

Méthodologie de Conception

CH3: Description et caractéristique du processus unifié (UP)

Rappel:

- ❑ **Unified process** ou processus unifié en français est un processus de **conception** et de **réalisation** de logiciels développés avec des langages de programmation **orientés objet** (Java par exemple).
- ❑ C'est un guide méthodologique pour réaliser des logiciels en **conseillant** et **pilotant** l'équipe dans ses différentes activités pour réduire la complexité du projet .

Les caractéristiques du processus unifié (UP)

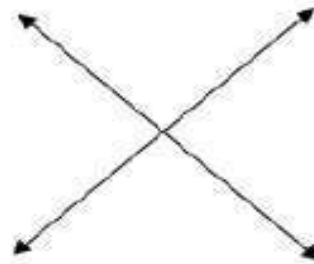
Les 4 grandes caractéristiques du Processus Unifié sont liées

Processus
itératif et
incrémental

Piloté par
les risques

Piloté par
les cas
d'utilisation

Centré sur
l'architecture



❖ Itérative et incrémentale :

Contrairement à la méthode MERISE qui est séquentielle.

Itérative puisque chaque phase de la démarche UP comprend des itérations.

Chaque phase est ponctuée par un jalon qui marquera la décision que les objectifs initiaux ont été remplis.

Une itération a pour but de maîtriser une partie des risques et apporte une preuve tangible de faisabilité.

Une itération est un cycle de développement logiciel (ou système) complet depuis le recueil des besoins jusqu'à l'implantation et aux tests.

L'**itération** se termine par la sortie d'une version exécutable du projet, c'est à dire un prototype (exécutable, testé et intégré) avec une qualité égale à celle d'un produit fini et pouvant être évalué.

Incrémentale puisqu'il y a un incrément du projet par itération, c'est à dire que le logiciel et le modèle évoluent suivant des incréments. C'est le développement d'une série de prototypes qui vont en s'améliorant

❖ UP piloté par les cas d'utilisation

❑ L'objectif principal d'un système logiciel étant de **rendre service à ses utilisateurs, cela nécessite de bien comprendre leurs désirs et leurs besoins.**

➡ Les cas d'utilisation UML :

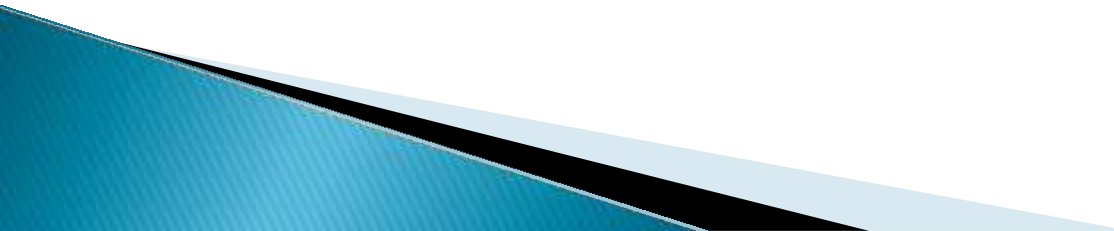
- ✓ décrivent les besoins fonctionnels, et leur ensemble constitue **le modèle des cas d'utilisation décrivant les fonctionnalités complètes du système**

- ✓ vont complètement **guider le processus de développement à travers** l'utilisation de modèles basés sur l'utilisation du langage UML

- ✓ **garantissent la cohérence du processus de développement du système**

- ✓ **doivent absolument être développés avec l'architecture du système.**

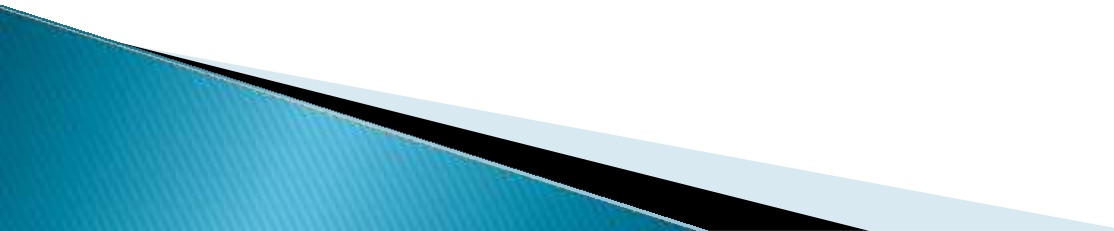
Liens entre cas d'utilisation et architecture ?

- ☐ Tout produit est à la fois forme et fonction.
Les cas d'utilisation doivent une fois réalisés, trouver leur place dans l'architecture.
 - ☐ L'architecture doit prévoir la réalisation de tous les cas d'utilisation.
 - ☐ L'architecture et les cas d'utilisation doivent évoluer de façon concomitante.
- 

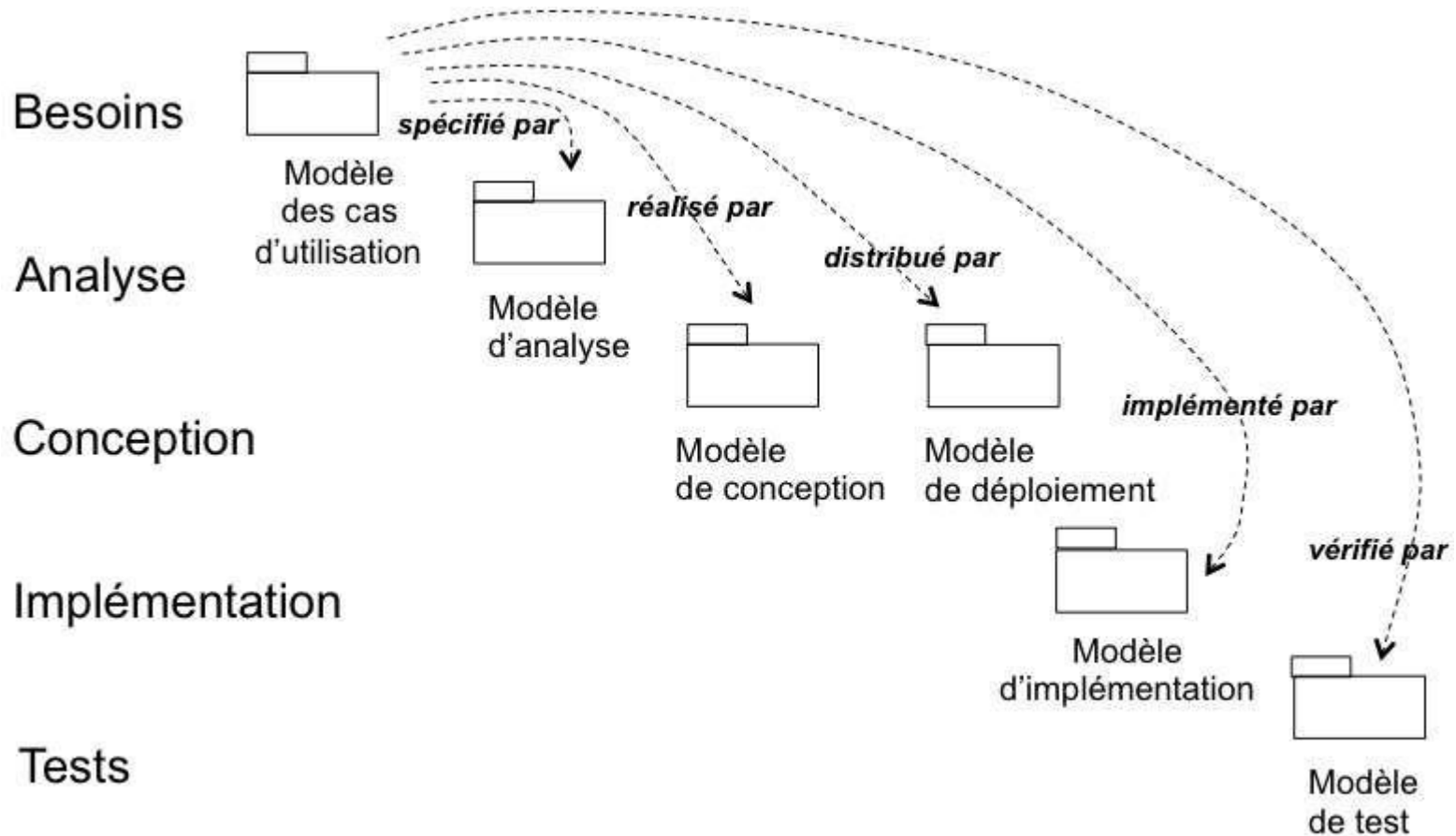
A partir du modèle des cas d'utilisation, les développeurs créent une série **de modèles de conception et d'implémentation** réalisant les cas d'utilisation,

- Chacun des modèles successifs est ensuite révisé pour **en contrôler la conformité par rapport au modèle des cas d'utilisation**,

- Enfin, les testeurs testent l'implémentation pour s'assurer que les **composants du modèle d'implémentation** mettent correctement en œuvre les cas d'utilisation.



Les CU lient les modèles



A partir des cas d'utilisation, Les développeurs créent une série de modèles UML.

Modèle des cas d'utilisation	Expose les cas d'utilisation et leurs relations avec les utilisateurs
Modèle d'analyse	Détaille les cas d'utilisation et procède à une première répartition du comportement du système entre divers objets
Modèle de conception	Définit la structure statique du système sous forme de sous système, classes et interfaces ; Définit les cas d'utilisation réalisés sous forme de collaborations entre les sous systèmes les classes et les interfaces
Modèle d'implémentation	Intègre les composants (code source) et la correspondance entre les classes et les composants
Modèle de déploiement	Définit les nœuds physiques des ordinateurs et l'affectation de ces composants sur ces nœuds.
Modèle de test	Décrit les cas de test vérifiant les cas d'utilisation
Représentation de l'architecture	Description de l'architecture

❖ UP est piloté par les risques

Qu'est-ce qu'un risque ?

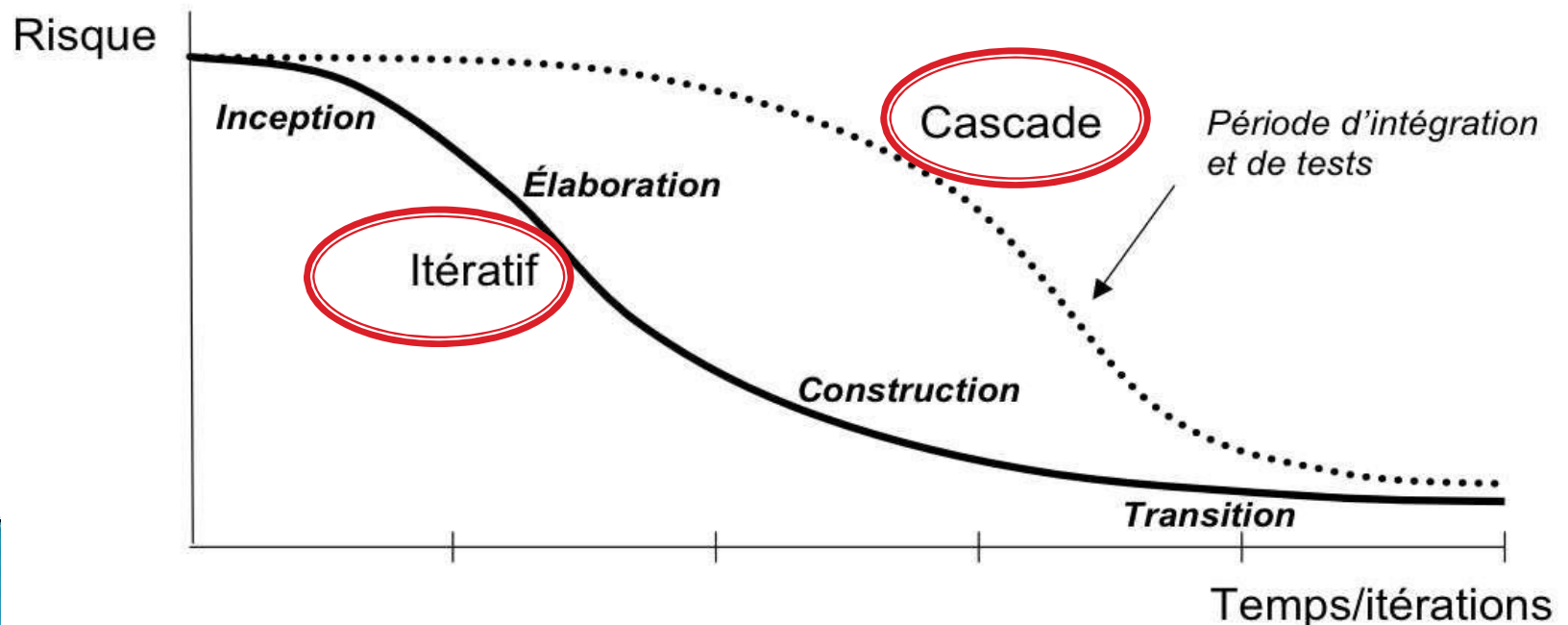
- Risque que le projet de construction du système soit un echec
- Nature des risques
 - besoins / technique / autres
- Exemples
 - le système construit n'est pas le bon
 - architecture inadaptée, utilisation de technologies mal maîtrisées, performances insuffisantes
 - personnel insuffisant, problèmes commerciaux ou financiers (risques non techniques, mais bien réels)

Gestion des risques

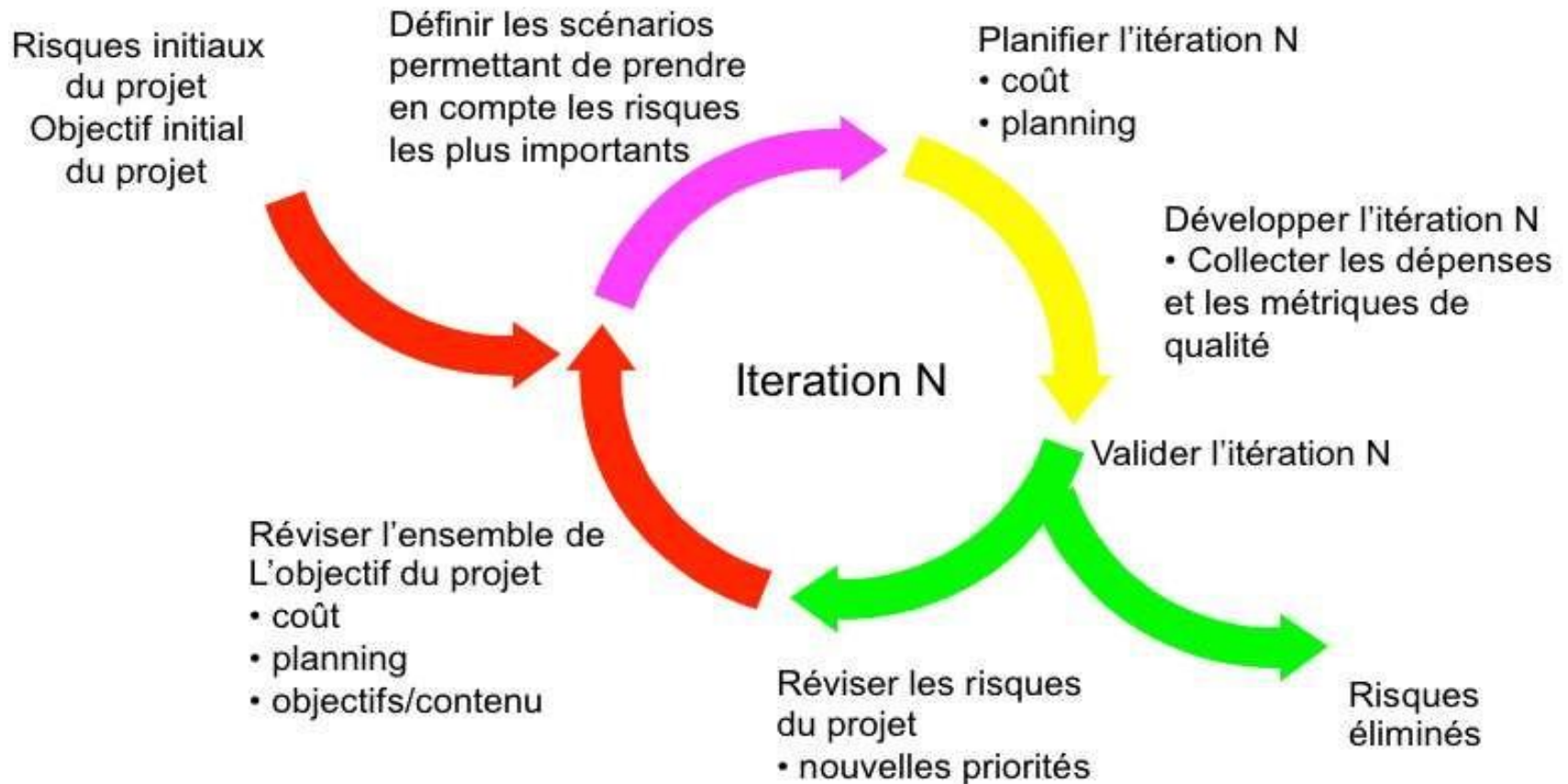
- Identifier et classer les risques par importance
- Agir pour diminuer les risques
 - Ex. changer les besoins, confiner la portée à une petite partie du projet, faire des tests pour vérifier leur présence et les éliminer
- S'ils sont inévitables, les évaluer rapidement
 - tout risque fatal pour le projet est à découvrir au plus tôt

Pilotage par les risques et itérations

- Identifier, lister et évaluer la dangerosité des risques
- Construire les itérations en fonction des risques
 - s'attaquer en priorité aux risques les plus importants qui menacent le plus la réussite du projet
 - « provoquer des changements précoces »
 - ex. stabiliser l'architecture et les besoins liés le plus tôt possible



Itérations pilotées par la réduction des risques



Itérations et risques

- On ordonne les itérations à partir des priorités établies par les CU et l'étude du risque
 - Plan des itérations
 - Chaque prototype réduit une part du risque et est évalué comme tel
 - Les priorités et l'ordonnancement de construction des prototypes peuvent changer au court du temps

Exemple : étude de cas

- Une application classique de commerce électronique : **La librairie en ligne**
les fonctionnalités de cette application sont inspirés de sites existants, comme www.amazon.fr,
- les clients peuvent acheter des ouvrages sur le web en les ajoutant à leur panier électronique.
Lorsque les clients souhaitent régler leurs achats, ils valident leur panier. Le prix du panier est alors calculé et le paiement de la commande peut être effectué.

Le chef de projet doit donc prendre en compte de façon combinée la priorité fonctionnelle et l'estimation du risque :

- Si la priorité est haute et le risque également, il faut planifier le cas d'utilisation dans une des toutes premières itérations (exemple : Maintenir le catalogue).
- Si la priorité est basse et le risque également, on peut reporter le cas d'utilisation à une des toutes dernières itérations (exemple : Consulter l'aide en ligne).
- si les deux critères sont antagonistes ! Le chef de projet doit alors décider en pesant le pour et le contre. Il peut être amené à négocier avec le client pour le convaincre qu'il vaut mieux pour le projet traiter en premier un cas d'utilisation risqué mais peu prioritaire, au lieu d'un cas d'utilisation plus prioritaire mais ne comportant pas de risque.

Cas d'utilisation	Priorité	Risque
Chercher des ouvrages	Haute	Moyen
Gérer son panier	Haute	Bas
Effectuer une commande	Moyenne	Haut
Créer un compte client	Haute	Bas
Consulter ses commandes	Basse	Moyen
Consulter l'aide en ligne	Basse	Bas
Gérer son compte client	Moyenne	Bas
Maintenir le catalogue	Haute	Haut
Maintenir les informations éditoriales	Moyenne	Bas
Maintenir le site	Moyenne	Bas

❖ UP est Centré sur l'architecture :

Tout système complexe doit être décomposé en parties modulaires afin de garantir une maintenance et une évolution facilitées.

Recherche de la forme générale du système dès le début

- Approche systématique pour trouver une « bonne » architecture qui soit :
 - support des cas d'utilisation
 - adaptable aux changements
 - pour et avec la réutilisation
 - compréhensible intuitivement

❖ UP est Centré sur l'architecture :

L'architecture émerge des besoins de l'entreprise, tels qu'ils sont exprimés par les utilisateurs et autres intervenants et tels qu'ils sont reflétés par les cas d'utilisation

Cette architecture doit prévoir la réalisation de tous les cas d'utilisation.

 L'architecture et les cas d'utilisation évoluent donc de façon concomitante.

Dès le démarrage du processus, on aura une vue sur l'architecture à mettre en place.

Tout produit est à la fois forme et fonction. Les cas d'utilisation doivent une fois réalisés, trouver leur place dans l'architecture. L'architecture doit prévoir la réalisation de tous les cas d'utilisation. L'architecture et les cas d'utilisation doivent évoluer de façon concomitante.

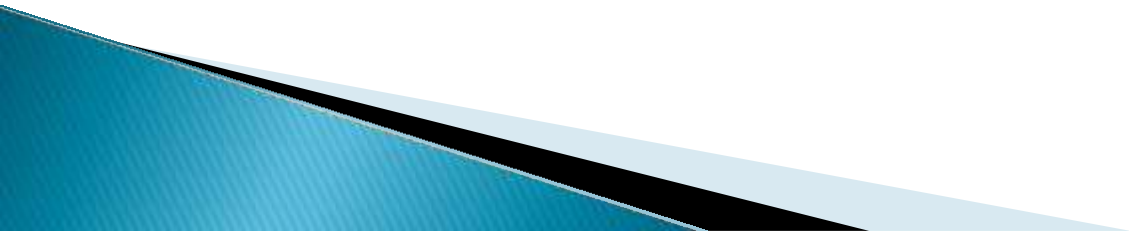
Processus centré sur l'architecture

- L'architecture sert de lien pour l'ensemble des membres du projet
 - réalisation concrète de prototypes incrémentaux qui « démontrent » les décisions prises
 - Vient compléter les CU comme « socle commun »
- **Contrainte de l'architecture**
 - plus le projet avance, plus l'architecture est difficile à modifier
⇒ les risques liés à l'architecture sont très élevés, car très coûteux
- **Objectif pour le projet**
 - établir dès la phase d'élaboration des fondations solides et évolutives pour le système à développer, en favorisant la réutilisation
 - l'architecture s'impose à tous, contrôle les développements ultérieurs, permet de comprendre le système et d'en gérer la complexité
- L'architecture est contrôlée et réalisée par l'architecte du projet

Objectif / architecture

- Construire une architecture
 - Comme forme dans laquelle le système doit s'incarner
 - les CU réalisés doivent y trouver leur place / la réalisation
 - la réalisation des CU suivant doit s'appuyer sur l'architecture
 - Qui permet de promouvoir la réutilisation
 - Qui ne change pas trop
 - converger vers une bonne architecture rapidement

Les activités de UP




Expression des besoins


L'expression des besoins comme son nom l'indique, permet de définir les différents besoins :

- définir les **besoins principaux** et fournir une liste de **leurs fonctions**
- recenser les **besoins fonctionnels** (du point de vue de l'utilisateur) qui conduisent à l'élaboration des modèles de cas d'utilisation
- appréhender les **besoins non fonctionnels (technique)** et livrer une liste des exigences.

Le modèle de cas d'utilisation présente le système du point de vue de l'utilisateur et représente sous forme de cas d'utilisation et d'acteur, les besoins du client.



Analyse

- ▮ L'objectif de l'analyse est d'accéder à une compréhension des besoins et des exigences du client.
 - ▮ Il s'agit de livrer des spécifications pour permettre de choisir la conception de la solution.
 - ▮ Un modèle d'analyse livre une spécification complète des besoins issus des cas d'utilisation et les structure sous une forme qui facilite la compréhension (scénarios), la préparation (définition de l'architecture), la modification et la maintenance du futur système.
- 

Conception

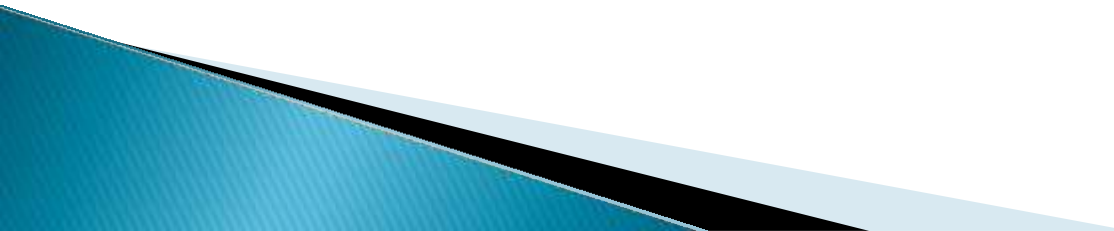
La conception permet d'acquérir une compréhension approfondie des contraintes liées au langage de programmation, à l'utilisation des composants et au système d'exploitation.

- Elle détermine les principales interfaces et les transcrit à l'aide d'une notation commune.
- Elle constitue un point de départ à l'implémentation:
 - ✓ Elle décompose le travail d'implémentation en sous-système
 - ✓ Elle crée une abstraction transparente de l'implémentation.

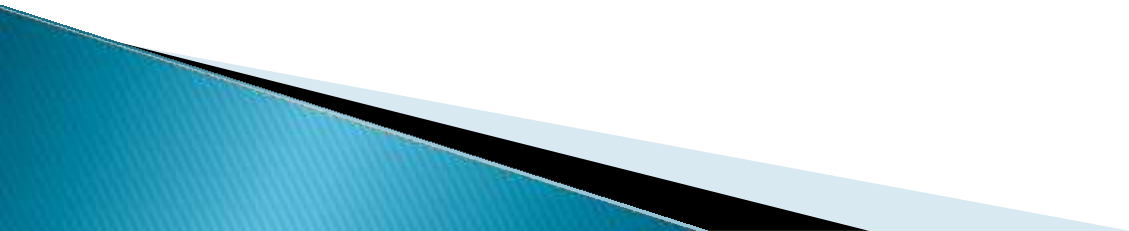
Implémentation

- L'implémentation est le résultat de la conception pour implémenter le système sous formes de composants, c'est-à-dire, de code source, de scripts, de binaires, d'exécutables et d'autres éléments du même type.
- Les objectifs principaux de l'implémentation sont de planifier les intégrations des composants pour chaque itération, et de produire les classes et les sous-systèmes sous formes de codes sources.

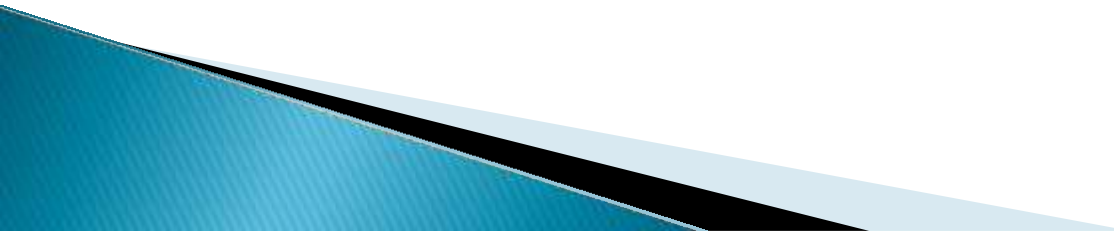
Test

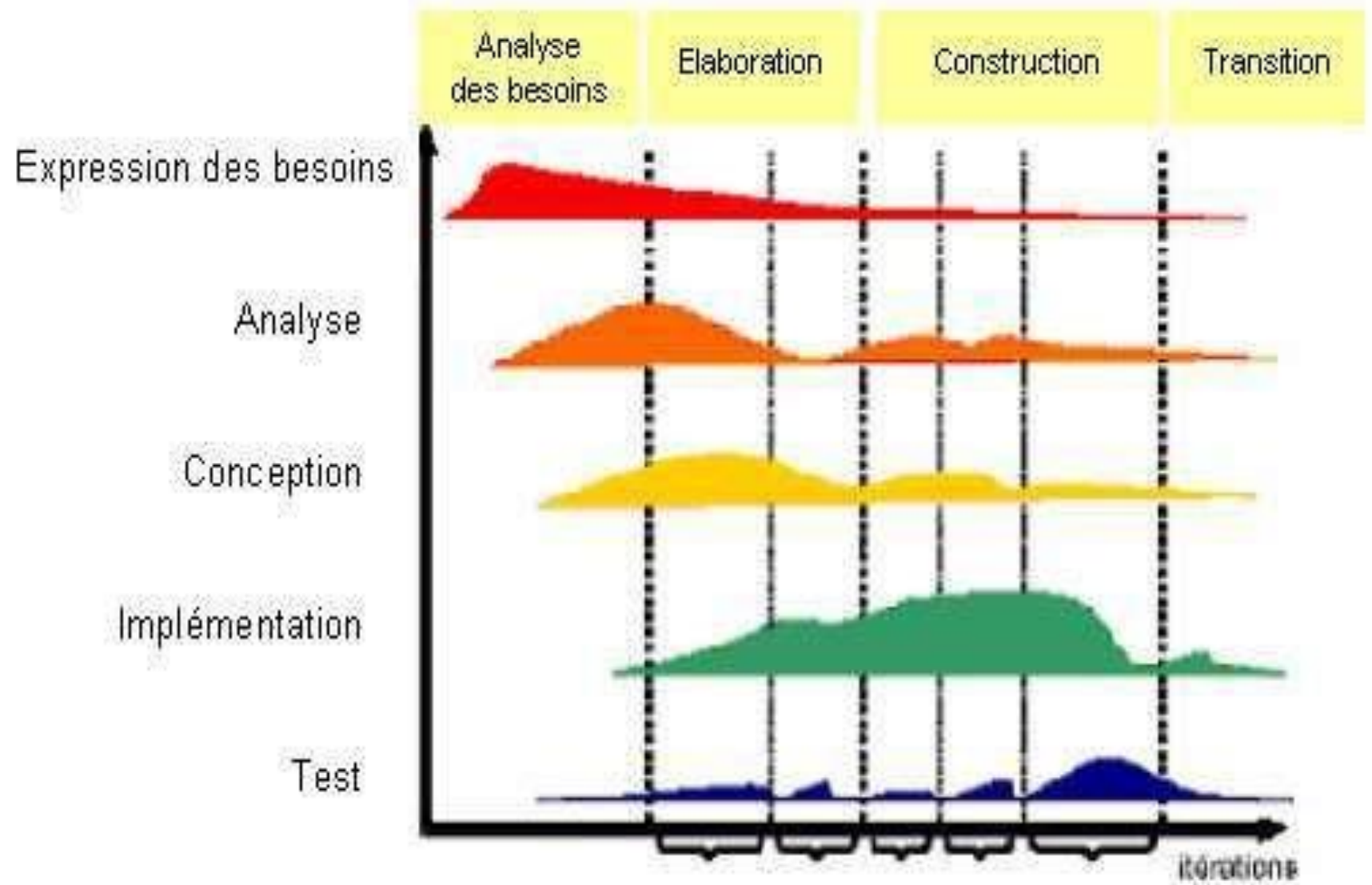
- Les tests permettent de vérifier des résultats de l'implémentation en testant la construction.
 - Pour mener à bien ces tests, il faut les planifier pour chaque itération, les implémenter en créant des cas de tests, effectuer ces tests et prendre en compte le résultat de chacun.
- 

Les phases du processus UP

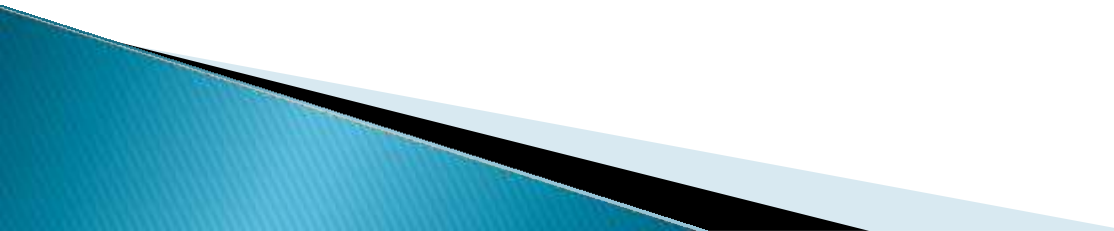


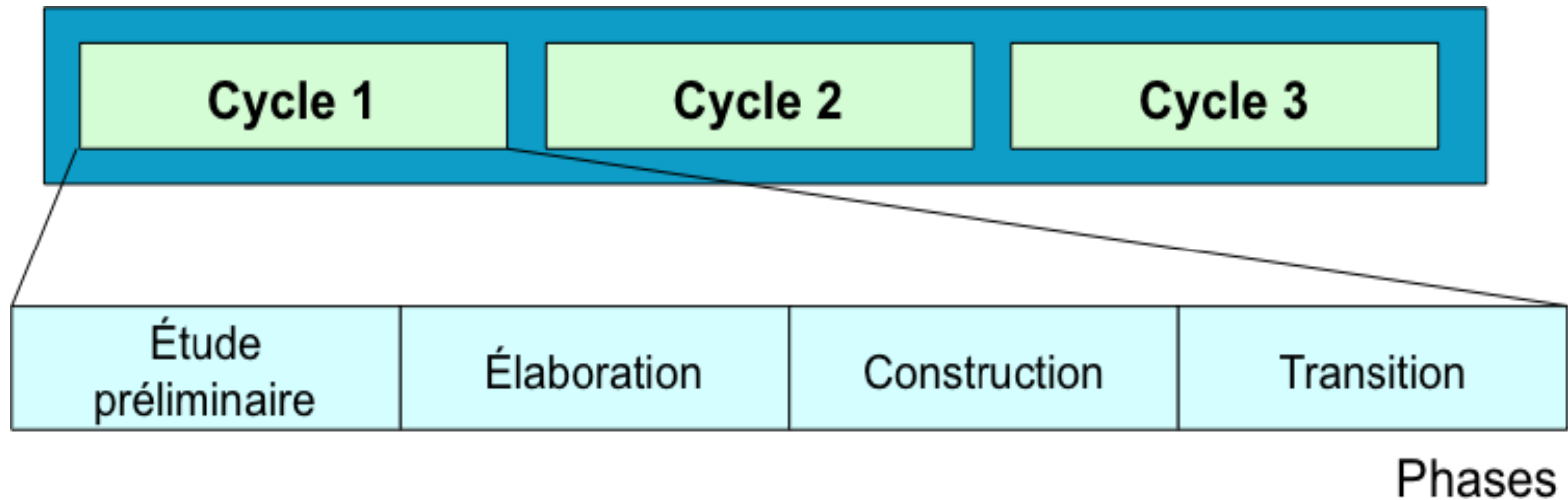
Les phases du processus UP

- Le cycle de vie de la méthode UP se décompose en 4 phases principales au cours desquelles les activités d'ingénierie sont mises en œuvre.
 - Un certain nombre d'itérations est effectué à l'intérieur de chaque phase.
 - Chaque itération met en œuvre certaines activités.
- 



1) Phase de pré-étude (faisabilité) ou inception :

- définition du cadre et de la portée du projet
 - L'objectif de cette phase est de comprendre le problème et de décider du go ou no go du projet.
 - Elle dure environ 1 mois pour un projet d'un an
- 



Considérer un produit logiciel quelconque par rapport à ses versions

- ▮ Un cycle produit une version

Gérer chaque cycle de développement comme un projet ayant quatre phases.

- ▮ vue gestionnaire (manager)

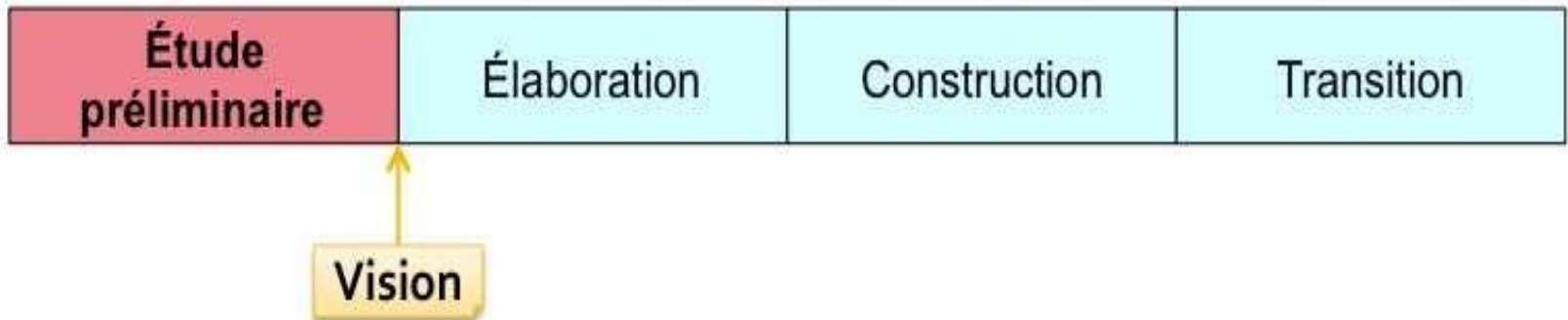
- ▮ Chaque phase se termine par un point de contrôle (ou jalon) permettant aux chefs de projet de prendre des décisions.

REMARQUE

Un jalon, dans le cadre de la gestion de projet, est la fin d'une étape, la fin d'un travail. La plupart du temps, le jalon est aussi un événement important, comme la signature d'un contrat, le lancement d'un produit

Phase 1 : Etude préliminaire

Analyse des besoins



- Déterminer
 - ce que fait le système ?
 - à quoi pourrait ressembler l'architecture ?
 - quels sont les risques ?
 - quel est le coût estimé du projet ? Comment le planifier ?
- Jalon
 - « vision du projet » = document
 - Accepter le projet ?
- Coût faible

Phase 1 : Etude préliminaire

Analyse des besoins

- L'analyse des besoins donne une vue du projet sous forme de produit fini.
- Cette phase porte essentiellement sur :
 - les besoins principaux (du point de vue de l'utilisateur),
 - l'architecture générale du système,
 - les risques majeurs,
 - les délais et les coûts.

- Elle répond aux questions suivantes :
 - que va faire le système ? par rapport aux utilisateurs principaux, quels services va-t-il rendre?
 - quelle va être l'architecture générale (cible) de ce système.
 - quels vont être : les délais, les coûts, les ressources, les moyens à déployer?

2) Phase d'élaboration :

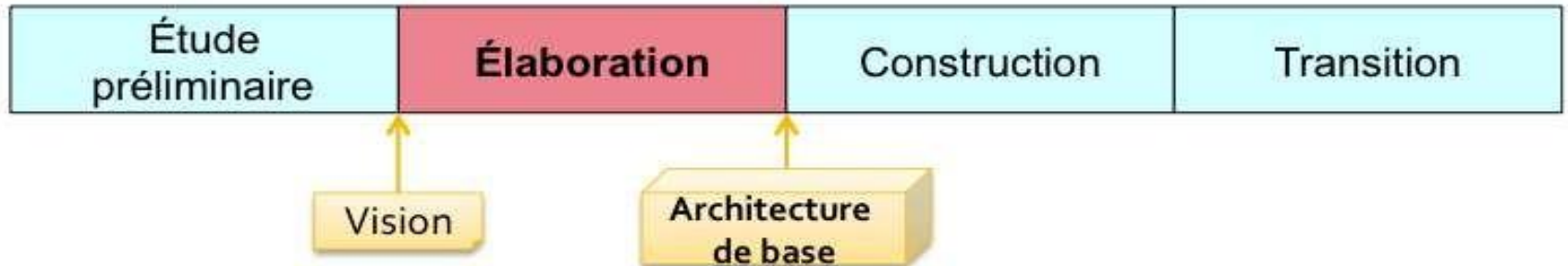
Objectif :

- analyser le système
- analyser le domaine du problème.
- comprendre la solution développer le plan du projet.

Durée : 2 à 4 mois pour un projet d'un an



Phase2 : élaboration



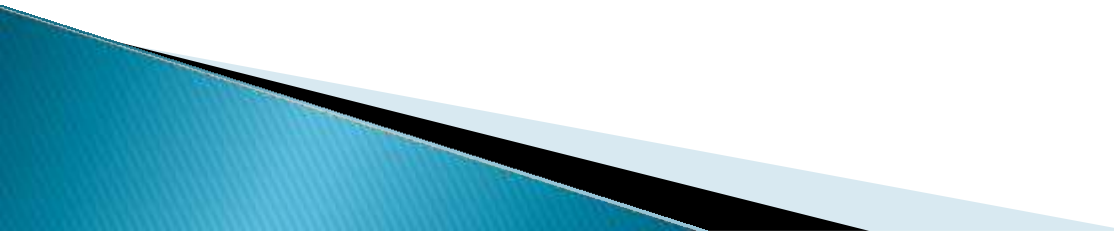
- Spécifier la plupart des cas d'utilisation
- Concevoir l'architecture de base (squelette du système)
- Mettre en œuvre cette architecture
 - CU critiques, <10 % des besoins
- Faire un planification complète
- Jalon
 - « architecture du cycle de vie » = premier prototype qui démontre la validité des choix architecturaux
 - Peut-on passer à la construction ?
 - besoins, architecture, planning stables ? Risques contrôlés ?
- Coût faible

Phase2 : élaboration

L'élaboration permet de préciser la plupart des cas d'utilisation, de concevoir l'architecture du système et surtout de déterminer l'architecture de référence.

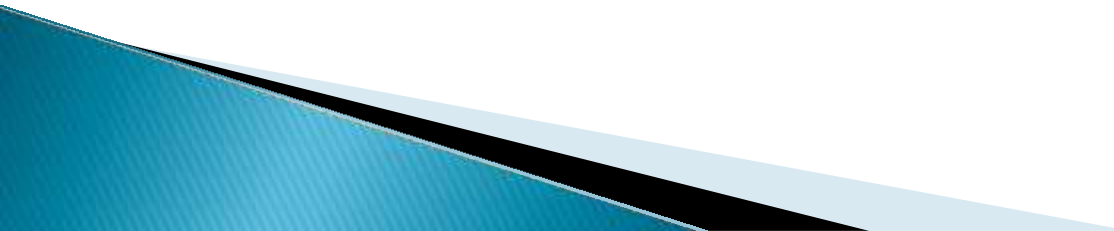
Au terme de cette phase, les chefs de projet doivent être en mesure de prévoir les activités et d'estimer les ressources nécessaires à l'achèvement du projet.

Les tâches à effectuer dans la phase élaboration sont les suivantes :

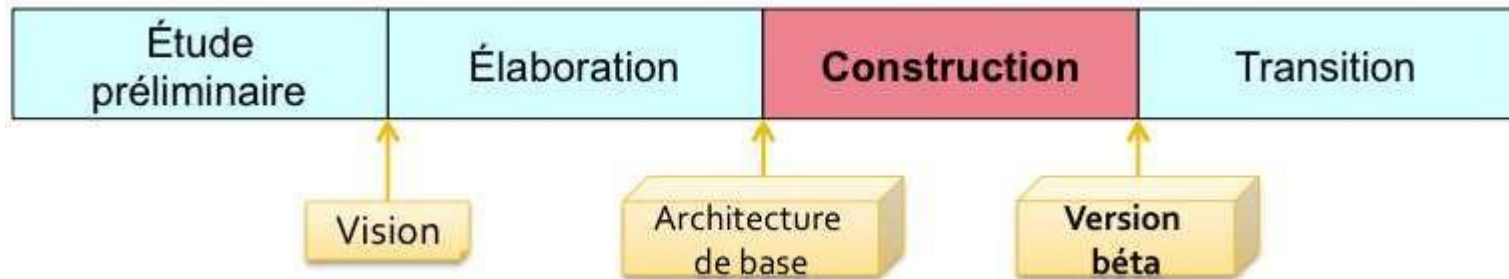
- créer une architecture de référence
 - identifier les risques, ceux qui sont de nature à bouleverser le plan, le coût et le calendrier
 - définir les niveaux de qualité à atteindre
 - formuler les cas d'utilisation pour couvrir les besoins fonctionnels et planifier la phase de construction
 - élaborer une offre abordant les questions de calendrier, de personnel et de budget
- 

3) Phase de construction :

Objectif :

- ✓ réaliser la solution (version bêta)
 - ✓ développement du système
 - ✓ Durée : 6 à 9 mois pour un projet d'un an
- 

Phase 3 : Construction



- Développer par incréments
 - architecture stable malgré des changements mineurs
 - le produit contient tout ce qui avait été planifié
 - il reste quelques erreurs
- Jalon
 - « capacité opérationnelle initiale » = version bêta
 - Le produit est suffisamment correct pour être installé chez un client ?
- Phase la plus couteuse
 - > 50% du cycle
 - Englobe conception, code, tests, intégration...

Phase 3 : Construction

La construction est le moment où l'on construit le produit.

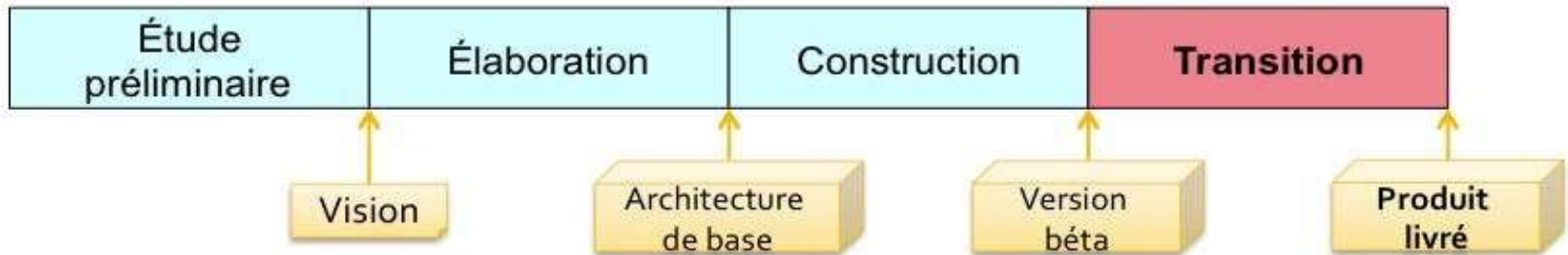
L'architecture de référence se métamorphose en produit complet.

Le produit contient tous les cas d'utilisation que les chefs de projet, en accord avec les utilisateurs ont décidé de mettre au point pour cette version.

4) Phase de transition :

- ▮ Objectif : mise en service de la solution
- ▮ livraison du système aux utilisateurs finaux
- ▮ Durée : 1 mois pour un projet d'un an

Phase 4: Transition



- Transition
 - Livrer / déployer le produit
 - Corriger le reliquat d'erreurs
 - Améliorer le produit
 - Former les utilisateurs
 - Mettre en place l'assistance en ligne
- Jalon
 - « livraison du produit » = produit déployé chez le client
 - tests suffisants ? Produit satisfaisant ? Manuels prêts ?

Phase 4: Transition

Le produit est en version bêta.

Un groupe d'utilisateurs essaye le produit et détecte les anomalies et défauts.

Cette phase suppose des activités comme la formation des utilisateurs clients, la mise en œuvre d'un service d'assistance et la correction des anomalies constatées.

PU : activités et phases

- Activités : description technique de la méthode
 - que faut-il faire ?
 - sous quelle forme (modèles, documents textuels...) ?
 - comment obtenir les produits ?
- Phases : description du déroulement du projet
 - planifier les itérations / phases suivantes
 - pour chaque phase :
 - quels sont les buts à atteindre ?
 - quels sont les livrables ?
 - quels aspects décrire, avec quel niveau d'abstraction ?

Les activités selon les phases

