

**Cycle de vie de logiciel &
Méthodes de conception**



Rappel sur les cycles de vie et les méthodes de conception



Partie 1



**Mr Sellami K
Informatiqueii9@gmail.com**

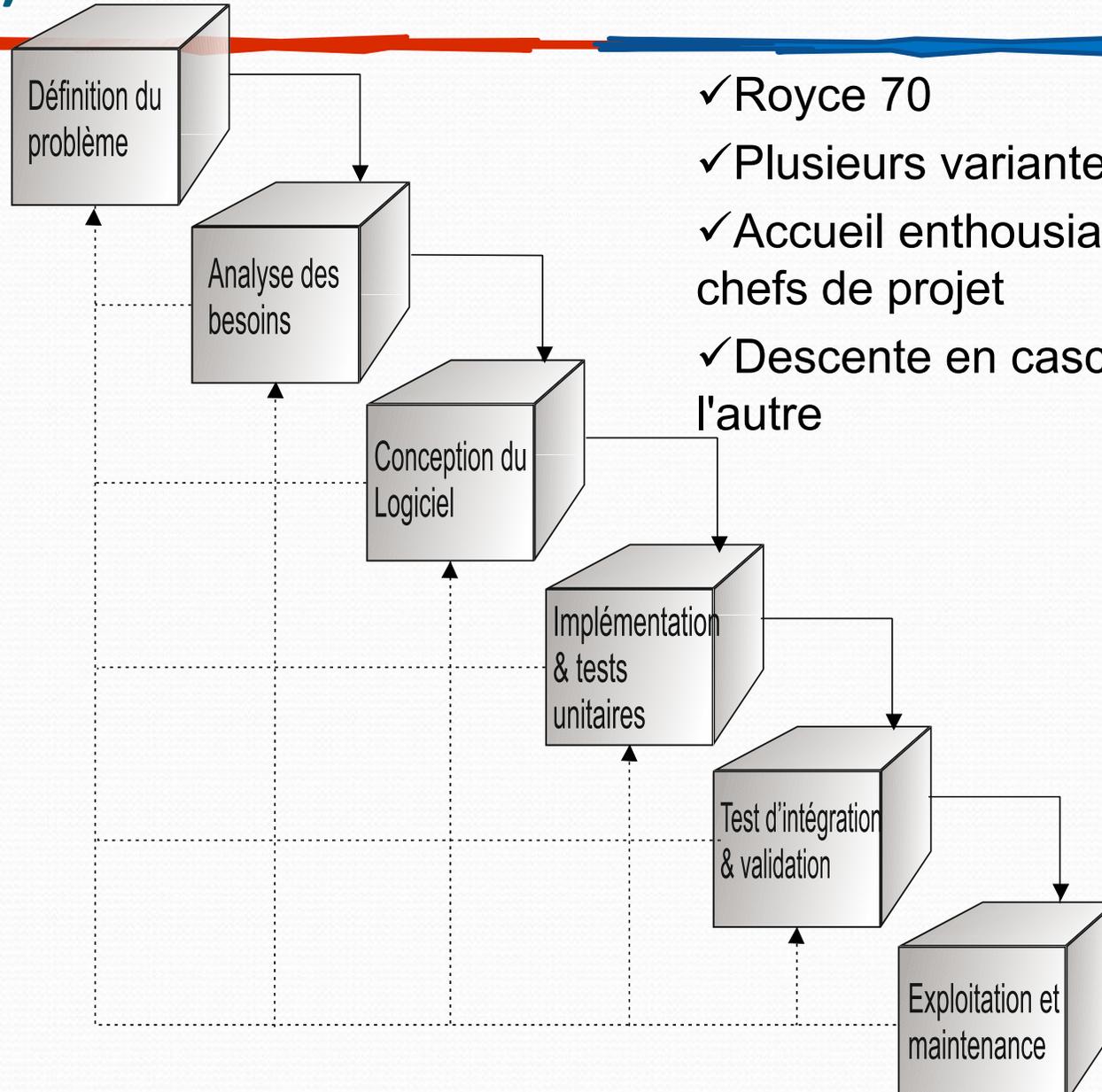
Processus de développement de logiciel

Suivre le cycle de vie étape par étape offre plusieurs avantages :

- Limiter les risques en identifiant les problèmes dès le début et à chaque étape du processus.
- Réduire le temps nécessaire à la mise sur le marché du projet.
- Contrôler et suivre l'évolution du développement, peu importe l'ampleur du projet.
- Établir une communication entre les différents acteurs du projet qui doivent coopérer.
- Définir une structure claire, ce qui assure une approche méthodique et organisée du développement.

Les modèles de développement:

1) modèle de la cascade



✓ Royce 70

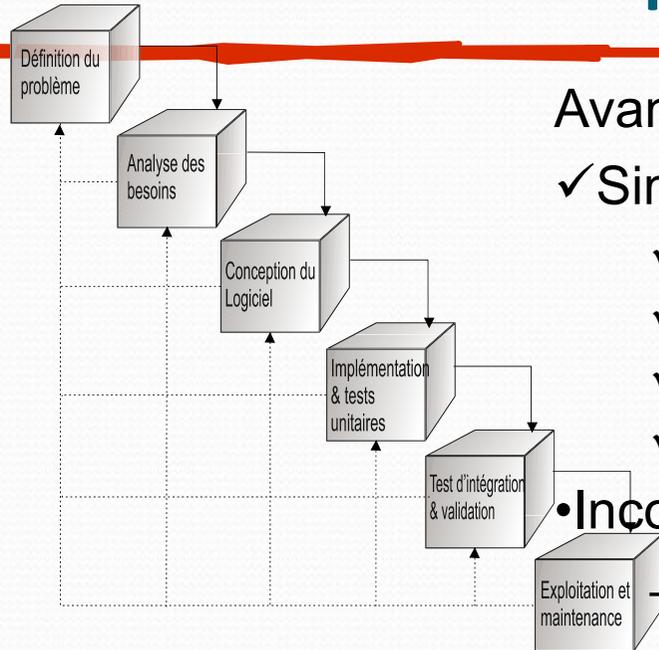
✓ Plusieurs variantes

✓ Accueil enthousiaste de la part des chefs de projet

✓ Descente en cascade d'une étape à l'autre



Les modèles de développement: modèle de la cascade



Avantages :

✓ Simplicité

- ✓ Def & planif plus précises
- ✓ Danger détectés à l'avance
- ✓ Responsabilité claires
- ✓ Respect Échéances

• Inconvénients

- Flexibilité ✗
- Retour en arrière ✗
- Risque de déception du client
- Cout des Changements ↑



Les modèles de développement: modèle de la cascade

ÉTAPES TYPES DU CYCLE DE VIE TRADITIONNEL

•a. *Définition du problème*

•Actions :

- On identifie le problème (quel est-il ? pour qui ? pourquoi veut-on le résoudre ?)
- Et on décide ou non de la mise en place d'un projet logiciel.

•Moyens:

- Traitements de textes, tableurs

•Délivrables en sortie:

- Ouverture d'un dossier projet
- Identification du maître de l'ouvrage et du maître de l'œuvre

•Intervenants:

Les intervenants sont fort peu nombreux, on distingue essentiellement:

- des consultants qui peuvent être
 - internes, c'est à dire appartenant à l'organisation considérée.
 - externes, c'est à dire n'appartenant pas à l'organisation mais plutôt à un cabinet de conseil extérieur. Et cette solution est souvent meilleure surtout si une prise de recul est nécessaire.
- Aucune structure de type comité directeur, groupe de projet, groupe d'utilisateurs n'est mise en place car il n'y a pas encore de projet logiciel.

•Remarque:

- Le projet décidé peut faire partie d'un ensemble de projets planifiés. Il est alors inscrit dans le plan informatique de l'E (entreprise ou organisation). Il est bon de rapprocher les objectifs du projet décidé des objectifs stratégiques de l'E et de voir s'il entre dans le cadre général du plan de développement de l'E.



Les modèles de développement: modèle de la cascade

ÉTAPES TYPES DU CYCLE DE VIE TRADITIONNEL

•b. Analyse des besoins

•Actions :

- Le concepteur recense et étudie les besoins du client (maître de l'ouvrage) c'est à dire :
- les fonctions que le logiciel doit effectuer
- les conditions d'exploitation
- le contrat de service
- la qualité requise (performances, sûreté, fiabilité,...)

•Moyens:

- Les différents moyens d'enquête (interviews, sondages, entretiens, consultation de documents, brainstorming, etc.) et ceci en fonction du projet et de l'interlocuteur.
- Langages de description formelle (PDL: Program Design Language), techniques graphiques de modélisation (SADT, E/A,...)
- Outils de maquettage, et autres AGL

•Traitements de textes

•Délivrables en sortie:

•• Cahier des charges: document qui est un engagement et qui fait partie du contrat de réalisation du projet.

•Intervenants:

Les intervenants dans cette phase peuvent être nombreux, et on peut principalement distinguer:

- Le chef de projet qui doit animer, orienter et veiller au respect des contraintes: Délais imposés, Qualité requise, Coûts à minimiser.
- Les consultants qui apportent de l'aide et de l'expérience
- Les experts qui permettent de vérifier ou valider la pertinence d'une solution.
- Les utilisateurs qui expriment leurs besoins, leurs attentes et souhaits.
- Le comité directeur qui décide du choix de l'orientation à prendre,



Les modèles de développement: modèle de la cascade

ÉTAPES TYPES DU CYCLE DE VIE TRADITIONNEL

•b. Analyse des besoins

•Actions :

- Le concepteur recense et étudie les besoins du client (maître de l'ouvrage) c'est à dire :
- les fonctions que le logiciel doit effectuer
- les conditions d'exploitation

Remarque:

L'analyse des besoins comprend généralement une analyse de l'existant.

•Mo La validation de cette étape est **EXTREMEMENT IMPORTANTE**. En effet, c'est lors de cette validation que la décision de poursuivre ou d'arrêter le projet sera prise.

De plus, c'est à ce moment que l'on décidera des types de solutions envisageables (achat de progiciel, développement effectué à l'intérieur de l'E ou sous-traités à des SSII, réorganisations, versions successives, etc.).

•Dé On vérifie la faisabilité (économique, technologique, organisationnelle, sociale) et on choisit la solution la plus appropriée.

•• C

•Intervenants:

Les intervenants dans cette phase peuvent être nombreux, et on peut principalement distinguer:

- Le chef de projet qui doit animer, orienter et veiller au respect des contraintes: Délais imposés, Qualité requise, Coûts à minimiser.
- Les consultants qui apportent de l'aide et de l'expérience
- Les experts qui permettent de vérifier ou valider la pertinence d'une solution.
- Les utilisateurs qui expriment leurs besoins, leurs attentes et souhaits.
- Le comité directeur qui décide du choix de l'orientation à prendre,

Les modèles de développement: modèle de la cascade

ÉTAPES TYPES DU CYCLE DE VIE TRADITIONNEL

•c. *Conception du logiciel*

•Actions :

•Le concepteur étudie les moyens (informatiques, organisationnels, etc.) nécessaires aux fonctions définies dans le cahier des charges :

- choix du matériel
- la structure des données
- l'architecture des traitements
- éventuellement les nouvelles organisations du travail (définition des postes, formations à mettre en place, etc.) et les nouvelles procédures de travail.

•Moyens:

- Outils de conception
- Simulation
- Outils de prototypage
- Méthodes et/ou modèles (Organigrammes, algorithmes, E/A, Pétri, SADT,...)

•Délivrables en sortie:

•Le document des spécifications techniques (générales et détaillées) qui précise comment sera réalisé physiquement le futur logiciel.

•Intervenants:

Principalement, on peut distinguer:

- Le chef de projet qui gère, négocie, contrôle et coordonne.
- Les analystes qui détaillent les besoins et les solutions retenues
- Les analystes programmeurs qui décrivent d'une façon complète et détaillée les modules.
- Les techniciens qui sont garant de l'architecture technique à mettre en place: matériels, logiciels, et AGL.
- Les experts qui apportent leur compétence.
- Le responsable qualité qui veille à la qualité des travaux réalisés.
- Les utilisateurs pour consultation éventuelle.
- Ainsi que le comité directeur.

Les modèles de développement: modèle de la cascade

ÉTAPES TYPES DU CYCLE DE VIE TRADITIONNEL

d. Implémentation et test unitaire

• **Actions :**

- Le concepteur élabore les différents programmes et crée les fichiers de données.
- Il procède aux tests unitaires (de premier niveau)

• **Moyens:**

- Environnements de programmation qui supportent un certain nombre de langages
- AGL
- Outils de vérification et de tests

• **Délivrables en sortie:**

- Les programmes, structure des fichiers de données.
- La documentation des programmes et la documentation utilisateur
- les rapports des tests unitaires
- le rapport de programmation (réalisation)

• **Intervenants:**

En plus des intervenants de l'étape précédente, on a:

- Les programmeurs qui procèdent au codage des différents modules.
- Configurateur du système d'exploitation
- Outils de métrologie
- Installateur de corrections et de versions

Les modèles de développement: modèle de la cascade

ÉTAPES TYPES DU CYCLE DE VIE TRADITIONNEL

•e. *Exploitation et maintenance*

•Actions :

- Réaliser un essai complet sur un morceau majeur de logiciel

•Moyens:

- Environnements de programmation qui supportent un certain nombre de langages
- AGL
- Outils de maintenance

•Délivrables en sortie:

- Recette définitive
- Contrat de maintenance
- Le registre des incidents
- Le rapport de suivi de maintenance

•Intervenants:

- Responsable de projets si la maintenance demande des efforts importants.
- Les analystes
- Les programmeurs
- Les utilisateurs
- Le comité directeur, si la maintenance est importante et de grande ampleur.

Les modèles de développement: modèle de la cascade

ÉTAPES TYPES DU CYCLE DE VIE TRADITIONNEL

•e. *Exploitation et maintenance*

•Actions :

- Réaliser un essai complet sur un morceau majeur de logiciel

•Moyens:

- Environnements de programmation qui supportent un certain nombre de langages
- AGL
- Outils de maintenance

•Délivrables en sortie:

- Recette définitive
- Contrat de maintenance
- Le registre des incidents
- Le rapport de suivi de maintenance

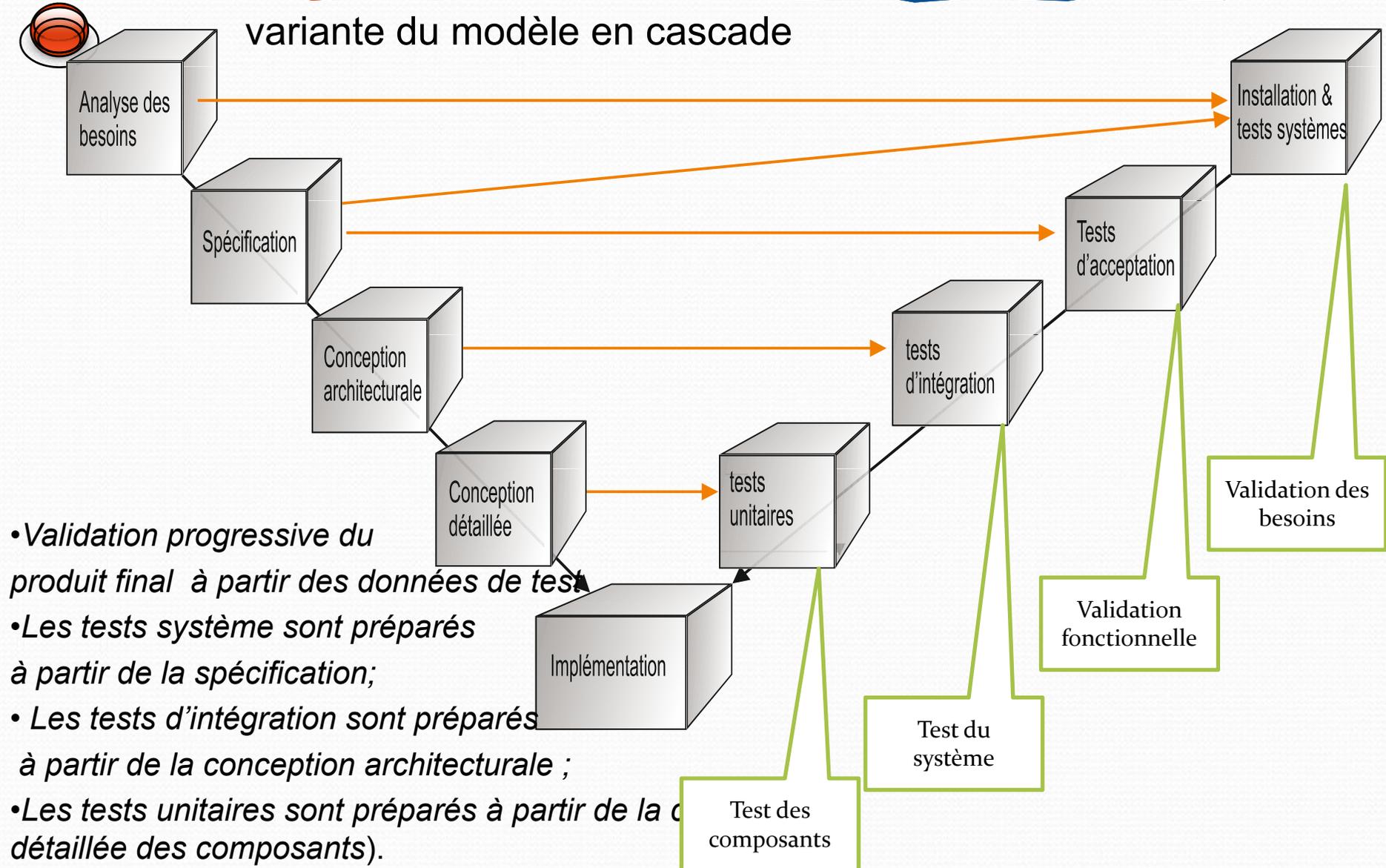
•Intervenants:

- Responsable de projets si la maintenance demande des efforts importants.
- Les analystes
- Les programmeurs
- Les utilisateurs
- Le comité directeur, si la maintenance est importante et de grande ampleur.

Les modèles de développement:

2) modèle en V

variante du modèle en cascade



Les modèles de développement:

2) modèle en V

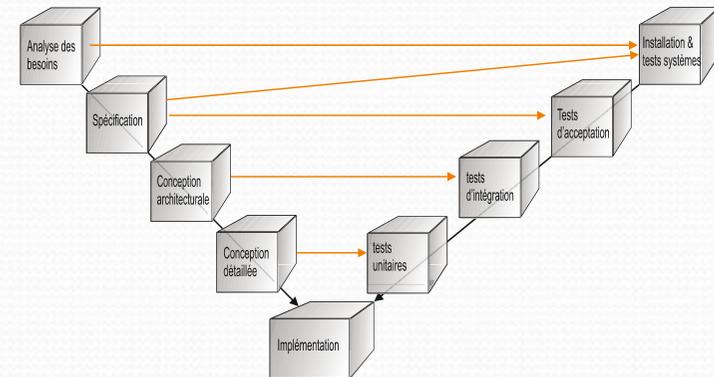


Avantages :

- Facilité d'utilisation
- Met l'accent sur les tests ;

•Inconvénients

- Pas de prise en compte des activités parallèles
- Pas de gestion des changements dans la spécification
- Pas d'activité d'analyse des risques



Les modèles de développement:

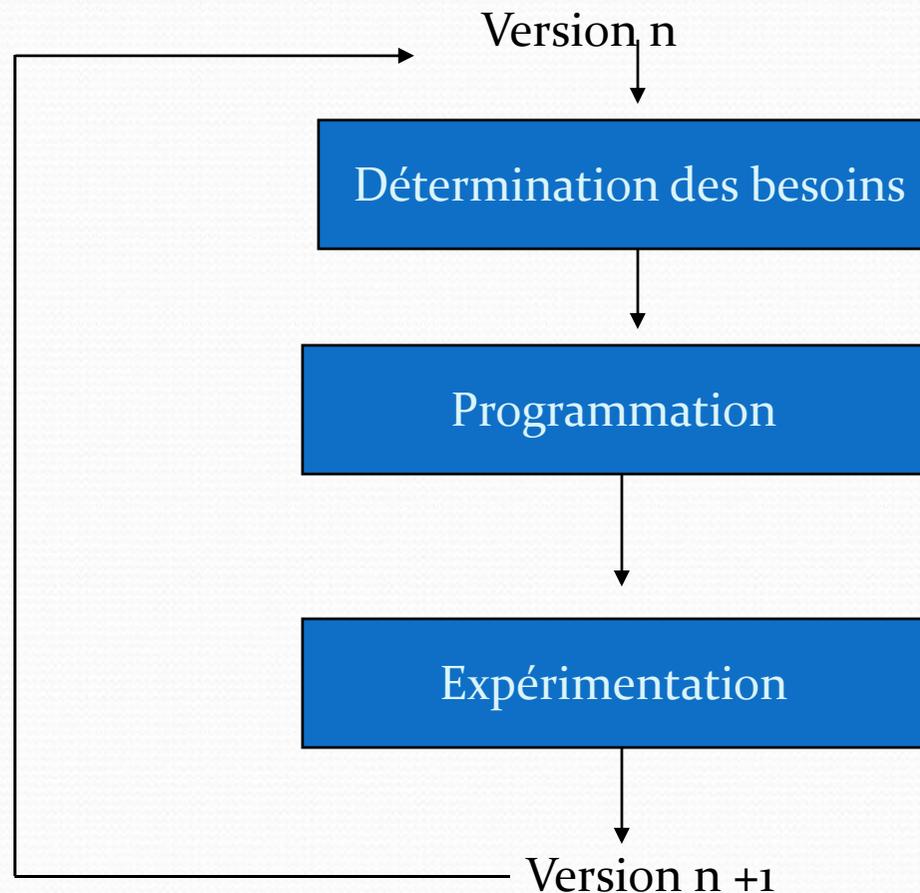


3) Développement évolutif



BUT: Construire progressivement le système de façon participative

Critère d'arrêt: le client juge le système satisfaisant.



Les modèles de développement:

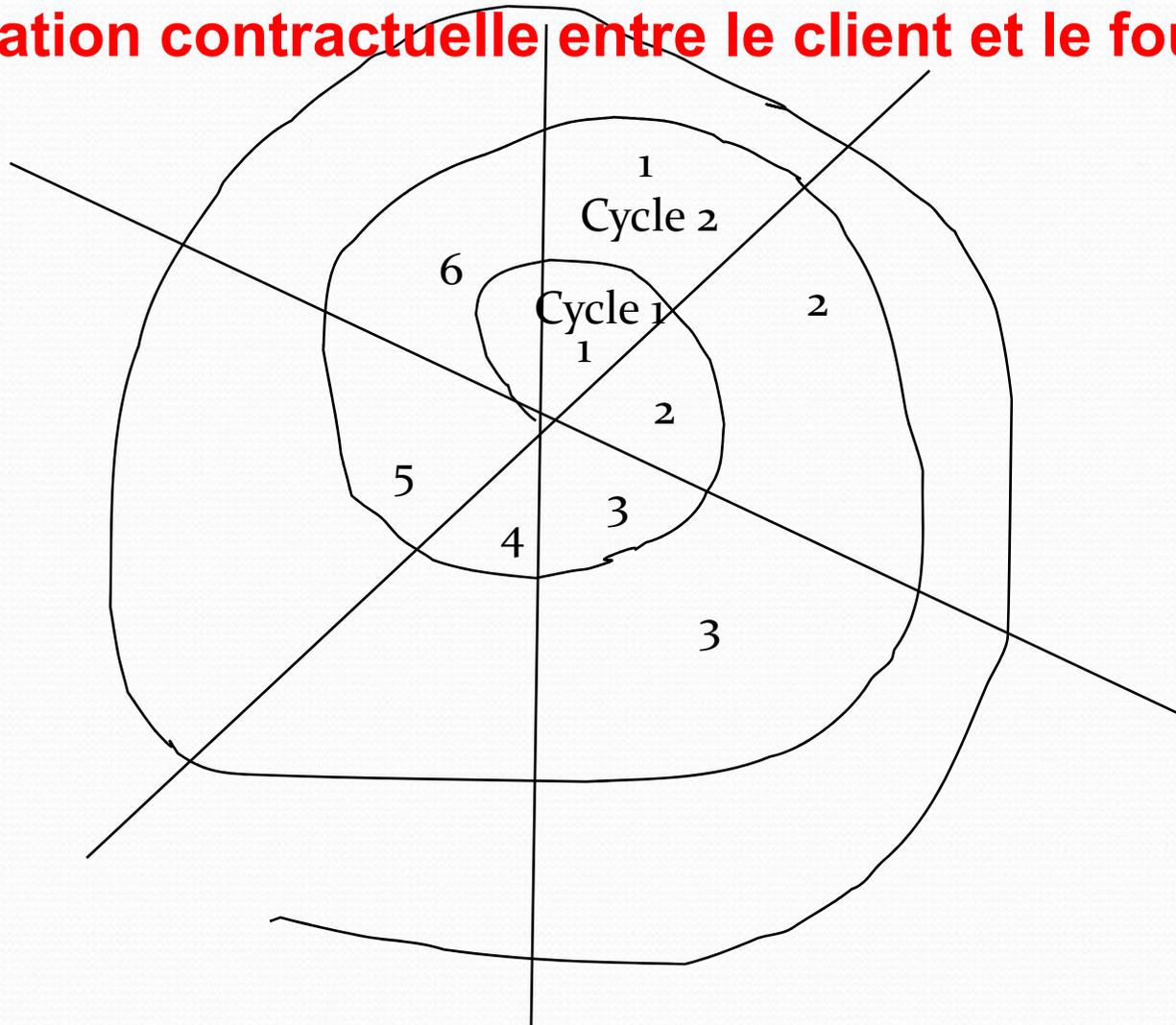


3) Modèle de la spirale



Même principe que le modèle évolutif

Relation contractuelle entre le client et le fournisseur.



Les modèles de développement:



3) Modèle de la spirale



• Chaque cycle de la spirale est composé de

- 1. Analyse du risque
- 2. Développement d'un prototype
- 3. Simulation et essais du prototype
- 4. Détermination des besoins, à partir des résultats des essais
- 5. Validation des besoins par un comité de pilotage
- 6. Planification du cycle suivant

Les modèles de développement:



3) Modèle de la spirale

- Le dernier cycle comprend :
 - en phase 2 développement de la version finale
 - en phase 3 tests et installation
 - et s'arrête là.

Les modèles de développement:

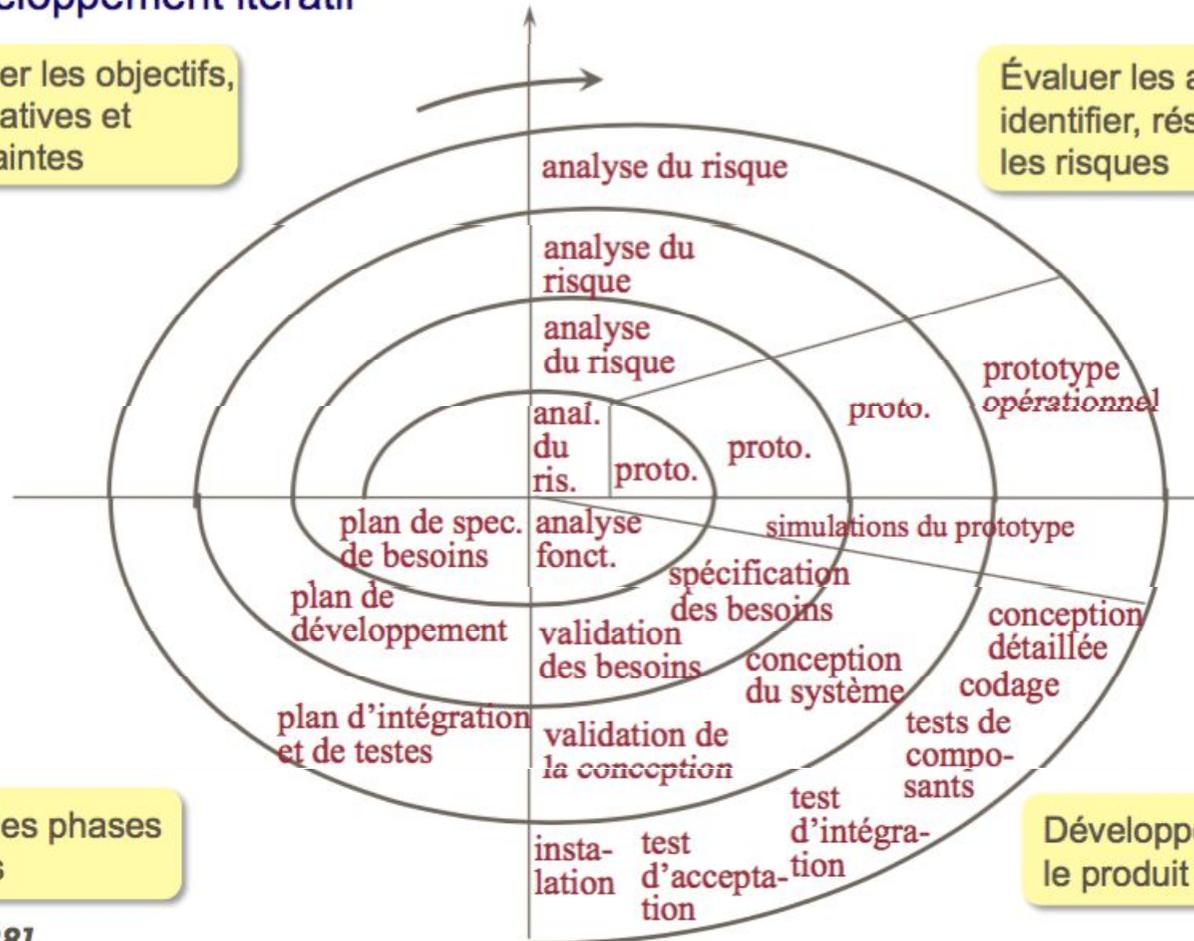


3) Modèle de la spirale

Développement itératif

Déterminer les objectifs, les alternatives et les contraintes

Évaluer les alternatives, identifier, résoudre les risques



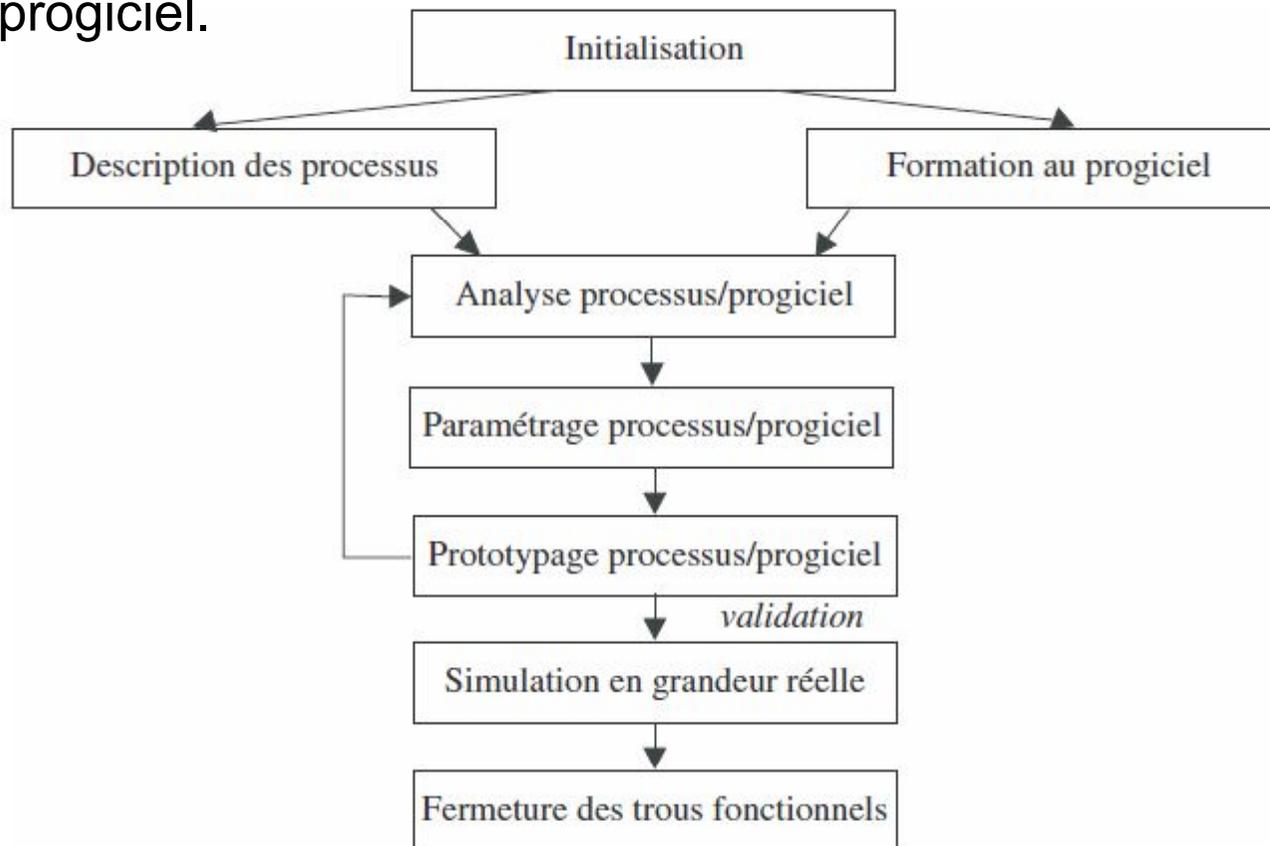
[Boehm 88]

LES MODÈLES DE CYCLE DE VIE SPÉCIFIQUES:



Le cycle ERP

- Tirer le meilleur parti du progiciel
- Construction d'un système améliorant la performance de l'entreprise
- A la fin; prendre en compte ce qui est resté en dehors du champ couvert par le progiciel.



LES MODÈLES DE CYCLE DE VIE SPÉCIFIQUES:



Le modèle RUP

- Le modèle RUP (*Rational Unified Process*) est représentatif d'une approche combinant plusieurs modèles. Sa structure fait l'objet d'un assez large accord, notamment parmi les praticiens.

La démarche du modèle de processus unifié est ..

- Itérative et incrémentale

Développement graduel des incréments à partir des premières ébauches, évolution au fur et à mesure des modèles

- Dirigée par la gestion des risques fonctionnels et techniques

Les décisions prises lors des premières étapes réduisent en priorité les risques fonctionnels

- Centrée sur l'architecture de la solution

Organisation du logiciel progressive au fur et à mesure de la réalisation des classes et objets

- Guidée par les exigences des utilisateurs

Réponse le plus possible aux exigences exprimés initialement par les utilisateurs, et validé régulièrement.

LES MODÈLES DE CYCLE DE VIE SPÉCIFIQUES:



Le modèle RUP

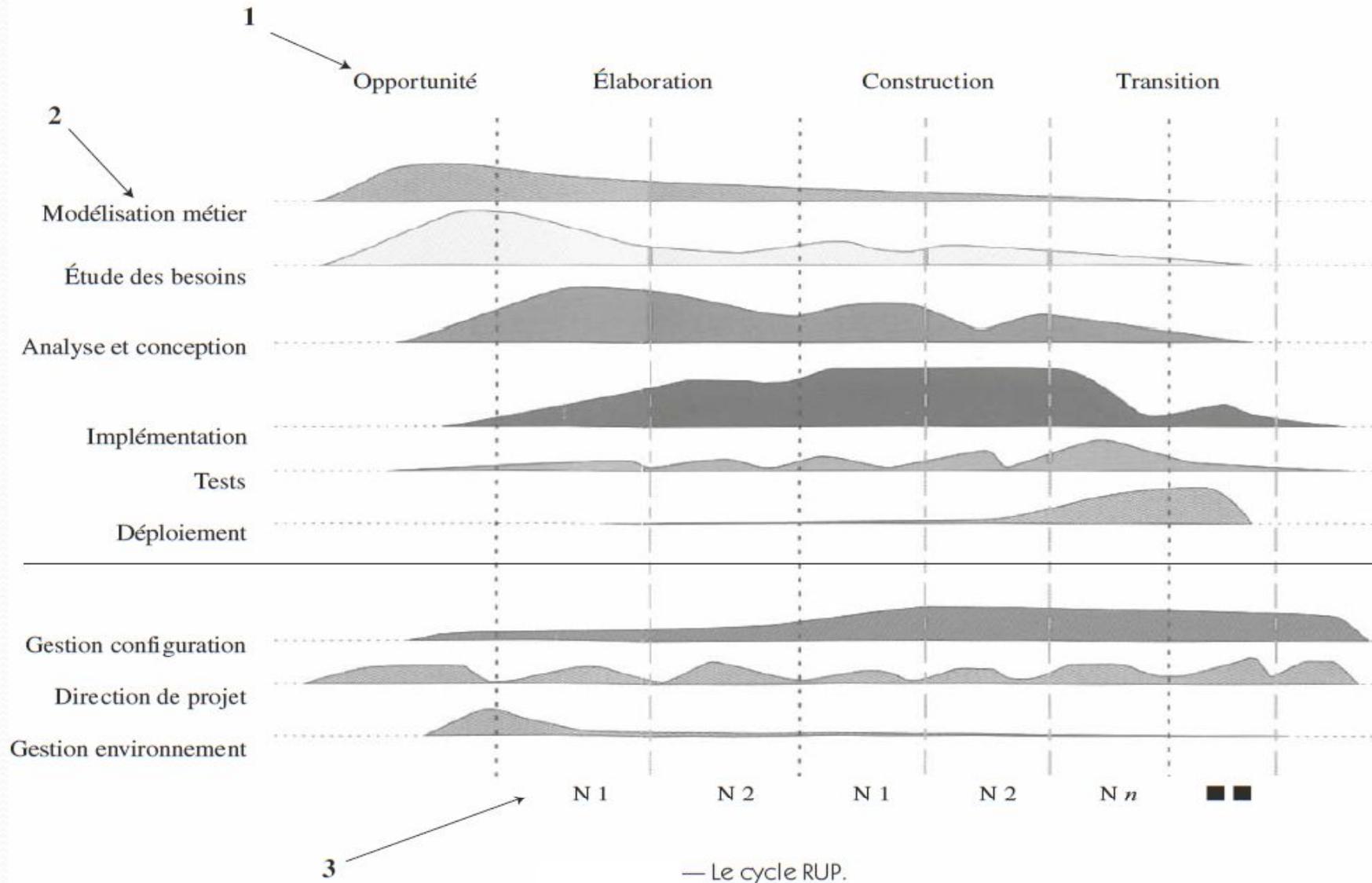
Processus itératif et incrémental divisé en 4 phases :

1. **Initialisation** : définition de l'objectif du projet et préparation d'un contrat de réalisation
2. **Élaboration** : planification du projet, spécification de la solution proposée, proposition d'une architecture de base
3. **Construction** : développement du produit
4. **Transition** : transition du produit chez les utilisateurs.

Ces 4 phases sont elles-mêmes divisées en N itération, à la fin desquelles Il y a une version du produit livrable

LES MODÈLES DE CYCLE DE VIE SPÉCIFIQUES:

Le modèle RUP



LES MODÈLES DE CYCLE DE VIE SPÉCIFIQUES:

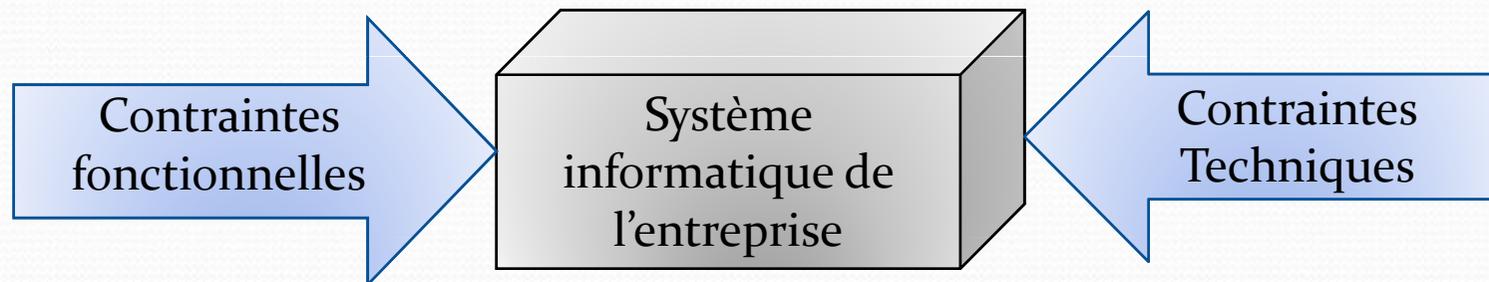
Le modèle 2TUP

Le processus de développement: **2 Tracks Unified Process**, prononcez « **toutiyoupi** ».

Il s'agit des chemins:

- Fonctionnels et
- Architecture technique ,

qui correspondent aux deux axes de changement imposés au système informatique.



LES MODÈLES DE CYCLE DE VIE SPÉCIFIQUES:



Le modèle 2TUP

Exemple : Pour illustrer cet axiome,

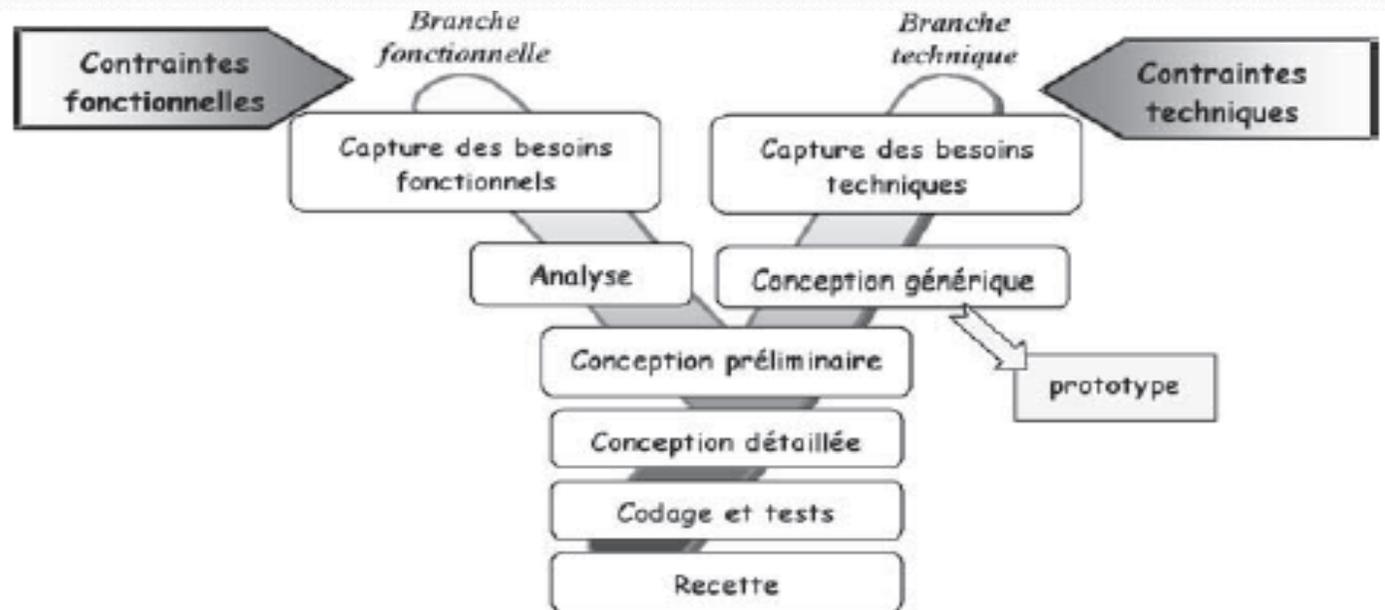
1. une agence de tourisme passe des accords avec une compagnie aérienne de sorte que le calcul des commissions change. En l'occurrence, les résultats issus de la branche fonctionnelle qui évoluent pour prendre en compte la nouvelle spécification ;
2. cette même entreprise décide d'ouvrir la prise de commande sur le Web. Si rien ne change fonctionnellement, en revanche, l'architecture technique du système évolue ;
3. cette entreprise décide finalement de partager son catalogue de prestations avec les vols de la compagnie aérienne. D'une part, la fusion des deux sources d'informations imposera une évolution de la branche fonctionnelle, d'autre part, les moyens techniques de synchronisation des deux systèmes conduiront à étoffer l'architecture technique du système. L'étude de ces évolutions pourra être menée indépendamment, suivant les deux branches du 2TUP.

LES MODÈLES DE CYCLE DE VIE SPÉCIFIQUES:



Le modèle 2TUP

À l'issue des évolutions du modèle fonctionnel et de l'architecture technique, la réalisation du système consiste à fusionner les résultats des deux branches.



Le processus de développement en Y

LES MODÈLES DE CYCLE DE VIE SPÉCIFIQUES:



Le modèle 2TUP

La branche gauche (fonctionnelle) comporte :

- la capture des besoins fonctionnels, qui produit un modèle des besoins

focalisé sur le métier des utilisateurs.

But: Eviter au plus tôt le risque de produire un système inadapté aux utilisateurs. De son côté,

Démarche : la maîtrise d'œuvre consolide les spécifications et en vérifie la cohérence et l'exhaustivité l'analyse, qui consiste à étudier précisément la spécification fonctionnelle de manière à obtenir une idée de ce que va réaliser le système en termes de métier.

Les résultats de l'analyse ne dépendent d'aucune technologie particulière.

LES MODÈLES DE CYCLE DE VIE SPÉCIFIQUES:



Le modèle 2TUP

La branche droite (architecture technique) comporte :

- la capture des besoins techniques, qui recense toutes les contraintes et les choix dimensionnant la conception du système. Les outils et les matériels sélectionnés ainsi que la prise en compte de contraintes d'intégration avec l'existant conditionnent généralement des prérequis d'architecture technique ;
- la conception générique, qui définit ensuite les composants nécessaires à la construction de l'architecture technique.

Cette conception est la moins dépendante possible des aspects fonctionnels.

BUT: Uniformiser et réutiliser les mêmes mécanismes pour tout un système.

L'architecture technique construit le squelette du système informatique et écarte la plupart des risques de niveau technique. L'importance de sa réussite est telle qu'il est conseillé de réaliser un prototype pour assurer sa validité.

LES MODÈLES DE CYCLE DE VIE SPÉCIFIQUES:



Le modèle 2TUP

La branche du milieu comporte :

- la conception préliminaire, qui représente une étape délicate, car elle intègre le modèle d'analyse dans l'architecture technique de manière à tracer la cartographie des composants du système à développer ;
- la conception détaillée, qui étudie ensuite comment réaliser chaque composant;
- l'étape de codage, qui produit ces composants et teste au fur et à mesure les unités de code réalisées ;
- l'étape de recette, qui consiste enfin à valider les fonctions du système développé.

LES MODÈLES DE CYCLE DE VIE SPÉCIFIQUES:



Le modèle 2TUP

LES BRANCHES DU “Y” PRODUISENT DES MODÈLES RÉUTILISABLES

La branche gauche capitalise la connaissance du métier de l'entreprise.

➡ Investissement pour le moyen et le long terme.

➡ Les fonctions du système d'informations sont en effet indépendantes des technologies utilisées.

En pratique **x** , car dans bien des cas, la connaissance fonctionnelle d'un produit se perd dans les milliers de ligne de code de sa réalisation.

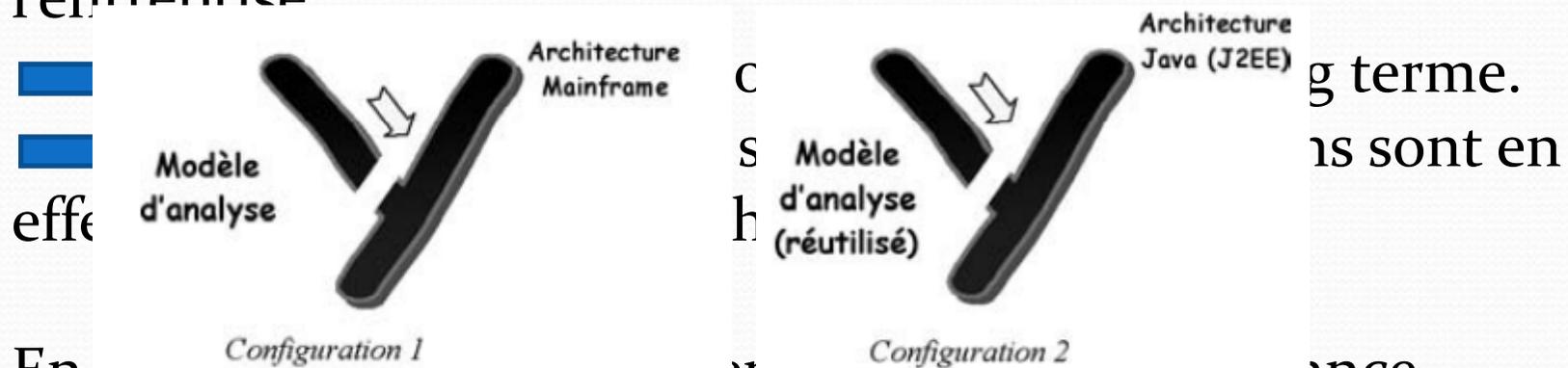
L'entreprise qui maintient le modèle fonctionnel de sa branche gauche est pourtant à même de le réaliser sous différentes technologies. Il suffit de « greffer » une nouvelle architecture technique pour mettre à jour un système existant.

LES MODÈLES DE CYCLE DE VIE SPÉCIFIQUES:

Le modèle 2TUP

LES BRANCHES DU "Y" PRODUISENT DES MODÈLES RÉUTILISABLES

La branche gauche capitalise la connaissance du métier de l'entreprise



En fonctionnelle d'un produit se perd dans les milliers de ligne de code de sa réalisation.

L'entreprise qui maintient le modèle fonctionnel de sa branche gauche est pourtant à même de le réaliser sous différentes technologies. Il suffit de « greffer » une nouvelle architecture technique pour mettre à jour un système existant.

LES MODÈLES DE CYCLE DE VIE SPÉCIFIQUES:



Le modèle 2TUP

LES BRANCHES DU “Y” PRODUISENT DES MODÈLES RÉUTILISABLES

La branche droite capitalise quant à elle un savoir-faire technique. Elle I



Investissement pour le court et le moyen terme.



Les techniques développées pour le système peuvent l'être en effet indépendamment des fonctions à réaliser.

L'architecture technique est d'ailleurs de moins en moins la réoccupation des services informatiques dont l'entreprise n'a pas vocation à produire du code.

L'existence de produits tels que les serveurs d'application ou la standardisation des services Web reflète cette tendance à pouvoir disposer sur le marché d'architectures techniques « prêtes à intégrer ».

Une architecture technique est en effet immédiatement réutilisable pour les différentes composantes fonctionnelles d'un même système d'entreprise.

LES MODÈLES DE CYCLE DE VIE SPÉCIFIQUES:



Le modèle 2TUP

LES BRANCHES DU “Y” PRODUISENT DES MODÈLES RÉUTILISABLES

La branche droite capitalise quant à elle un savoir-faire technique. Elle I



Investissement pour le court et le moyen terme.



Les techniques développées pour le système peuvent l'être en effet indépendamment des fonctions à réaliser.

L'architecture technique est d'ailleurs de moins en moins la réoccupation des services informatiques dont l'entreprise n'a pas vocation à produire du code.

L'existence de produits tels que les serveurs d'application ou la standardisation des services Web reflète cette tendance à pouvoir disposer sur le marché d'architectures techniques « prêtes à intégrer ».

Une architecture technique est en effet immédiatement réutilisable pour les différentes composantes fonctionnelles d'un même système d'entreprise.

LES MODÈLES DE CYCLE DE VIE SPÉCIFIQUES:

Le modèle 2TUP

LES BRANCHES DU "Y" PRODUISENT DES MODÈLES RÉUTILISABLES

La branche droite capitalise quant à elle un savoir-faire technique. Elle I
Investissement pour le court et le moyen terme.



L'existence de produits tels que les serveurs d'application ou la standardisation des services Web reflète cette tendance à pouvoir disposer sur le marché d'architectures techniques « prêtes à intégrer ».

Une architecture technique est en effet immédiatement réutilisable pour les différentes composantes fonctionnelles d'un même système d'entreprise.

LES MODÈLES DE CYCLE DE VIE SPÉCIFIQUES:



Le modèle 2TUP

Un processus itératif et incrémental piloté par les risques

ITÉRATION

Une itération est une séquence distincte d'activités avec un plan de base et des critères d'évaluation, qui produit une release (interne ou externe). Le contenu d'une itération est porteur d'améliorations ou d'évolutions du système et il peut être évalué par les utilisateurs.

INCRÉMENT

Un incrément est la différence (delta) entre deux releases produites à la fin de deux itérations successives.

LES MODÈLES DE CYCLE DE VIE SPÉCIFIQUES:



Le modèle 2TUP

Un processus itératif et incrémental piloté par les risques

Une ou plusieurs itérations s'inscrivent dans chacune des phases de gestion de projet et servent en conséquence à réaliser l'objectif propre à chaque phase.

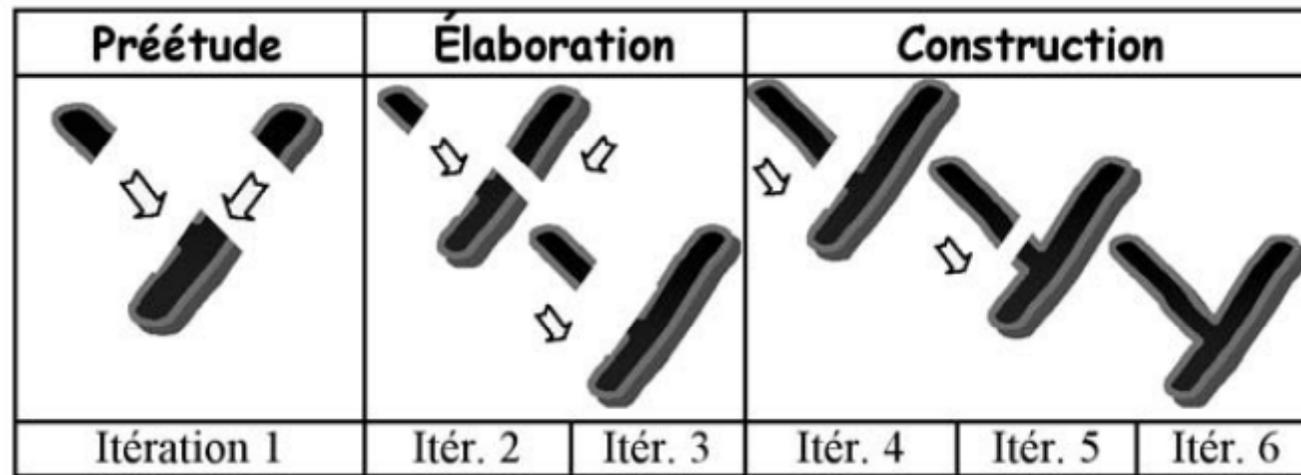
- En **préétude** (*inception*), les itérations servent à évaluer la valeur ajoutée du développement et la capacité technique à le réaliser.
- En phase **d'élaboration**, elles servent à confirmer l'adéquation du système aux besoins des utilisateurs et à livrer l'architecture de base.
- En phase de **construction**, elles servent à livrer progressivement toutes les fonctions du système.
- En phase de **transition**, elles servent à déployer le système sur les sites opérationnels.

LES MODÈLES DE CYCLE DE VIE SPÉCIFIQUES:



Le modèle 2TUP

Le processus de développement en Y se reproduit à différents niveaux d'avancement en se terminant sur la livraison d'une nouvelle release.



Temps

scénario typique d'avancement sur les 3 premières phases de gestion de projet.

LES MODÈLES DE CYCLE DE VIE SPÉCIFIQUES:



Le modèle 2TUP

L'itération 1 développe les fonctions de validation du principe du système et intègre les outils prévus pour le développement.

- **L'itération 2** est focalisée sur l'architecture ; elle peut être considérée comme le prototype de réalisation technique.

- **L'itération 3** avance dans la réalisation des fonctions les plus prioritaires de manière à présenter une première version de déploiement pour les utilisateurs. Elle permet entre-temps d'améliorer et de compléter l'architecture technique.

- Les itérations suivantes avancent dans la réalisation des fonctions jusqu'à l'obtention complète du système initialement envisagé.

LES MODÈLES DE CYCLE DE VIE SPÉCIFIQUES:



Le modèle 2TUP

Une itération – toutes les activités du processus en Y.

Néanmoins l'effort n'est pas toujours identique suivant la phase de développement du projet.

Exemple:

Effort relatif par activité	Itération 1 (préétude)	Itération 2 (élaboration)	Itération 4 (construction)
Capture des besoins fonctionnels	16 %	1 %	2 %
Analyse	22 %	9 %	4 %
Capture des besoins techniques	2 %	4 %	0 %
Conception générique	5 %	14 %	2 %
Conception préliminaire	10 %	10 %	10 %
Conception détaillée	10 %	14 %	20 %
Codage et tests	28 %	26 %	30 %
Recette	7 %	22 %	32 %
(Total des activités du Y)	100 %	100 %	100 %

