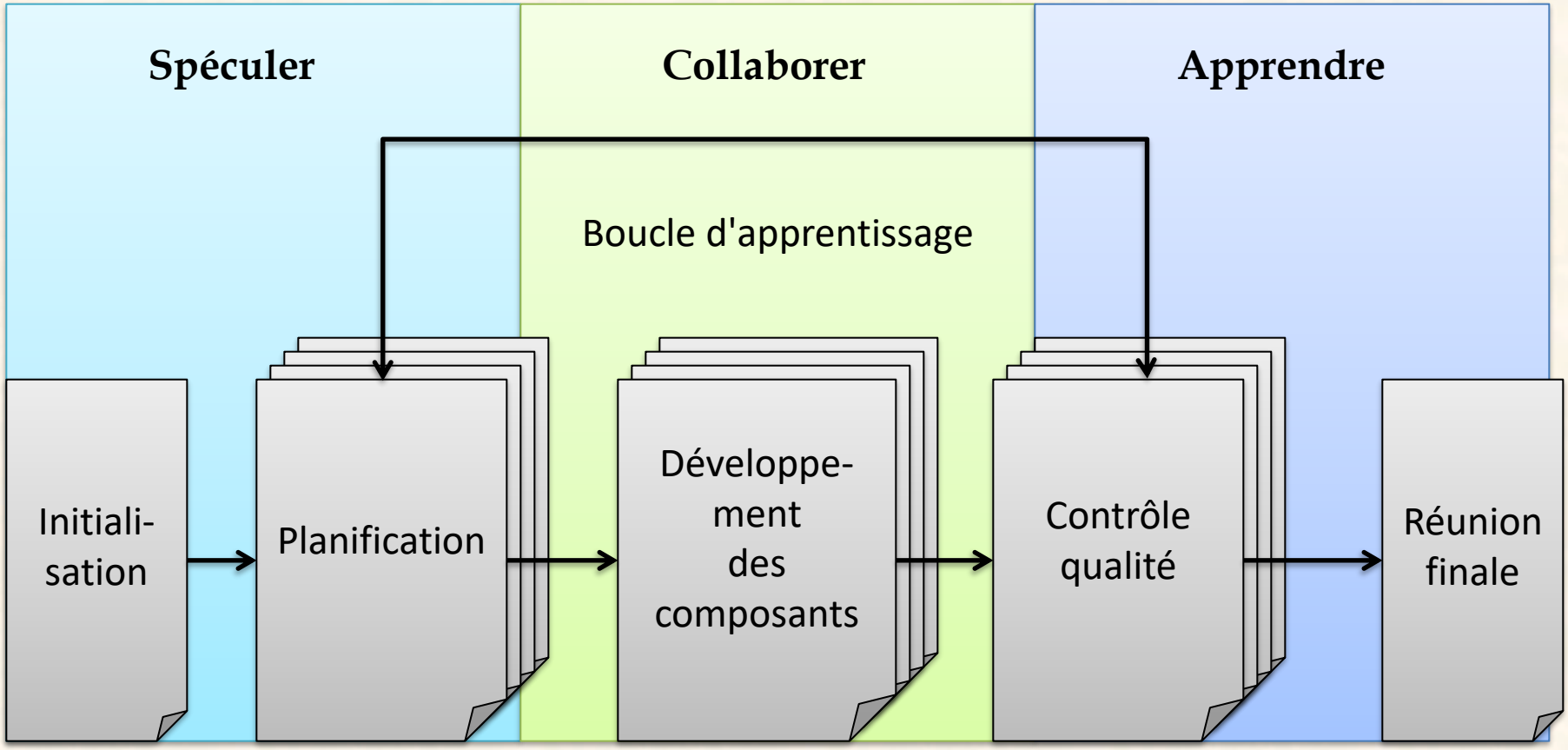
- **Effet tunnel**
  - Point de départ : connu
  - Point d'arrivée : **inconnu**

Rester dans le tunnel noir

- Pour les clients
  - Pendant très longtemps, pas de communication avec les membres de l'équipe de projet
  - Grand risque d'avoir un résultat non satisfaisant !

- Pour les développeurs
  - De grandes phases de refactoring du code

Modèle agile : ASD
Adaptive Software Development



Modèle agile : ASD

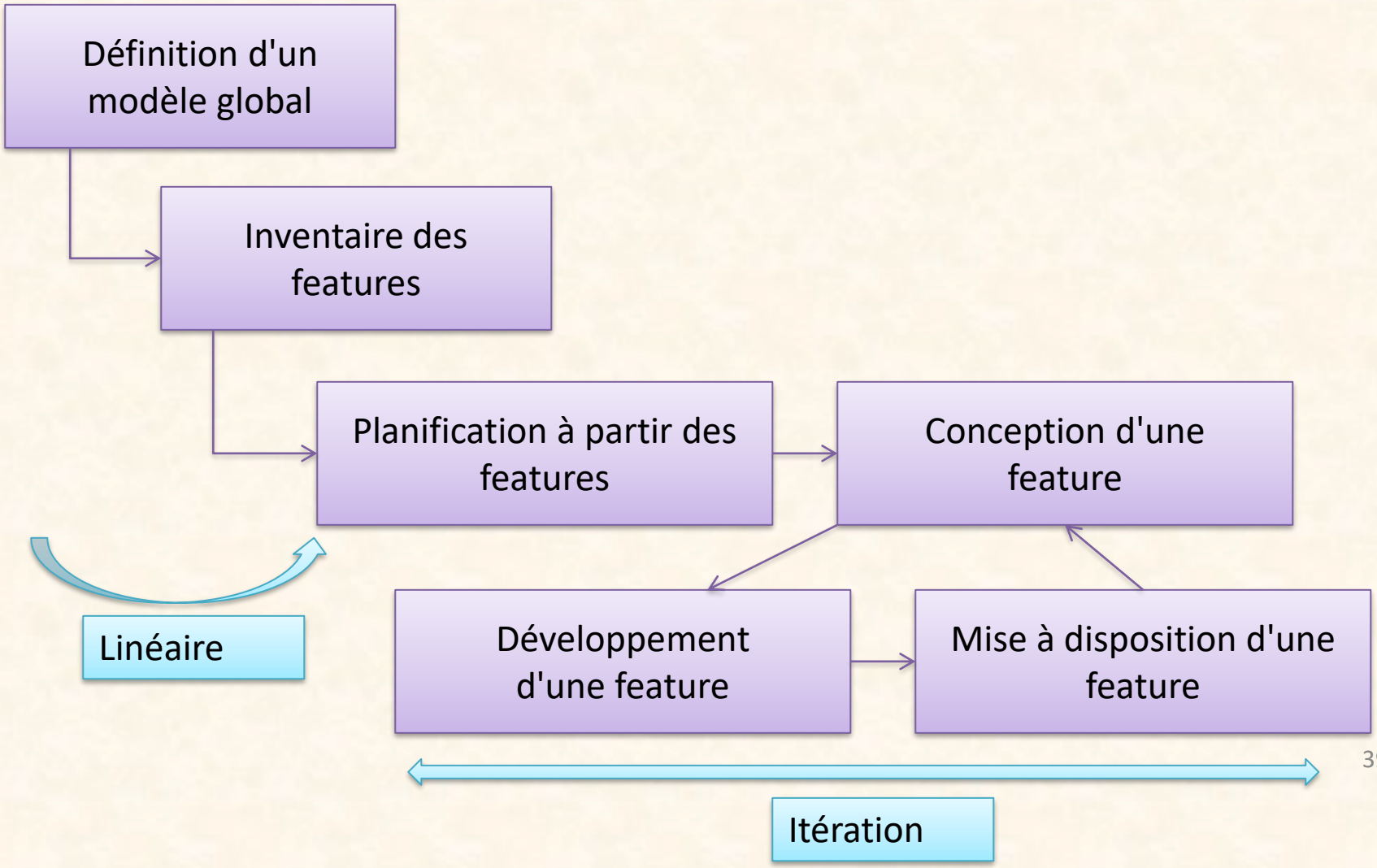
- **Spéculation**
  - Initialisation (cahier des charges) : Objectifs, contraintes, intervenants, dates, estimation des charges, risques critiques
  - Détermination du nombre optimal d'itérations et de leurs contenus (composants) → Un **tableau de bord**
- **Collaboration**
  - Collaboration optimale entre les membres de l'équipe
  - Assurer **l'efficacité** de la production
- **Apprentissage**
  - Vue de la maîtrise d'ouvrage (clients)
  - Vue de l'équipe : points positifs et négatifs
  - Vue de progression du projet
    - **Où en est-il par rapport à ce qui était prévu?**

Modèle agile : ASD

- **Caractéristiques**
  - S'adapte particulièrement aux projets **e-business**
    - Réalisation en des temps très courts
    - Support de nombreux changements et incertitudes
  - Focalisation : viser les résultats plutôt que les tâches
  - Itération : évolution en fonction des retours d'utilisateurs
  - Changement : capacité à supporter un changement fonctionnel ou technique en cours de développement
- **Avantages / Inconvénients**
  - Grande souplesse dans le changement
  - Rapidité, respect des délais
  - Implication du client
  - **Gestion des risques : absence totale de stabilité !**

Modèle agile : FDD

Feature Driven Development



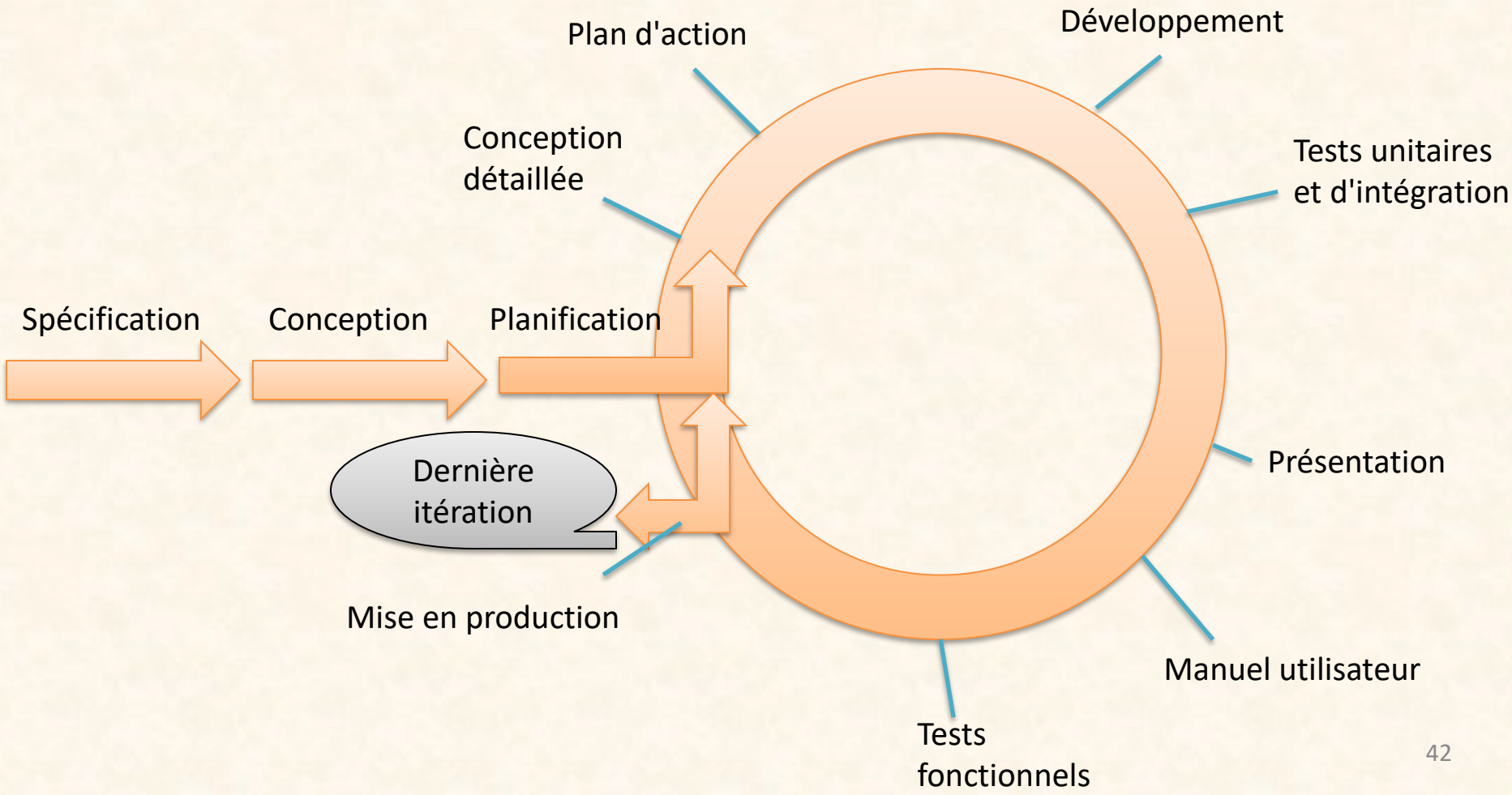
Modèle agile : FDD

- **Feature**
  - ***Client-valued fonctionality***
- **Description des phases**
  - Développer un modèle global
  - Etablir une liste détaillée de *features* classées par priorité (inventaires de *feautures*)
  - Planifier à partir des *features*
  - Concevoir à partir des *feature*
  - Construire (développement) à partir des *features*
  - Mise à disposition des *features*

Modèle agile : FDD

- **Principe**
  - Mise en places des itérations très courtes
  - Chaque itération produit un livrable fonctionnel
  - Bases : caractéristiques de l'application (*features*)
- **Avantages / Inconvénients**
  - Motiver les développeurs : livrable utilisable
  - Sécuriser le chef de projet : état de l'avancement visible au gré des itérations
  - Satisfaire les clients : livrables concrets, planning clair
  - Maintenance de l'ensemble des composants peut parfois se muer en challenge quotidien → programmation par binôme (*pair-programming*)

Modèle agile : Crystal



Modèle agile : Crystal

- **Description des phases**
  - Spécification : classer et prioriser les besoins
  - Conception : choix des technologies et de l'architecture
  - Planification des fonctionnalités à réaliser
  - Itération de développement
    - Conception détaillée → plan d'action
  - Présentations périodiques
    - Eviter toute dérive et assurer la conformité avec les exigences des utilisateurs
    - **Documentation régulière** avant le lancement de chaque itération

Modèle agile : Crystal

- **Principe**
  - Communication, promiscuité des développeurs et les rencontres avec les utilisateurs

    → améliorer la qualité des échanges
  - Livraisons périodiques → fonctionnalités mises à disposition des utilisateurs
  - Peu de normes, voire quasi inexistantes mais **beaucoup de documents !**

Les modèles de cycle de vie

- **Modèles séquentiels**
  - Cascade, Intégration, En V
- **Modèles itératifs**
  - Incrémental, Spiral de Boehm
- **Modèles agiles**
  - ASD, DSDM, FDD, Crystal, **XP (M1), Scrum (M2 pro),** Lean
- D'autres modèles
  - RAD, UP

Conclusion

La production d'un projet informatique est :

- choisir un cycle de vie
- ne pas rester figé dans un modèle (**changer, adapter**)
- planifier (découpage, contrôles qualité, revues)
- autoriser une certaine **souplesse**
- prévoir : **ce qui n'est pas fait aujourd'hui risque d'être oublié demain…**
- assurer une forte communication entre tous les intervenants
- **adhérer à la méthodologie utilisée**
- faire preuve de bon sens

CV : Les questions à poser avant tout

- Quelle est votre vision stratégique à long terme ?
  - Travailler dans un domaine plutôt qu'un autre
  - Devenir expert en …..
  - Être ingénieur de logiciel pour quel type de projet …
  - …
- Quel est votre marché et comment est il segmenté ?
  - domaines d'activité,
  - types d'entreprises…
- Quels sont les besoins et les motivations de ces « clients »

Préparer la rédaction du CV

- **Anticiper** les questions du recruteur
- **Préparer** la suite (entretien) en dévoilant une partie de vos talents dans le CV afin de **donner envie de vous rencontrer**
- **Se mettre** à la place du recruteur
- Si candidature spontanée, rechercher les coordonnées exactes de la personne qui va recevoir votre courrier (preuve de votre intérêt)

Importance de CV

- DRH : Celui qui sélectionne les CV avec une grande vitesse !
  - Très peu de temps à la lecture (entre **30 et 60 secondes** par CV)
  - A vous de l'aider dans son travail

- En résumé un CV doit
  - Refléter fidèlement qui vous êtes
  - Être adapté (au poste et à l'entreprise)
  - **Se limiter à l'essentiel tout en donnant envie au lecteur d'en savoir plus…** Et cela passera par un entretien !

CV : la forme

- En général : une page
- Des paragraphes traditionnels (**ordre à définir en fonction du profil**)
  - Formations (anti-chronologique)
  - Expériences/stages/projets
  - Domaines de compétences
  - Divers
  - Fluidité nécessaire
- Attention à la présentation

Le CV : la forme

- Utilisation à bon escient des caractères gras (lisibilité « à bout de bras »)
- Une utilisation parcimonieuse de la couleur
- Police pas trop exotique…
- Évidemment pas de fautes d'orthographe !!
- Enregistrer en format PDF pour être lu par tous
- Enregistrer sous un nom explicite (CV-VotreNom) pour éviter d'être noyé dans l'anonymat des dossiers de votre interlocuteur

CV : le fond

- Le fond :
  - Expériences et contexte à valoriser
    - Les savoirs = connaissances
    - Les savoirs-faires = compétences
    - Les savoirs-être = plaisir, goût, motivation, points forts, qualités
  - Vos domaines de compétences sont, non seulement ce que vous maîtrisez le mieux, mais également ce que vous voulez faire!
  - La mise en valeur des expériences (projets, stages, jobs, responsabilités) est essentielle et encore plus importante s'il s'agit d'un CV à destination des pays anglo-saxons.

CV : le fond

- Les informations nécessaires pour décrire vos expériences :
  - Dates (durée en heures si projet d'étudiant)
  - Responsabilités ou métier exercé et quelques détails sur les tâches effectuées et les résultats obtenus
  - Nom de l'entreprise ou du projet
  - Localisation
  - Quantifier dès que cela est possible ;
    - Exemples :
      - Volume horaire : 500 h ,
      - CA réalisé : 15 000 €
      - Responsabilité d'encadrement de xx personnes

Avant entretien

- Dédramatiser l'entretien pour arriver décontracté
  - L'enjeu est aussi important pour vous que pour le recruteur
  - Le but de la rencontre est d'apprendre à se connaître
  - Se préparer oui, mais pas de « répétition »
  - Gagnez en assurance avec la visualisation positive

Avant entretien

- Les erreurs à éviter en entretien d'embauche
  - Les plus grosses erreurs:
    - Etre arrogant ou désintéressé
    - Parler en termes négatifs de son employeur précédent
  - Les erreurs courantes:
    - Ne pas donner suffisamment d'exemples de réalisations concrètes
    - Ne pas s'habiller correctement
    - Ne pas poser les bonnes questions
    - Trop cool ou trop stressé

Avant entretien

- Adapter votre apparence
  - L'habit fait le moine ?! Non
  - Mais 80% de la communication est non verbale:
    - Choix vestimentaires
    - Coiffure, maquillage
    - Position du corps
    - Gestuelle

Le jour de l'entretien

- Etre à l'aise
  - Arrivez à l'heure! Sans être essoufflé ou en sueur devant le recruteur
  - Allez en repérage quelques jours avant pour identifier le lieu du rendez-vous si nécessaire
  - Revoyez avant de partir vos éléments de l'information sur l'entreprise
  - Connaissez votre CV par cœur

Le jour de l'entretien

- Dans le hall d'accueil
  - Si vous fumez, évitez de « griller » une dernière cigarette avant d'entrer dans les locaux de l'entreprise → **Odeur de tabac**…
  - Soyez calme et souriez quand vous vous présentez à l'hôtesse → Premier contact très important
  - Allez vous asseoir en respirant calmement
  - Coupez votre portable…

Le jour de l'entretien

- Dans le **bureau** du recruteur
  - Saluez le recruteur par un simple bonjour et surtout, souriez ☺
  - Remerciez-le de vous recevoir et attendez qu'il vous invite à vous asseoir
  - Otez votre manteau, posez votre sac à vos pieds et installez-vous confortablement
  - Demandez la **permission** avant de prendre des notes
  - Essayez de vous détendre : un entretien sert à mieux vous **connaître**, pas à vous **piéger**…

Le jour de l'entretien

- Dans le **bureau** du recruteur
  - Regardez votre interlocuteur dans les yeux sans toutefois le fixer avec excès
  - Evitez de monopoliser ou de couper la parole
  - Présentez vos expériences professionnelles sans avoir le nez collé sur votre CV
  - Posez des questions au recruteur sur l'entreprise, les missions liées au poste, les projets à venir
  - Remerciez-le en souriant en fin de l'entretien
  - Pensez à saluer l'hôtesse qui vous a accueilli