COA
Méthodologies de Développement

Cedric Dumoulin

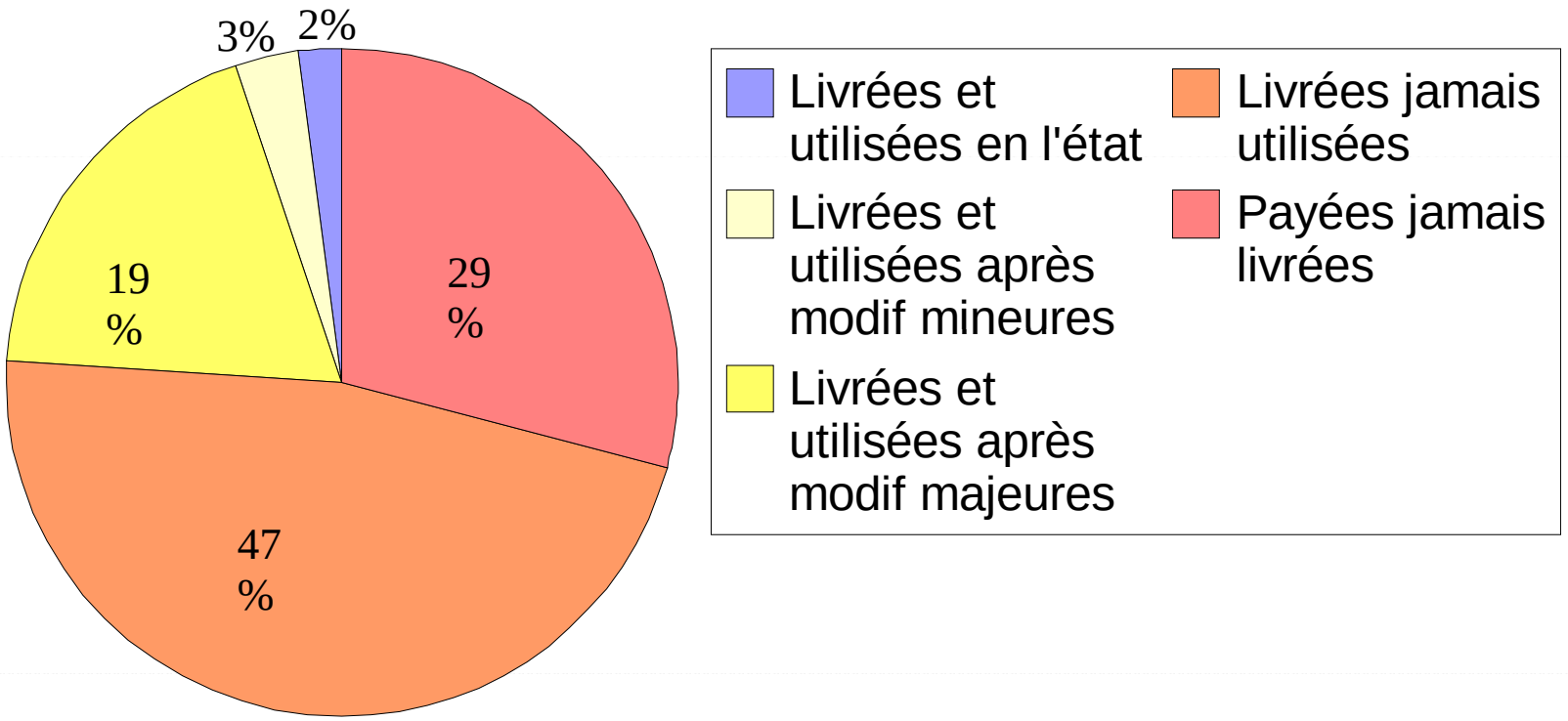
Pourquoi utiliser une méthodologie ?

Pourquoi une méthodologie ?

- Enquête du DOD 1980
  - Seul 2% des projets sont livrés et utilisés en état !
  - prés de 30% des projets sont payés et jamais livré !

- ➔ besoins de méthodes reproductibles

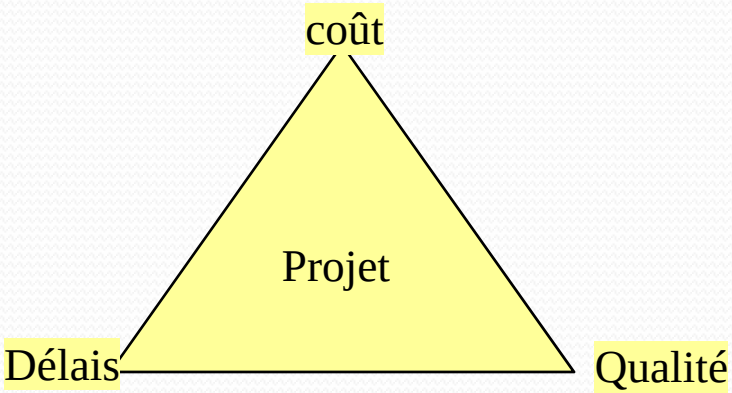
Enquête du DOD 1980



Standish group
Enquête de 1994

- Objectif: Comprendre les raisons des Echecs des projets Informatiques pour en déduire:
  - la portée des echecs de projets logiciels
  - les facteurs majeurs qui sont à l'origine des Echecs
  - les principaux facteurs de réussite a priori
- Enquête auprès des grosses, moyennes et petites entreprises
  - dans les secteurs de la banque, l'assurance, la sécurité, le BTP, la vente, la santé, des services travaillant tant au niveau local, que de l'état, ou de la fédération.
- 365 correspondants (8380 applications informatiques) vont participer à l'étude.

Présentation des résultats

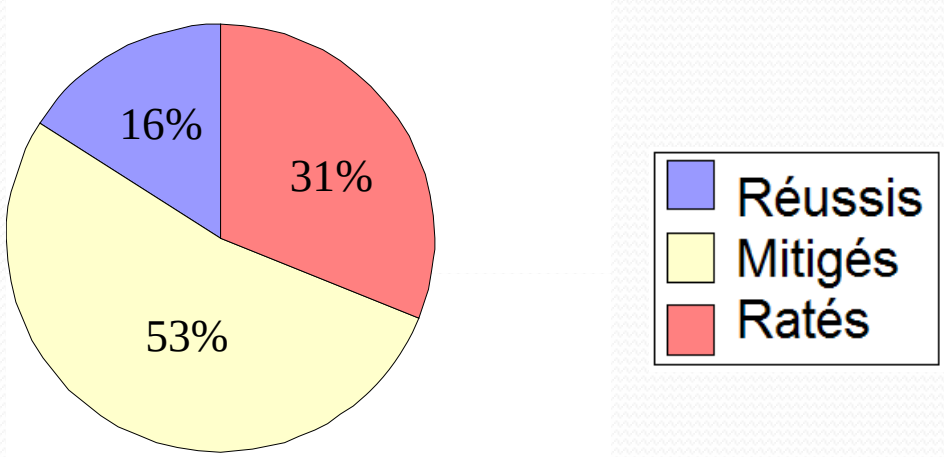


- **réussis**: le projet est achevé à temps, dans le cadre du budget prévu, conformément aux caractéristiques et fonctionnalités spécifiées (qualité 100%).

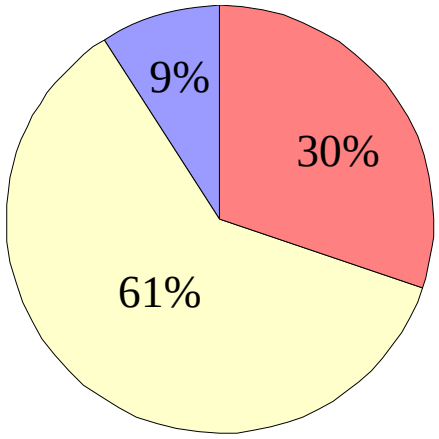
- **mitigés** : le projet est achevé et opérationnel mais un ou plusieurs des points suivants est constaté :
  - Il est hors délais
  - Il est hors budget (on ne compte pas les coûts induits)
  - sa qualité insuffisante (Les caractéristiques, fonctionnalités ne sont pas toutes disponibles)

- **projets ratés**: abandonné (après l'étude de faisabilité)

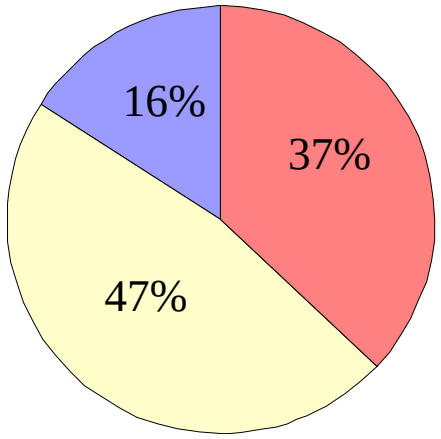
Résultats



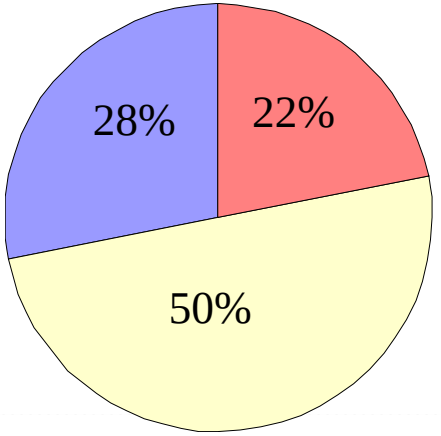
Très grosses entreprises



grosses entreprises



petites et moyennes entreprises



Les facteurs de réussite

- Découpage des livraisons
- Des estimations fiables et rigoureuses
- Des méthodologies formelles et utilisées
- Des spécifications précises et stables
- L'engagement de la direction
- l'Expérience du chef de projet
- L'implication des utilisateurs
- Des objectifs Métier clairs
- Résultats intermédiaires raisonnables
- Une infrastructure technologique normalisée
- compétence du personnel

Les facteurs de réussite (2000)

L'engagement de la direction	18
l'Expérience du chef de projet	14
L'implication des utilisateurs	16
Des objectifs Métier clairs	14
Résultats intermédiaires raisonnables	10
Une infrastructure technologique normalisée	8
Des spécifications précises et stables	6
Des méthodologies formelles et utilisées	6
Des estimations fiables et rigoureuses	5
Autres : Découpage des livraisons , compétence du personnel …	5

- On constate que les aspects techniques sont loin d'être essentiels et que la communication MOA & MOE est essentielle.

Constatation

- Relations (difficiles!?!)
  entre Maîtrise d'ouvrage (MOA) & Maîtrise d'oeuvre (MOE)

- **Années 60-70**: règne de l'informaticien- pas ou peu de MOA
- **Années 80** : Vision utilisateur.
  - Apparition de la MOA qui exprime les besoins dans un **cahier des charges**.
  - La MOE formalise, spécifie les fonctionnalités attendues (contrat entre MOA et MOE)
  - SI de gestion et de production.
- **Années 90**: Vision métier.
  - Le SI devient décisionnel.
  - La MOA a la responsabilité de formaliser les décisions métier, la MOE met en place techniquement les besoins exprimés par la MOA.
  - Pb: la MOA change souvent d'avis, manque de précision, ...
- **Années 2000**: Vers une vraie collaboration?
  - Le SI devient stratégique (tout le monde est concerné)
  - La MOA doit savoir exprimer ses besoins mais également organiser la mise à disposition des moyens (liés au métier) nécessaires à la MOE (utilisateurs,experts, ...)
  - La MOE formalise et explique comment elle compte s'y prendre

Recommandation

- Utiliser une méthodologie adaptée

Quelle méthodologie ?

Objectifs d'une méthodologie

- Gérer le cycle de vie de A à Z
- Gérer le risque
- Prendre en compte le changement
- Obtenir de manière répétitive des produits de qualité constante
- Organiser le travail

Cycle de vie

- cycle de vie =
  - étapes du développement d'un logiciel, de sa conception à sa disparition

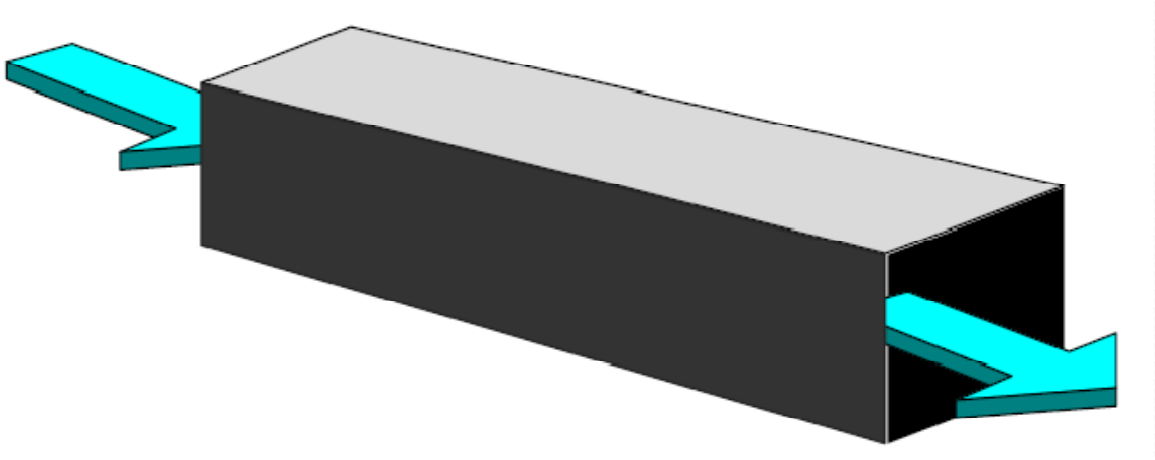
- **Définition des objectifs**, définir la finalité du projet et son inscription dans une stratégie globale.
- **Analyse des besoins et faisabilité**, expression, recueil et formalisation des besoins du demandeur (le client/MOA) et de l'ensemble des contraintes.
- **Conception générale**. élaboration des spécifications de l'architecture générale du logiciel.
- **Conception détaillée**, définir précisément chaque sous-ensemble du logiciel.
- **Codage** (Implémentation ou programmation), traduction dans un langage de programmation des fonctionnalités définies lors de phases de conception.
- **Tests unitaires**, vérifier individuellement que chaque sous-ensemble du logiciel est implémentée conformément aux spécifications.
- **Intégration**, s'assurer de l'interfaçage des différents éléments (modules) du logiciel. Fait l'objet de *tests d'intégration* consignés dans un document.
- **Qualification** (ou *recette*), vérification de la conformité du logiciel aux spécifications initiales.
- **Documentation**, produire les informations nécessaires pour l'utilisation du logiciel et pour des développements ultérieurs.
- **Mise en production**,
- **Maintenance**, comprenant toutes les actions correctives (maintenance corrective) et évolutives (maintenance évolutive) sur le logiciel.

- http://www.commentcamarche.net/contents/genie-logiciel/cycle-de-vie.php3

Modèles de cycles de vie

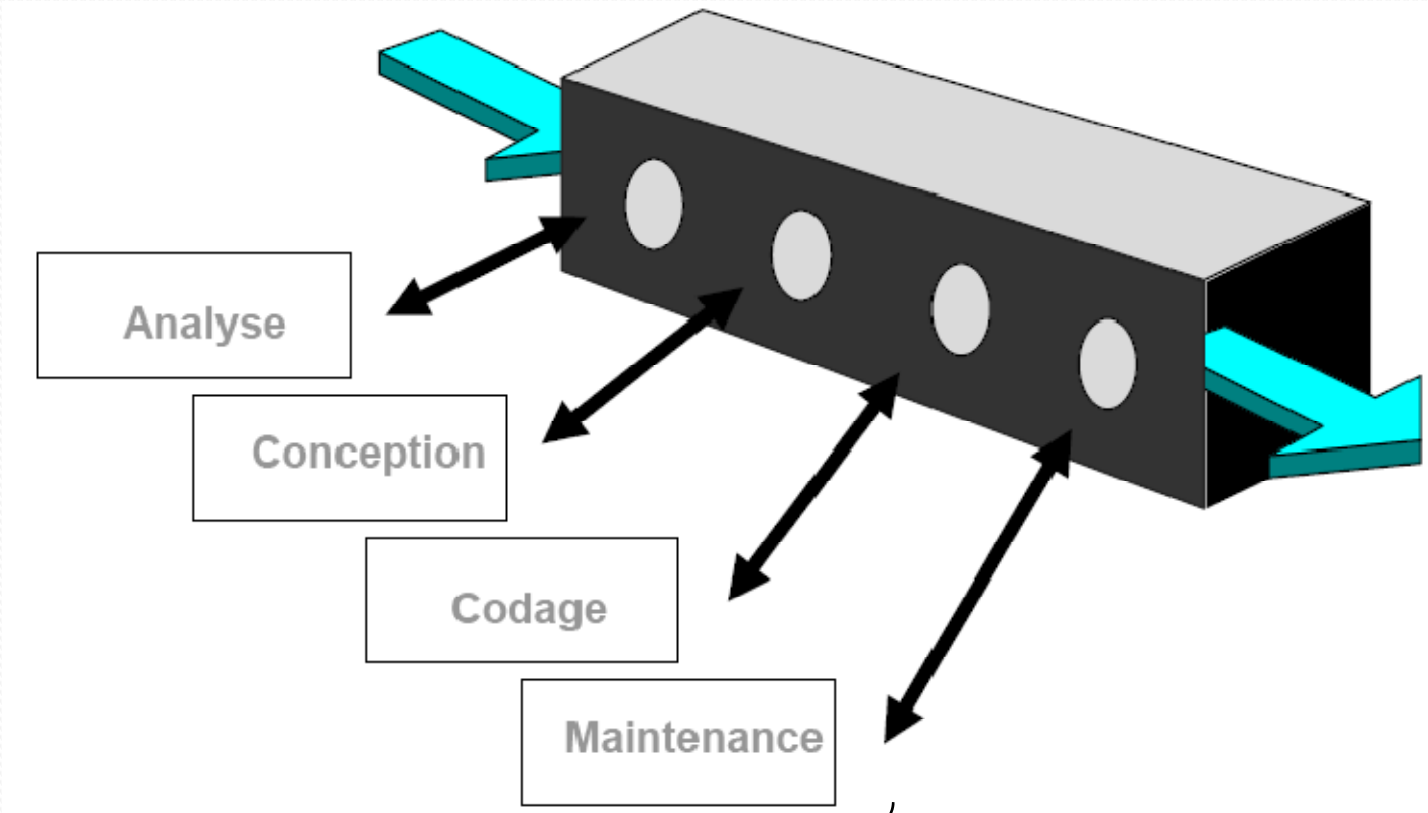
- Les cycles de vie linéaires
  - Le modèle en tunnel
  - Le modèle en cascade
  - Le modèle en V
- Limites des cycles de vie linéaires
- Cycles de vie itératifs

Le modèle en tunnel



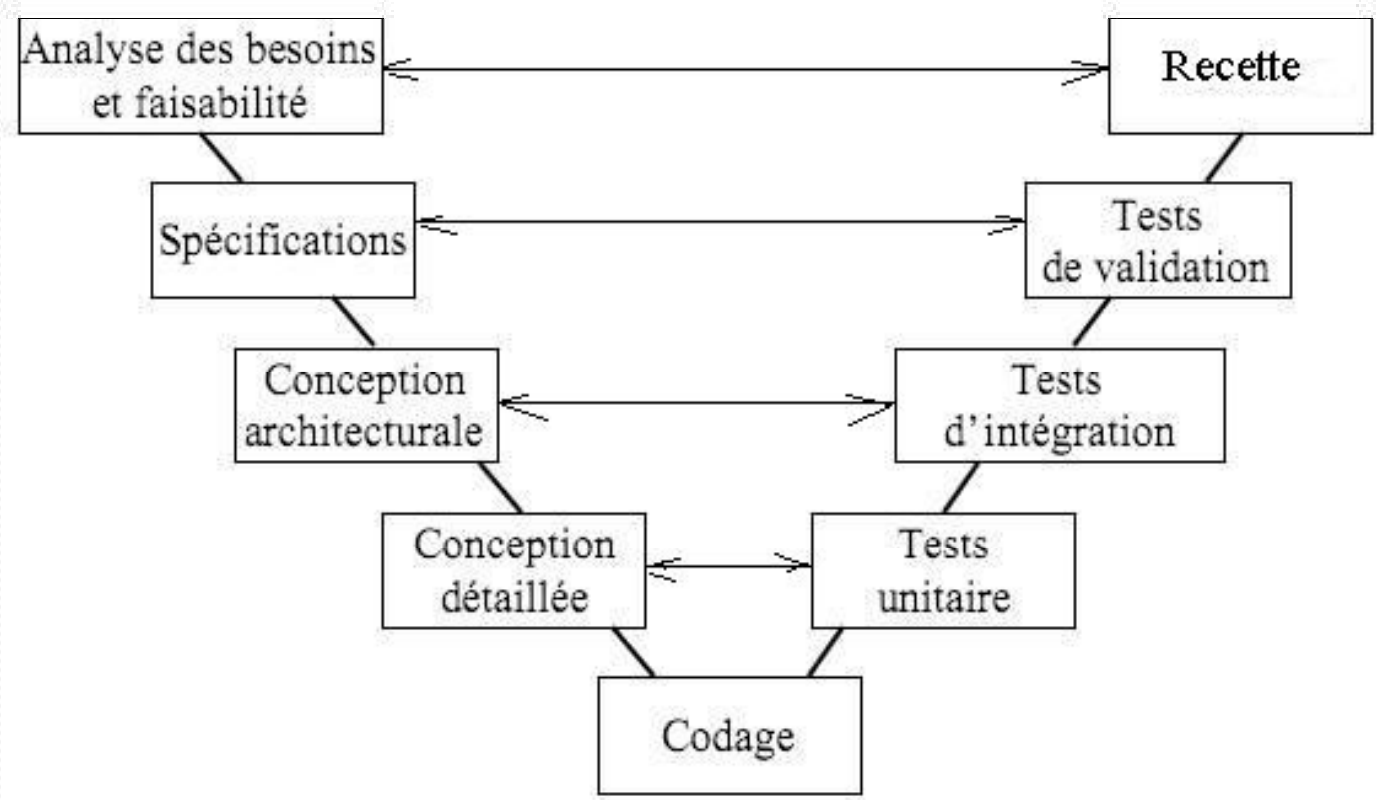
t_0 ⟶ ?

Le modèle en cascade



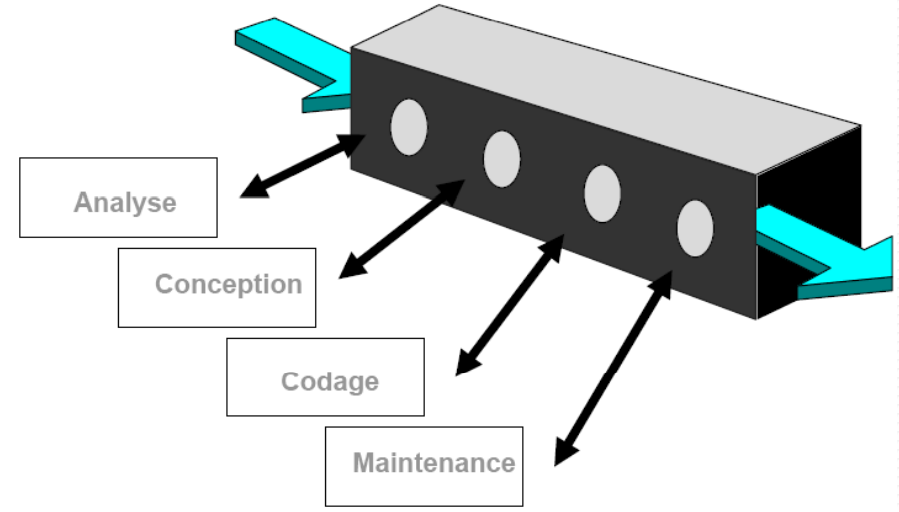
- Découpage en phases
  - retour en arrière limité
  - Mauvaise gestion des modifications et des erreurs

Le modèle en V



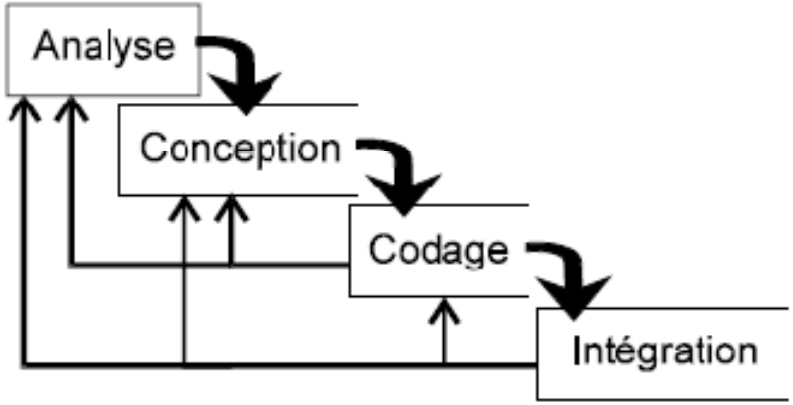
Caractéristiques du cycle de vie en cascade

- Linéaire, flot descendant
- Retour limité à une phase en amont
- Validation des phases par des revues
- Enchaînement depuis le cahier des charges jusqu'à la réalisation
- Bien adapté lorsque les besoins sont clairement identifiés et stables



Origines des risques liés au développement de logiciels

- Méconnaissance des besoins (client)
- Incompréhension des besoins (fournisseur)
- Instabilité des besoins
- Choix technologiques
- Mouvement de personnel



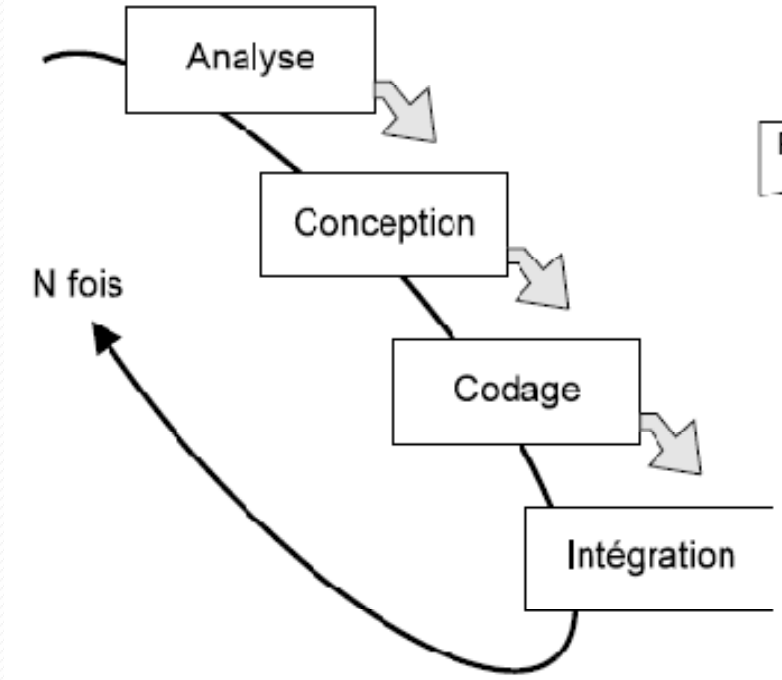
Amélioration du cycle de vie

- Construction du système par incréments
- Chaque itération a pour but de maîtriser une partie des risques et apporte une preuve tangible de faisabilité ou d'adéquation
- Enrichissement d'une série de prototypes
- Les versions livrées correspondent à une étape de la chaîne des prototypes

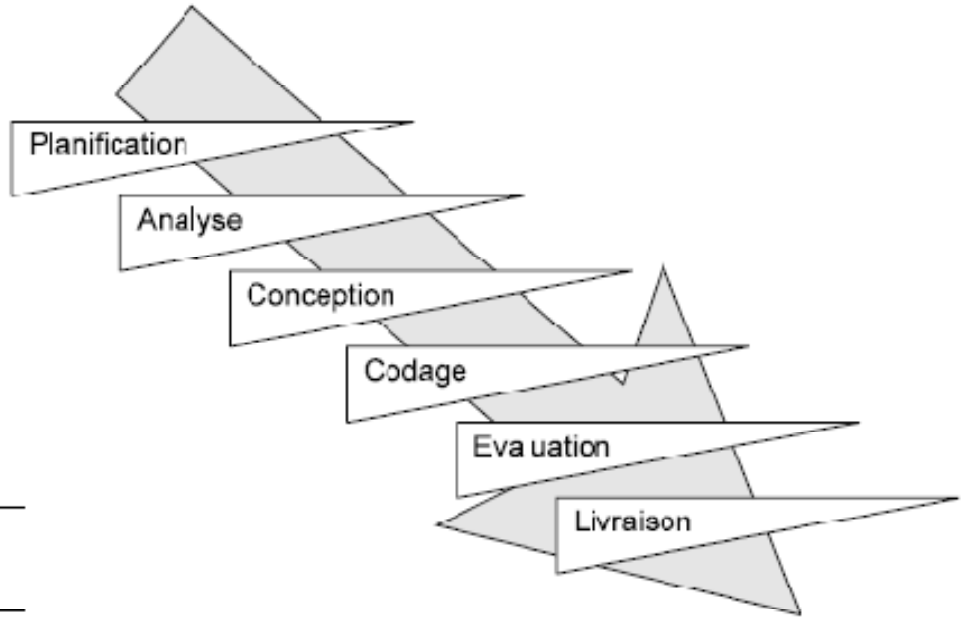
Cycle de vie itératif et incrémental

- Itératif : le processus de développement est appliqué plusieurs fois
- Incrémental : chaque itération augmente la quantité d'information
- Une amélioration du modèle en cascade

Une mini-cascade



Activités



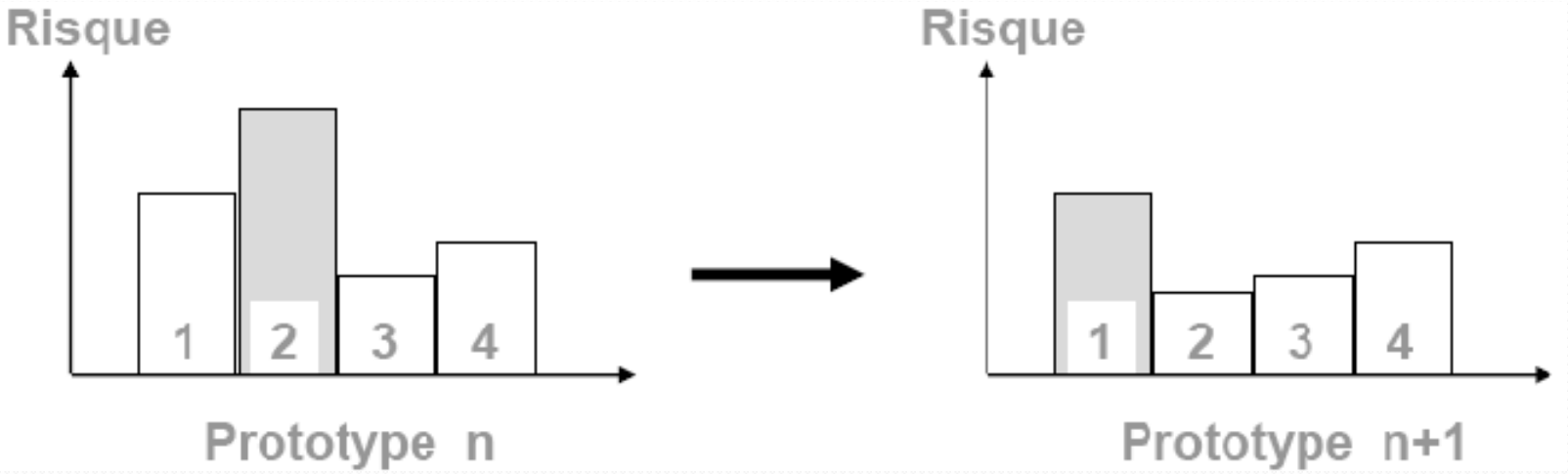
- Les activités sont remplacées par des phases :
- transition progressive entre phases

Approche itérative et incrémentale

- Segmentation du travail
- Concentration sur les besoins et les risques
- Les itérations sont des prototypes
  - Expérimentation et validation des technologies
  - Planification et évaluation
- Les prototypes « s'enroulent » autour du noyau de l'architecture

Risque et modèle itératif

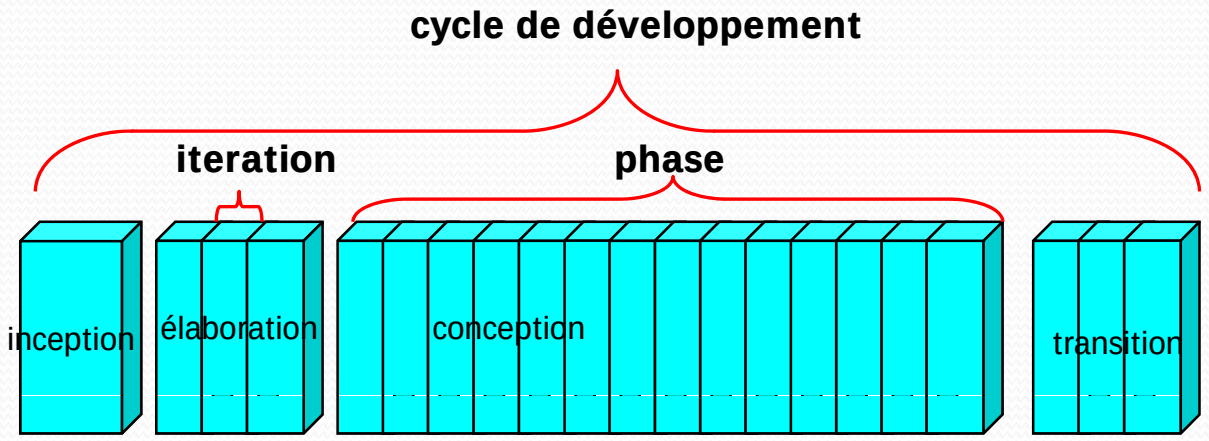
- Chaque prototype réduit une part du risque
- Un prototype est un programme exécutable qui peut s'évaluer quantitativement



Le processus unifié

Le processus unifié

- Processus itératif et incrémental
- Découpage en 4 phases
  - ex: inception, élaboration, conception, transition
- Elles même découpé en itération
- Chaque itération est composée d'activités
  - ex: analyse, conception, codage, intégration



Le Processus unifié
Les Phases

- Structuré en 4 Phases
  - **Opportunité** (***Inception***) : *Poser et comprendre le problème (MOA + MOE)*
  - **Élaboration**: *imaginer et comprendre la solution (implication de la MOA en baisse)*
  - **Construction**: *construire la solution (l'implication quasi exclusive MOE)*
  - **Transition**: *transférer la solution (MOA + MOE)*

- Le poids des phases:
  - l'effort se calcule en personne/mois ou hommes/jour
  - la durée en mois

	Opportunité	Elaboration	Construction	Transition
Effort	5%	20%	65%	10%
Durée	10%	30%	50%	10%

- Le passage par les 4 phases correspond à un cycle de développement et conduit à une génération de logiciel.

Le processus unifié
Les itérations

- Chaque phase se divise en itérations
- Une itération aboutit à un délivrable

- ➔ une itération regroupe plusieurs activités, par exemple :
  - Exigences
  - Conception
  - Implémentation, test, intégration et conception supplémentaire
  - Intégration finale et test système

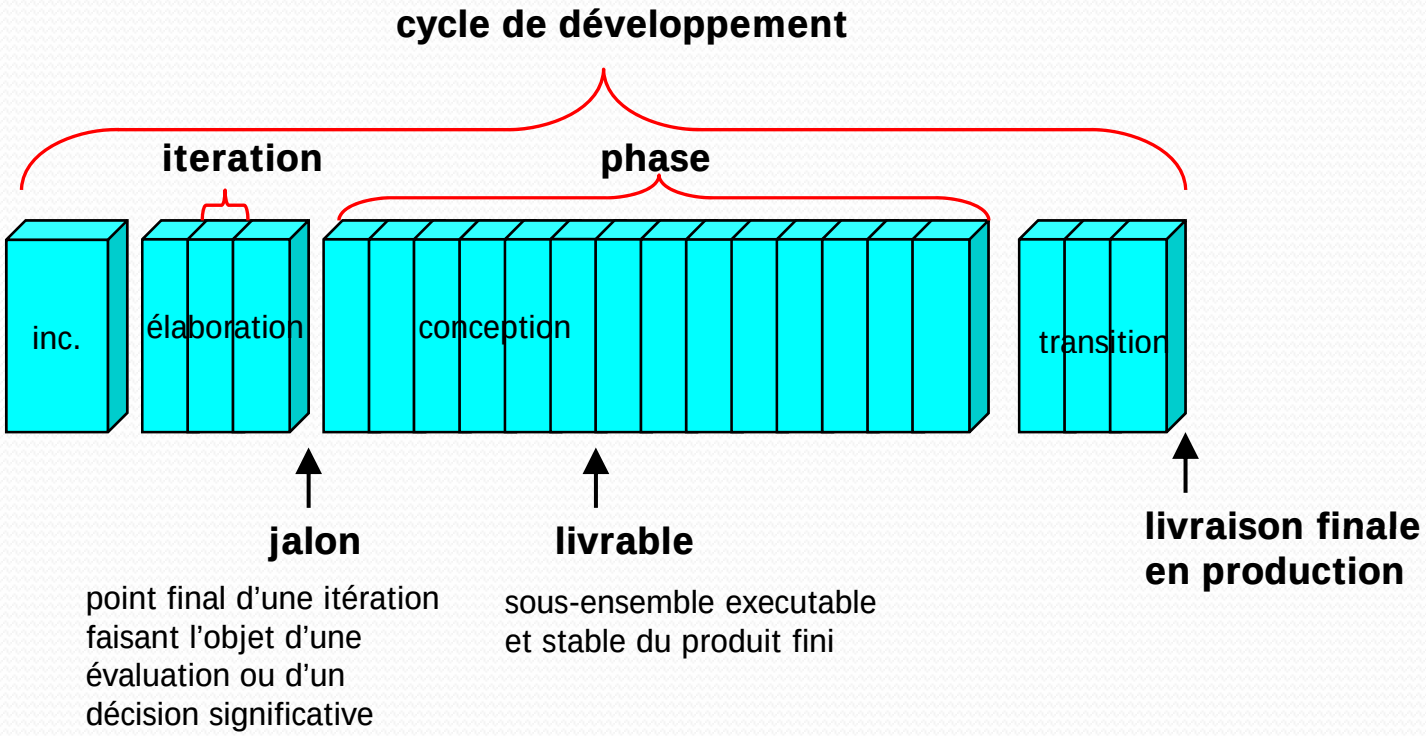
Le Processus unifié
Les Activités

- Chaque activité consomme et produit des **diagrammes UML**.
- Chaque activité consomme et produit des **documents**
- **5 activités de bases**
  - **Modélisation métier** est une abstraction de la structure que l'on veut étudier et pour laquelle on veut réaliser un projet. C'est un travail d'audit. C'est un état des lieux
    - modèle du domaine
  - **Ingénierie des besoins** **(Expressions des besoins/Exigence)**
    - modèle de Cas d'utilisation
  - **Analyse & conception**
    - Passage d'une vue externe du système à une vue interne, Choix de l'architecture logicielle
    - Modèle de conception
  - **Implémentation**
  - **Tests**
  - **Déploiement**

Le Processus unifié
Les Activités

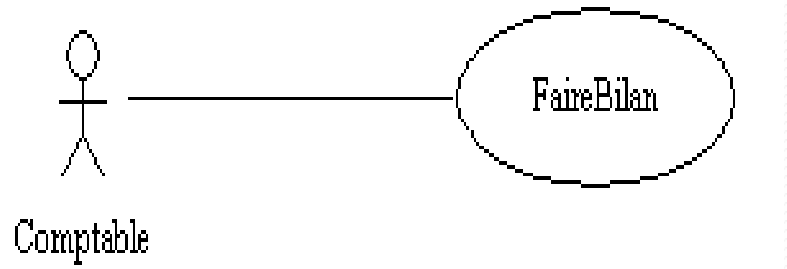
- Autres activités
  - **Les activités de support**
    - amélioration de l'environnement du projet: guide,normes, patrons, standards, outils logiciels (AGL, subversion,..) et matériels.
    - Gestion des configurations et des changements
  - **Les activités organisationnelles**
    - Planification (pert, Gantt, WBS,...).
    - Coût
    - Assurance Qualité (externe au projet), rédaction du PAQ

Phases, activités et itérations



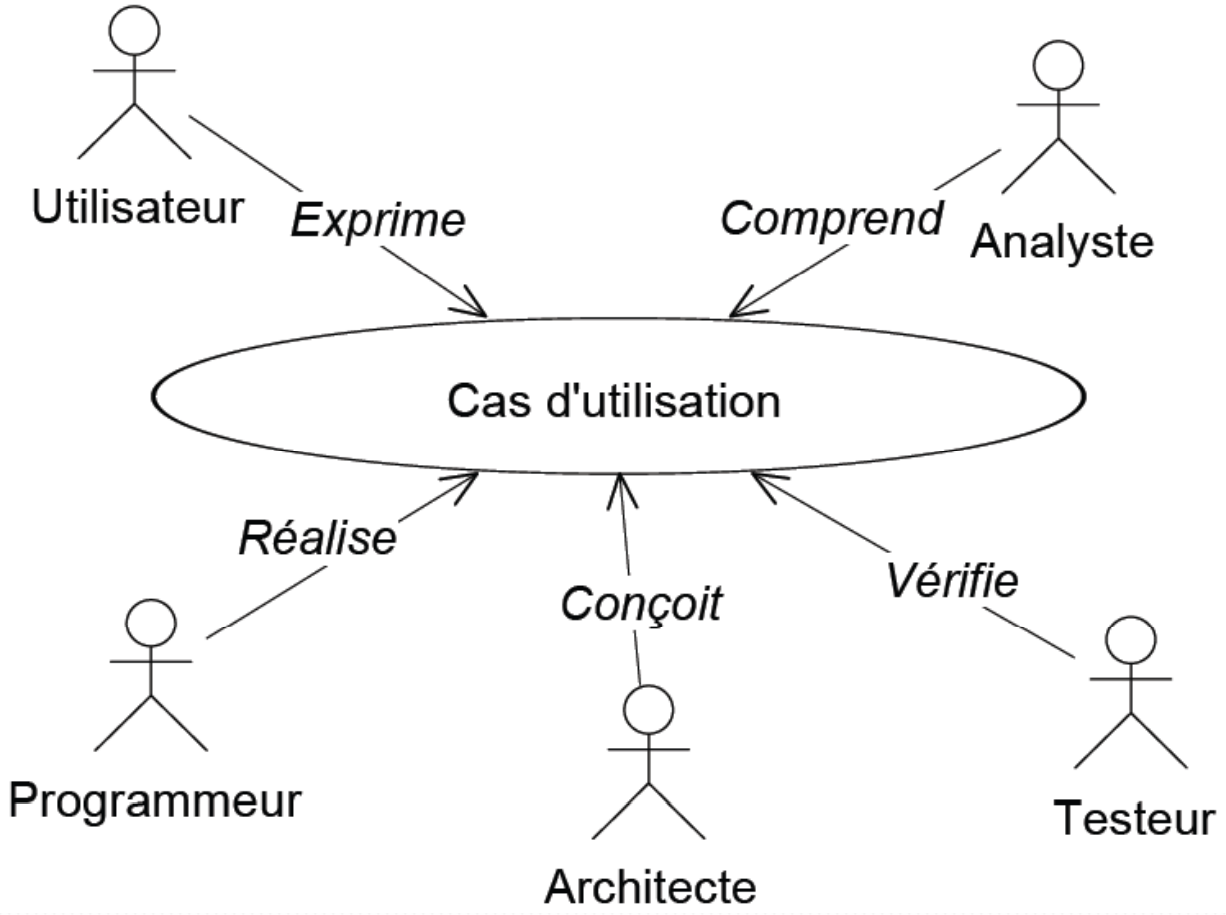
Le Processus unifié conduit par les Cas d'Utilisation

- Un cas d'utilisation (UC : Use Case)
  - représente un scénario d'utilisation de l'application
  - est traduit par une fonctionnalité de l'application déclenché le plus souvent par un utilisateur du logiciel



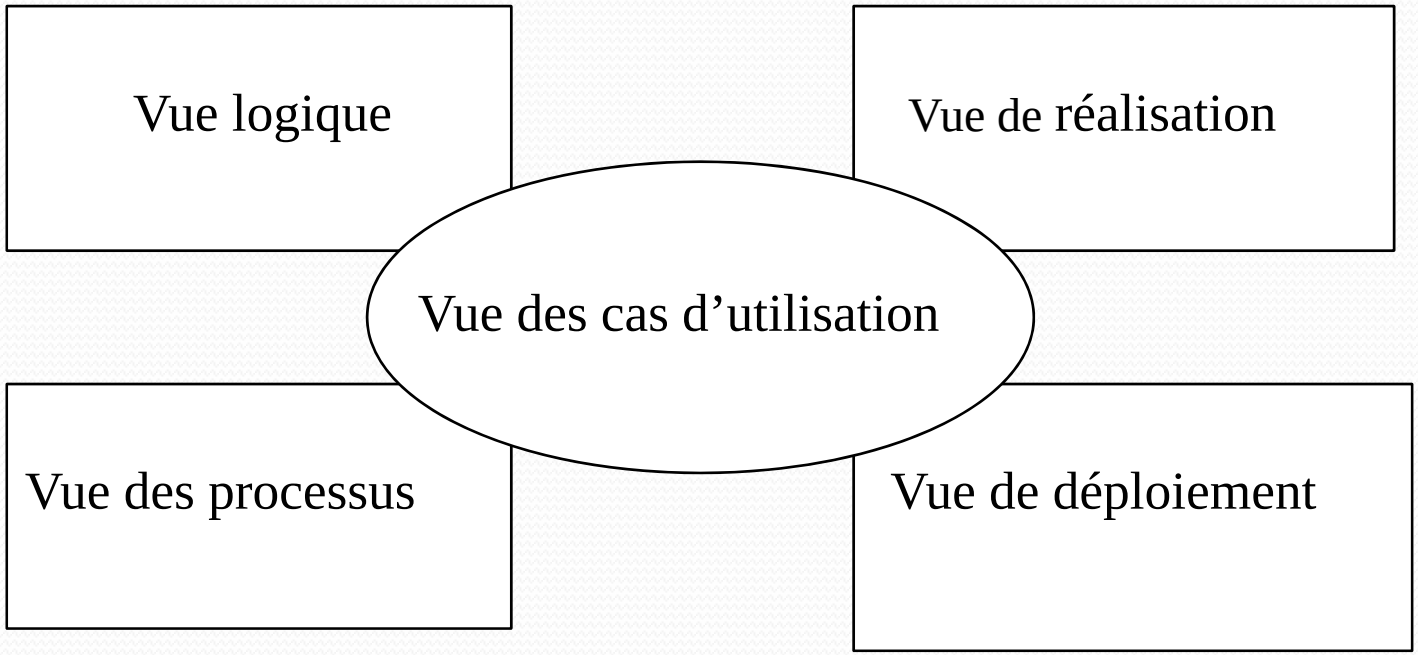
- Les cas d'utilisation
  - Structurent le système développé
  - Définissent le planning des itérations
  - Donnent la trame de la documentation utilisateur
  - Rythment le déploiement
  - Fédèrent les différents modèles (UML)

Le Processus unifié conduit par les UC



- Les UC sont au centre du Processus Unifié.

Le Processus unifié centré sur l'architecture



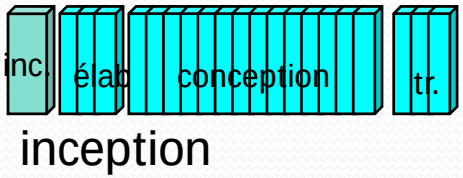
- Les autres vues sont des vues différentes sur les cas d'utilisation
- Elles représentent les cas d'utilisation de façon plus adapté aux objectifs de la vue

Le Processus unifié piloté par les risques

- Risque = Événement qui a une probabilité non nulle de se produire et qui affecte la réussite du projet.

- La réduction des risques doit être le fil conducteur du découpage en itérations
- Risques **majeurs**:
  - Risque de l'inadéquation besoins <-> développement
  - Risque humains (conflits, manque de compétence,…)
  - Risque incapacité de l'architecture à répondre aux Contraintes opérationnelles

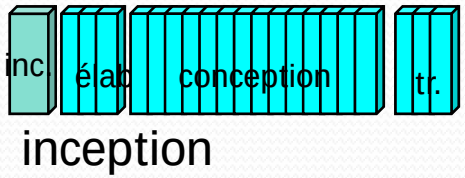
Phase d'inception (opportunité)



- Concerne MOA & MOE
- Poser et Comprendre le problème
  ➔Jalon «Objectifs»

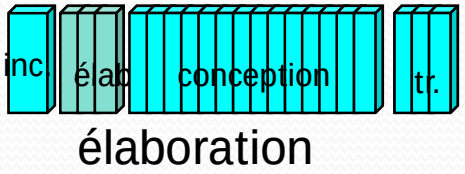
- Objectifs principaux
  - Partager la même compréhension du problème
    - Entre maîtrise d 'ouvrage et maîtrise d 'œuvre
    - En interne entre les membres des équipes (MOA-MOE)
  - Déterminer les objectifs à atteindre
  - Valider la faisabilité

Autres objectifs



- Déterminer la portée, le périmètre du système à développer
  (Ce qui doit être produit et ce qui ne doit pas l'être)
- Décrire comment le système sera utilisé
  - Déterminer les conditions limites d 'utilisation
  - Critères d 'acceptation
- Étudier les risques
- Préparer l'environnement de développement du projet:
  - Estimation du coût global et de la durée globale
  - Planification, organisation des ressources
  - Préparation (achat) des outils à utiliser
  - Compromis sur la conception: Que fait-on ?
    Que réutilise-t-on? Que sous-traite-t-on ? ...

Phase d ’élaboration

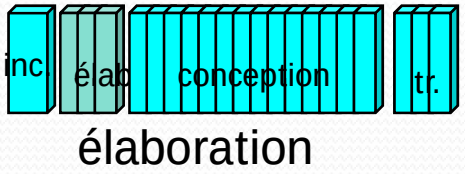


- concerne MOA & MOE
- Imaginer et comprendre la solution
  ➔Jalon «Architecture»

- Objectifs principaux
  - Déterminer la base de l ’architecture logicielle
  - Valider sa stabilité
  - Valider sa concordance aux besoins

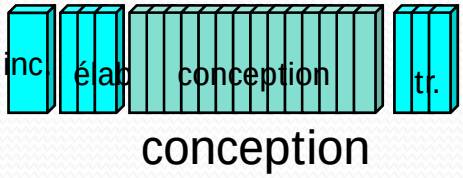
Phase d ’élaboration

Autres objectifs



- Stabiliser les besoins
- Etablir les cas de tests
- Affiner les priorités dans la réalisation les besoins
- Affiner les coûts et les délais
- Trouver le compromis satisfaction des besoins / choix techniques
- Identifier les opportunités de réutilisation de composants existants
- Décrire et valider l’architecture
- Choisir l’environnement de développement
- Planifier la mise en place des outils (AGL,..)
- Sélectionner les composants :
  - Faire ? Acheter ? Réutiliser ? Modifier ?

Phase de construction

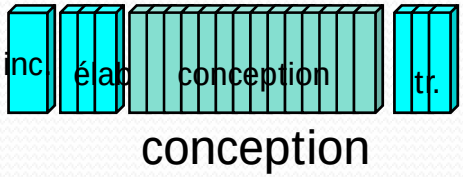


- concerne MOE
- Construire la solution
  ➔Jalon «1er produit opérationnel)

- Objectifs principaux :
  - Produire un logiciel utilisable conforme aux besoins
  - Confronter ce logiciel aux critères d'acceptation (élaborés pendant la phase d'opportunité)

Phase de construction

Autres objectifs

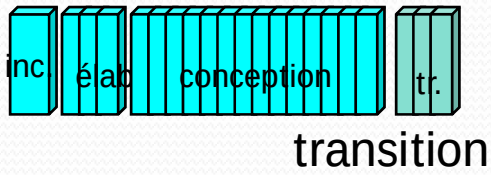


- Développer les composants manquants
- Tester l'application
- Minimiser les coûts de développement (lutter contre le gaspillage, code inutile, fait plusieurs fois,…)
- Préparation de la transition (procédures d'installation, tester la stabilité et la maturité du produit)

- Les tests:
  - On repart des jeux de tests établis lors de l'analyse des besoins.
  - Choisir une stratégie de tests
  - Tester une par une les fonctionnalités (tests unitaires)
  - Tester l'application (tests d'intégration)
  - Mettre en place les alpha et les béta-tests

  - Attention aux tests de non régression!

Phase de transition

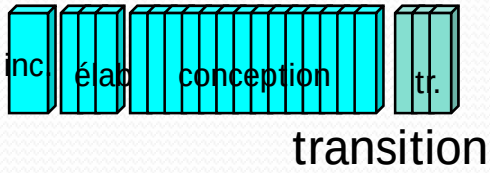


- concerne MOA & MOE
- Transmettre la solution ➔ jalon «génération»

- Objectifs généraux :
  - S ’assurer que le logiciel est opérationnel
  - S ’assurer qu ’on est capable de le livrer
  - Rendre les utilisateurs autonomes

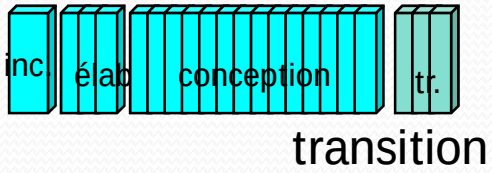
Phase transition

Autres objectifs



- Vérifier la satisfaction des objectifs des décideurs
- Vérifier la satisfaction des besoins (des exigences) des utilisateurs
- Gérer le retour des utilisateurs
- Finaliser le matériel de support (notices)
- Former les utilisateurs
- Planifier le déploiement
- Ajuster l ’installation aux différents sites

Phase transition
Autres objectifs



- Remarques
  - La transition
    - Peut-être relativement simple
      - Nouvelle version d 'un traitement de texte
    - Peut-être incroyablement complexe
      - Nouveau système de contrôle aérien
  - L 'ancien système doit être désactivé « proprement »
    - Coût parfois très élevé

Ce qu'il faut retenir

Conclusion

- Objectif du module de COA
  - Analyser et concevoir une application
  - En Appliquant le Processus Unifié piloté par les Cas d'Utilisation

Lectures

- Obligatoire (pour mercredi !!)
  - UML 2: Initiation, exemples et exercices corrigés
    - Chap. 3 « Les concepts de l'approche objet »
    - Chap. 4 « Modélisation des exigences (sauf représentation textuelle) »

Questions

Définitions

Maîtrise d’ouvrage (MOA)

- Exprime un besoin qui devient l'objectif du projet
- Élabore le Cahier des Charges fonctionnelle
- Prépare les cas de tests fonctionnels
  - Vérifier que les développements/paramétrages effectués par la MOE fonctionnent
- Exemple de MOA
  - le cas d'un projet d'implémentation du module CO (contrôle de gestion) du progiciel SAP, la MOA est constituée des contrôleurs de gestion impliqués sur le projet.

Maîtrise d'oeuvre (MOE)

- Répond informatiquement au besoin exprimé par la MOA
- Rédige une réponse au besoin
  - CDC technique (cahier des charges technique)
  - Dossier de paramétrage ou
  - Dossier de conception général.
- Réalise développements/paramétrages nécessaires

- Exemple de MOE
  - Un service informatique en interne et dédié au projet ou une SSII à qui l'entreprise en charge du projet sous-traite intégralement les développements informatiques

Les livrables

- Un livrable est tout résultat, document, mesurable, tangible ou vérifiable, qui résulte de l'achèvement d'une partie de projet ou du projet
- Exemples
  - Un cahier des charges et une étude de faisabilité sont des livrables