Etude de cas

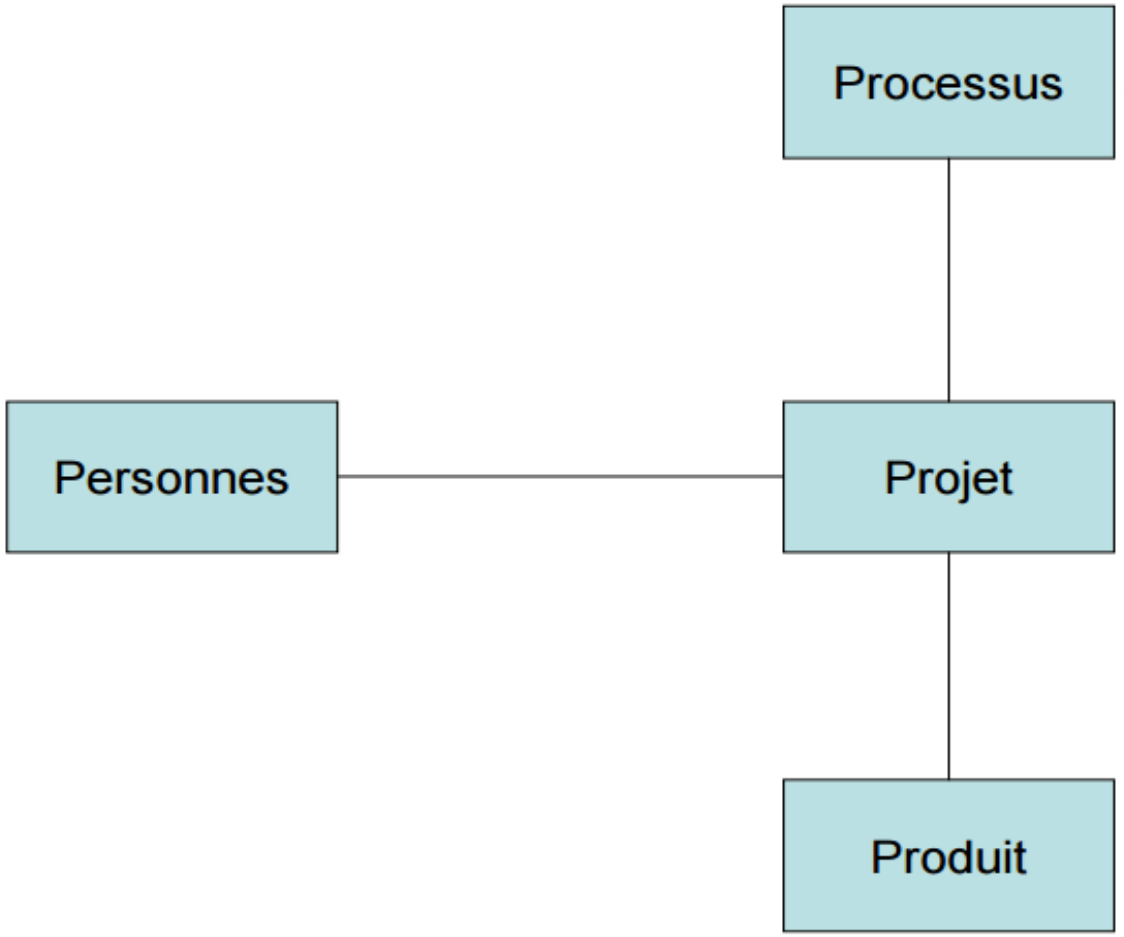
Processus Unifié

ESSABBAR DRISS
essabbar.emsi@gmail.com

Introduction

- l'objectif sous-jacent que poursuit une entreprise est d'exploiter les processus pour répondre au mieux à la demande du marché et accélérer la production de biens et de services de qualité à un prix raisonnable.

Les 4 «P» du développement logiciel



Processus

- Un processus définit une séquence d'étapes, en partie ordonnées, qui concourent à l'obtention d'un système logiciel ou à l'évolution d'un système existant.
- L'objet d'un processus de développement est de produire des logiciels de qualité qui répondent aux besoins de leurs utilisateurs dans des temps et des coûts prévisibles.

Processus unifié

- « ***Le processus unifié est un processus de développement logiciel, c'est-à-dire qu'il regroupe les activités à mener pour transformer les besoins d'un utilisateur en système logiciel » (Jacobson, Booch, Rumbaugh 1999)***

Grady Booch

Ivar Jacobson

James Rumbaugh

Qu'est ce que le Processus Unifié ?

- **Le Processus Unifié ou UP (Unified Process) est une méthode générique de développement de logiciel développée par les concepteurs d'UML.**
  - **Générique signifie qu'il est nécessaire d'adapter UP au contexte du projet, de l'équipe, du domaine et/ou de l'organisation.**
  - **Il existe donc un certain nombre de méthodes issues de UP comme par exemple RUP (Rational Unified Process), 2TUP (Two Track Unified Process) ...**

Qu'est ce que le Processus Unifié ?

- **L'objectif d'un processus unifié est de maîtriser la complexité des projets informatiques en diminuant les risques.**
- **UP répond aux préoccupations suivantes :**
  - **QUI participe au projet ?**
  - **QUOI, qu'est-ce qui est produit durant le projet ?**
  - **COMMENT doit-il être réalisé ?**
  - **QUAND est réalisé chaque livrable ?**

Les principes d'UP

- **Le processus de développement UP, associé à UML, met en œuvre les principes suivants :**
  - **Le processus unifié utilise le langage UML**
  - **Processus guidé par les cas d'utilisation,**
  - **Processus itératif et incrémental,**
  - **Processus centré sur l'architecture,**
  - **Processus orienté par la réduction des risques.**

Les principes d'UP

Processus guidé par les cas d'utilisation

Les cas d'utilisation :

- **Décrivent les interactions entre les acteurs et le système**
- **Permettent de capturer les vrais besoins des utilisateurs**
- **Guident tout le processus de développement.**
- **Favorisent un développement itératif ;**
  - **Chaque itération est centrée sur un sous-ensemble de cas**

Scrum : cas d'utilisation → use story

Les principes d'UP

Processus guidé par les cas d'utilisation

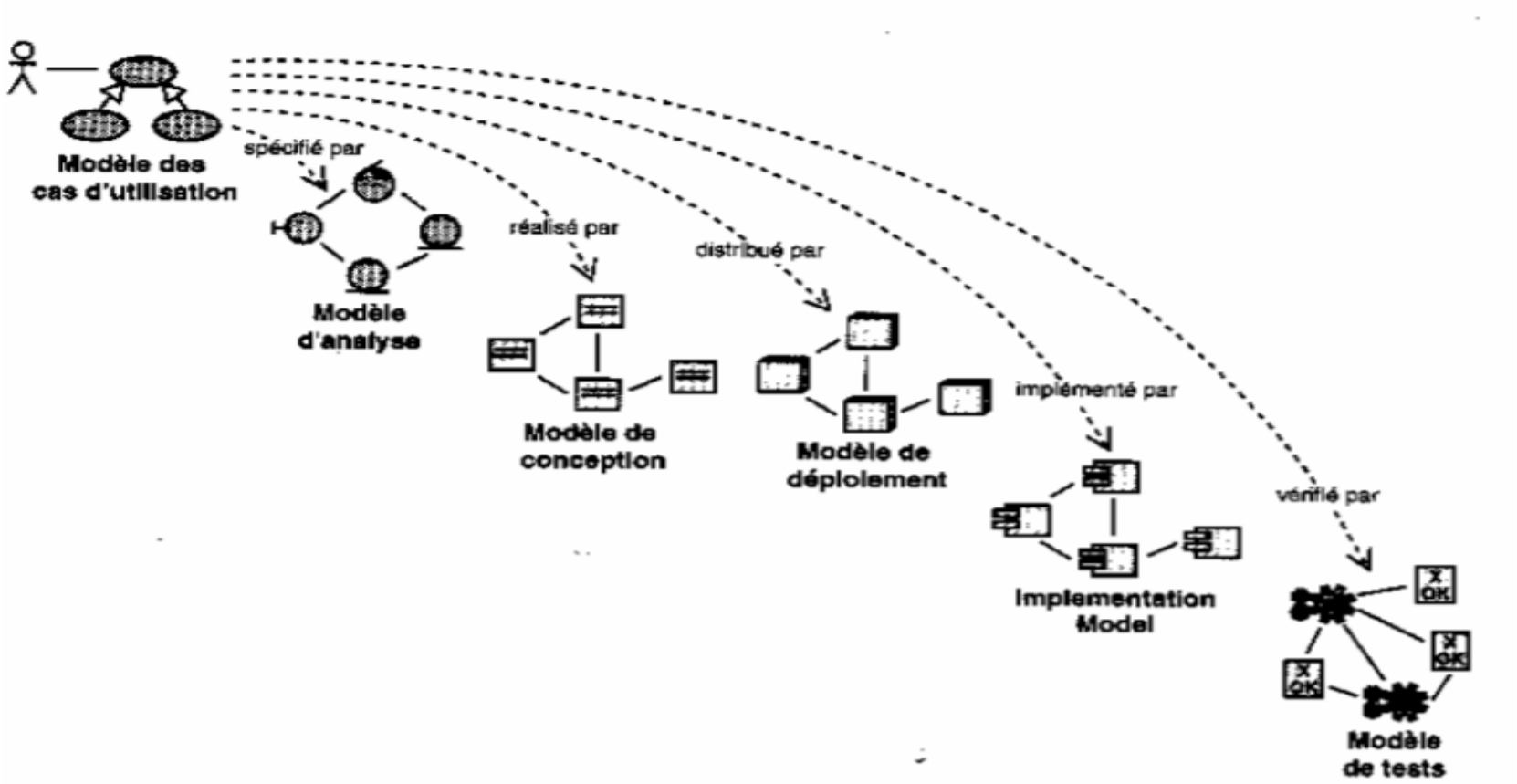
1) Le système à construire se définit d'abord avec les **utilisateurs**.

Les **cas d'utilisation** permettent d'exprimer les interactions du système avec les utilisateurs, donc de capturer les besoins.

2) Ainsi le développement peut se décomposer par cas d'utilisation et la réception du logiciel sera elle aussi articulée par cas d'utilisation.

Les principes d'UP

Processus guidé par les cas d'utilisation



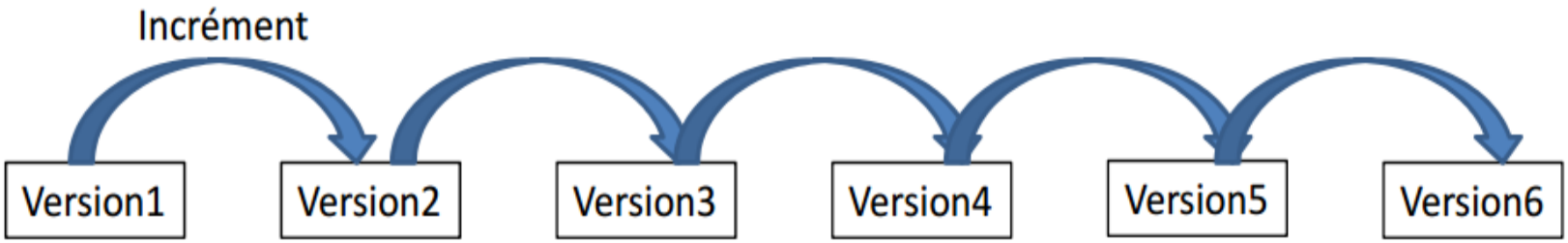
Les principes d'UP

Processus itératif et incrémental

- UP utilise le principe de développement par itérations successives.
  - La réalisation de maquette et prototype constitue la réponse pratique à ce principe.
- Le développement progressif, par incrément, est aussi recommandé (décomposition du système en cas d'utilisation).

Les principes d'UP

Processus itératif et incrémental



Plusieurs versions avant d'obtenir toutes les spécifications

Chaque version est testée, reçue par un utilisateur

Les principes d'UP

Processus centré sur l'architecture

Le rôle de l'architecture logicielle est comparable à celle que joue l'architecte dans la construction d'un bâtiment.

Le bâtiment est envisagé de différents points de vue: structure, services, conduite de chauffage, plomberie, etc.

Ce regard multiple dessine une image complète du bâtiment avant le début de la construction.

l'architecture d'un système logiciel.

Les principes d'UP

Processus centré sur l'architecture

Les auteurs d'UP mettent en avant la préoccupation de l'architecture du système dès le début des travaux d'analyse et de conception

Il est important de définir le plus tôt possible l'architecture qui sera retenue pour le développement, l'implémentation et ensuite le déploiement du système

Les principes d'UP

Processus centré sur l'architecture

Pourquoi une architecture?

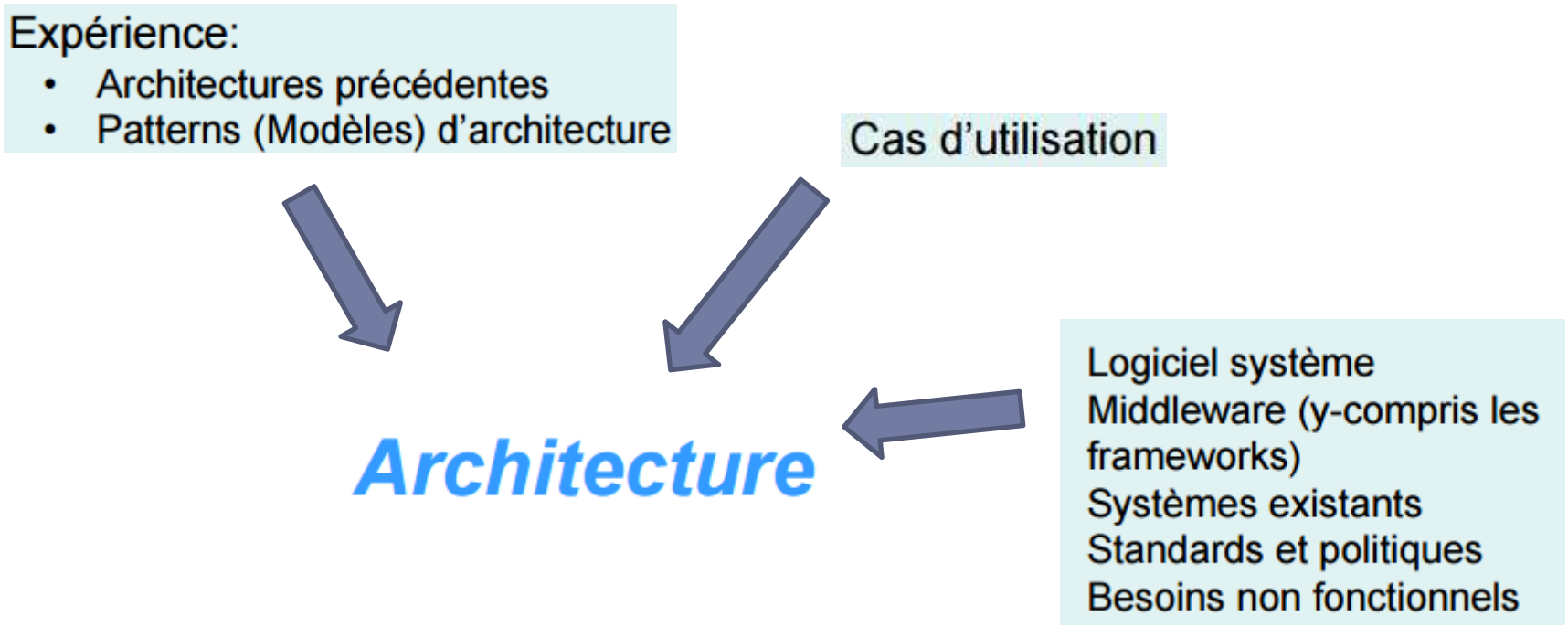
Il nous faut une architecture pour:

- **Comprendre le système**
- **Organiser le développement**
- **Favoriser la réutilisation**
- **Faire évoluer le système**
- **Réaction aux changements de l'environnement**

Les principes d'UP

Processus centré sur l'architecture

Facteurs d'influence de choix architecturaux



Les principes d'UP

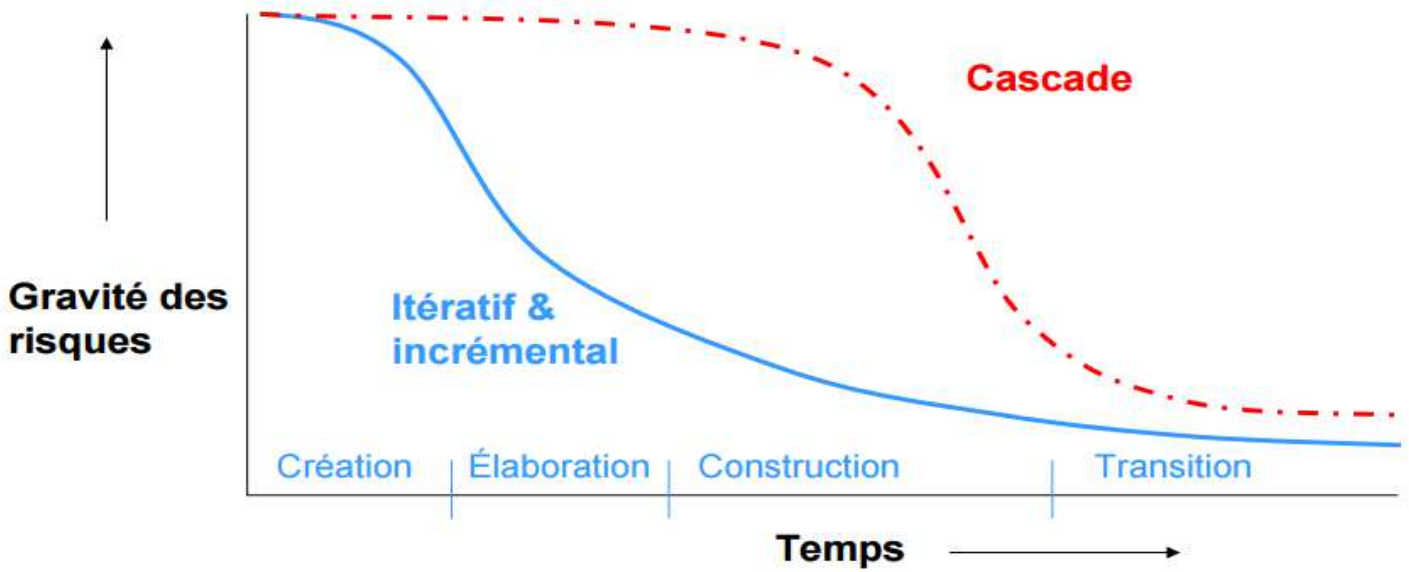
Processus orienté par la réduction des risques

Un risque est une variable d'un projet qui met en danger, voire compromet totalement, la réussite du projet en question.

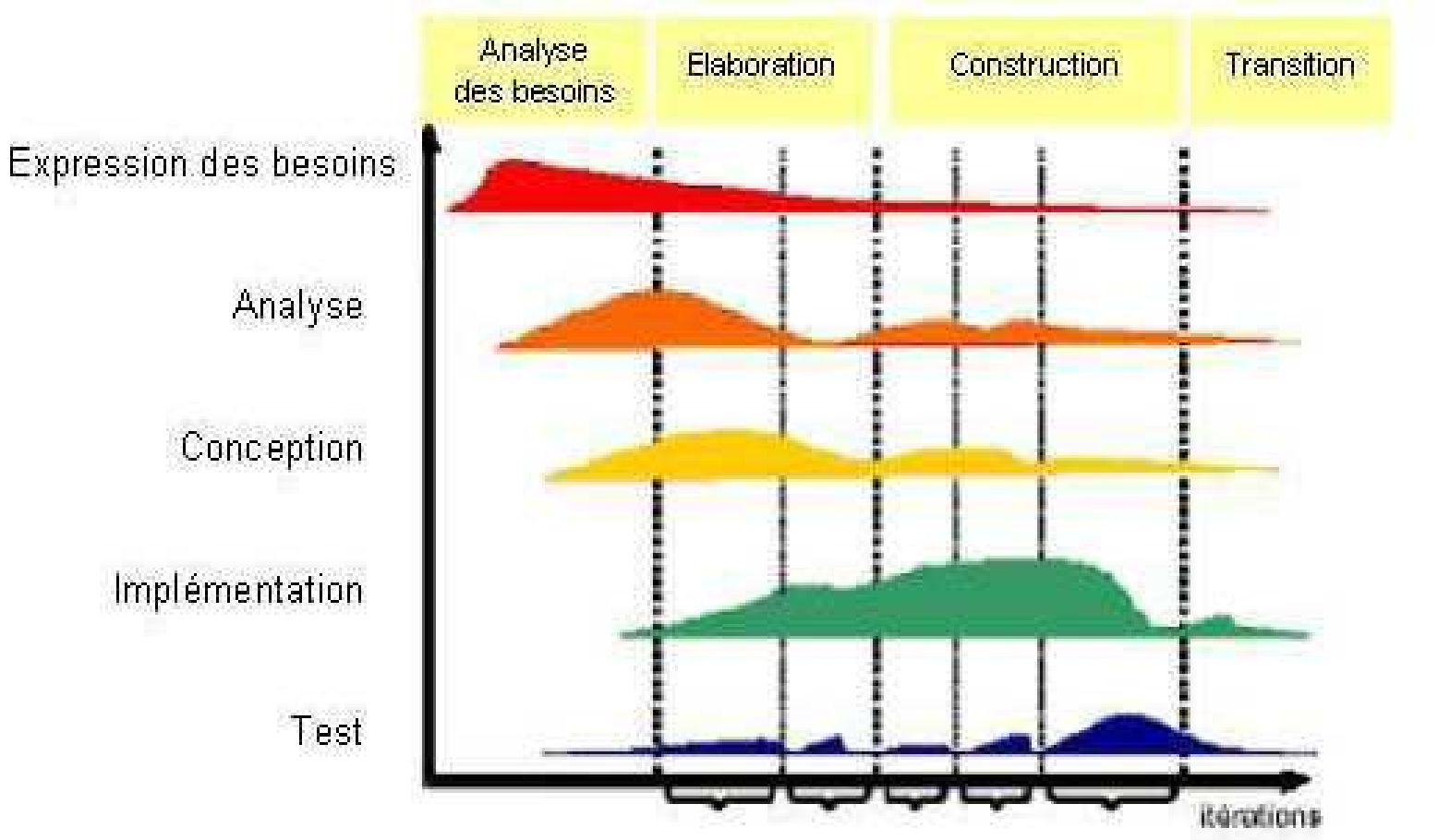
Les principes d'UP

Processus orienté par la réduction des risques

UP contribue à la diminution des risques au fur et à mesure du déroulement des itérations successives.



L'architecture bidirectionnelle



L'architecture bidirectionnelle

- **UP gère le processus de développement par deux axes.**
  - **L'axe vertical représente les principaux enchaînements d'activités, qui regroupent les activités selon leur nature.**
  - **L'axe horizontal représente le temps et montre le déroulement du cycle de vie du processus**

Les quatre phases

Phase 1 : Etude préliminaire (inception)

- Phase très courte (souvent une seule itération)
- Etape préliminaire à l'élaboration
  - ➢ Déterminer la faisabilité, les risques et le périmètre du projet

✓ Que doit faire le système ?
✓ A quoi pourrait ressembler l'architecture ?
✓ Quels sont les risques ?
✓ Estimation approximative des coûts et des délais
✓ Accepter le projet?

Les quatre phases

Phase 2 : Elaboration

- **Quelques itérations courtes et de durée fixe, pilotées par les risques**
  - ☑ **Identification et stabilisation de la plupart des besoins**
  - ☑ **Conception de l'architecture de base**

Les quatre phases

Phase 3 : Construction

- Phase la plus coûteuse (>50% du cycle)
  - ☑ Développement par incréments ;
  - ☑ Architecture stable malgré des changements mineurs
  - ☑ Le produit contient tout ce qui avait été planifié ;
    - Il reste quelques erreurs

Les quatre phases

Phase 4 : Transition

- Produit délivré (version béta)
- Correction du reliquat d'erreurs
- Essai et amélioration du produit,
- Formation des utilisateurs, installation de l'assistance en ligne…

Les activités

- **Ces phases se décomposent en activités :**
  - 1) **Expression des besoins:**
    - **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)**

Artefacts obtenus :

- **Liste des fonctions**
- **Modèle des cas d'utilisation**
- **Exigences supplémentaires**
- **Prototypage de l'IHM**
- **Plan de tests de validation**

Les activités

- 2) **Analyse** : **L'objectif de l'analyse est d'accéder à une compréhension des besoins et des exigences du client.**

Artefacts obtenus :

- **Diagramme de classe d'analyse**
- **…**

Les activités

- 3)**Conception** : La conception permet d'acquérir une compréhension approfondie des contraintes liées au **langage de programmation**, à l'utilisation des **composants** et **framework** et au **système d'exploitation**.

Artefacts obtenus :

- **diagramme de classes de conception**
- **diagramme d'interaction (séquence)**
- **diagramme d'activités et/ou d'états**
- **diagramme de déploiement**
- **plan de tests d'intégration**

Les activités

- **4) Implémentation :**
  - **la programmation. le résultat de la conception pour implémenter le système sous formes de composants**

Artefacts obtenus :

- code source, scripts, binaires
- diagramme de composant
- diagramme de déploiement
- plan de tests unitaires

Les activités

- 5)Test : 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.

Artefacts obtenus :

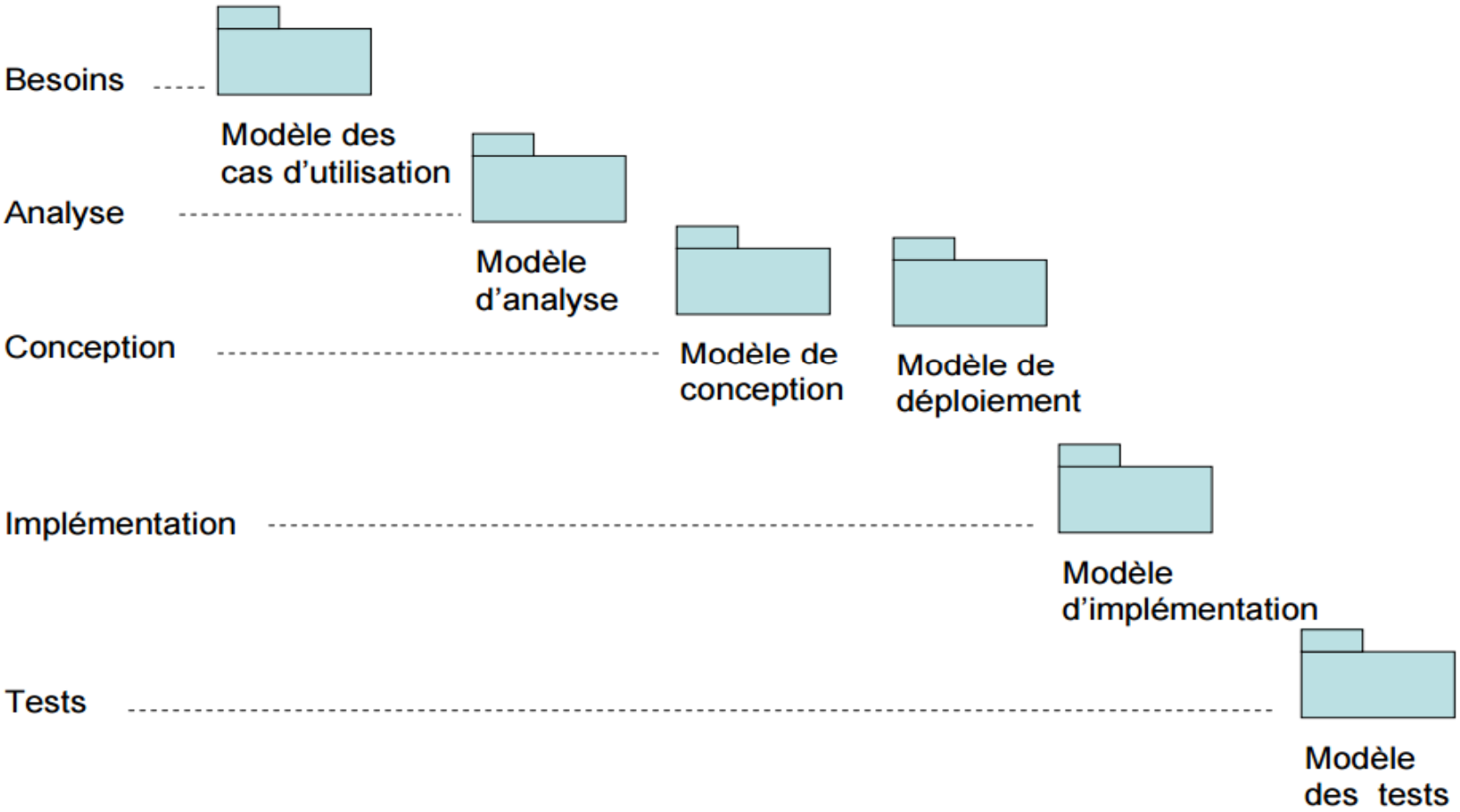
- Rapport des tests unitaires, d'intégration et de validation

- 6)Déploiement

Artefacts obtenus :

- manuels d'installation et d'utilisation

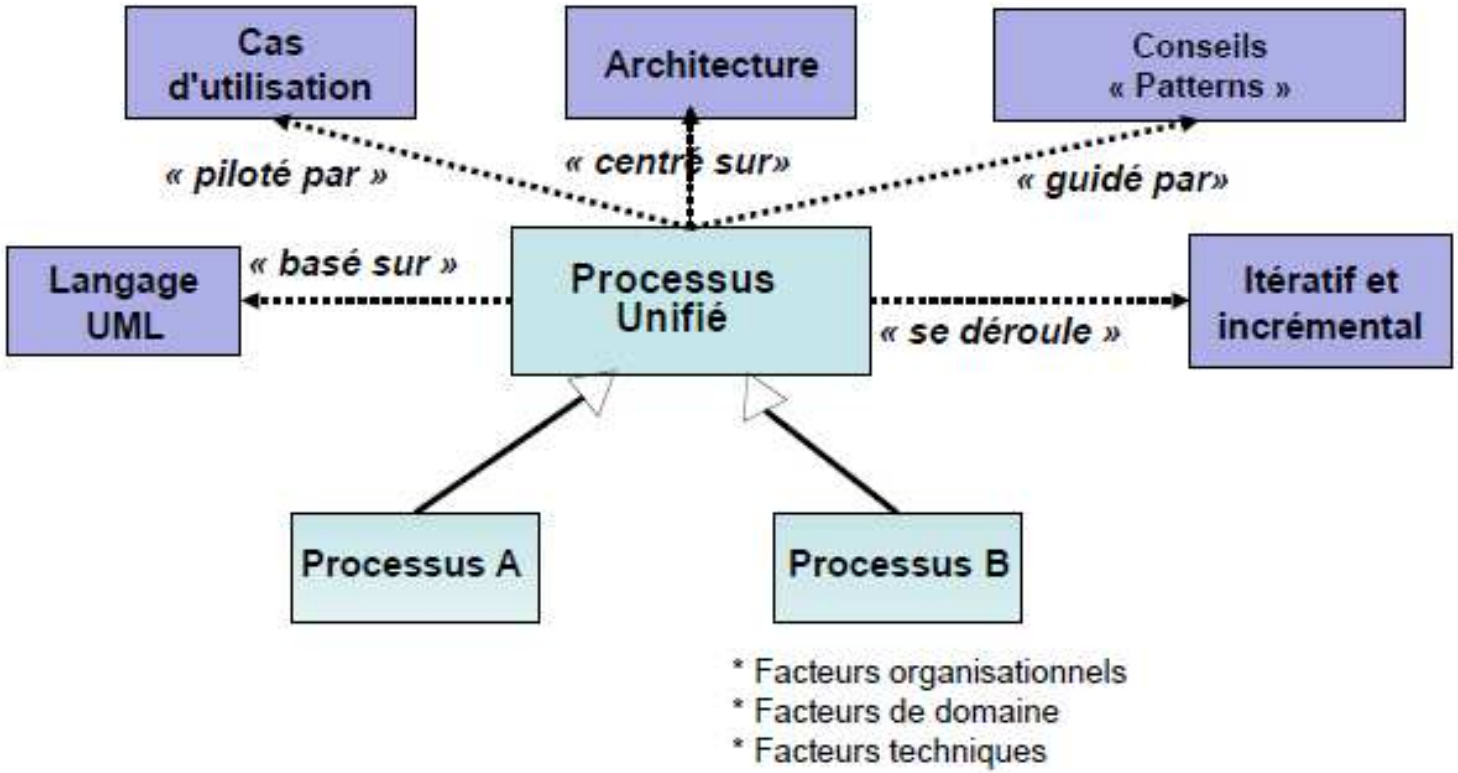
Les activités



UP – Implémentations

- **UP a plusieurs implémentation et / ou variation:**
  - RUP (Rational Unified Process)
  - 2TUP (2 tracks unified process)
  - Agile Unified Process (AUP) par Scott W. Ambler
  - Basic Unified Process (BUP) par IBM
  - Enterprise Unified Process (EUP), une extension de RUP
  - Oracle Unified Method (OUM),
  - ...

Résumé


Cas d'utilisation
Architecture
Conseils « Patterns »
« piloté par »
« centré sur»
« guidé par»
Langage UML
« basé sur »
Processus Unifié
Itératif et incrémental
« se déroule »
Processus A
Processus B
* Facteurs organisationnels
* Facteurs de domaine
* Facteurs techniques

Etude de cas

Processus Unifié – 2TUP

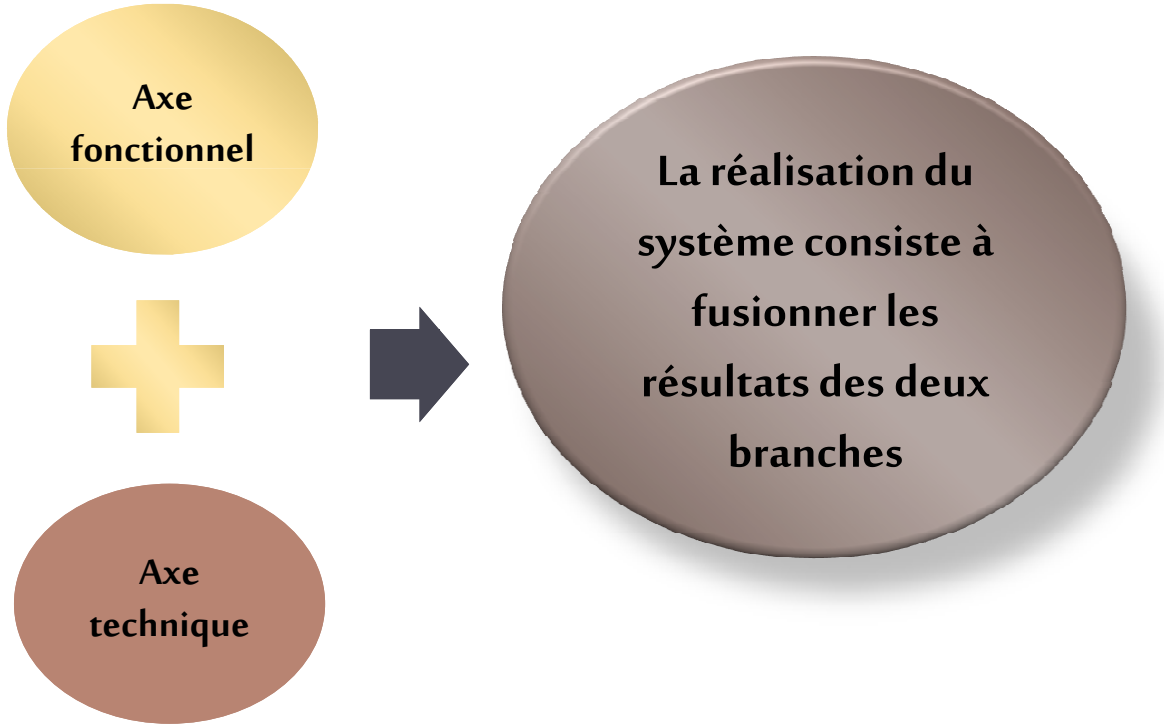
ESSABBAR DRISS
essabbar.emsi@gmail.com

Introduction

- **2TUP signifier « 2 Track Unified Process ».**
- **Processus proposé par Valtech (consulting), présenté dans**
  - **Pascal Roques, Franck Vallée (2004) UML2 en action (3ème édition), Eyrolles, 386 pp.**
- **Objectif :**
  - **Prendre en compte les contraintes de changement imposées aux systèmes d'information des organisations**

Principe

- **2 track signifie littéralement que le processus suit deus chemins :**
  - **Fonctionnel**
  - **Technique**



Introduction

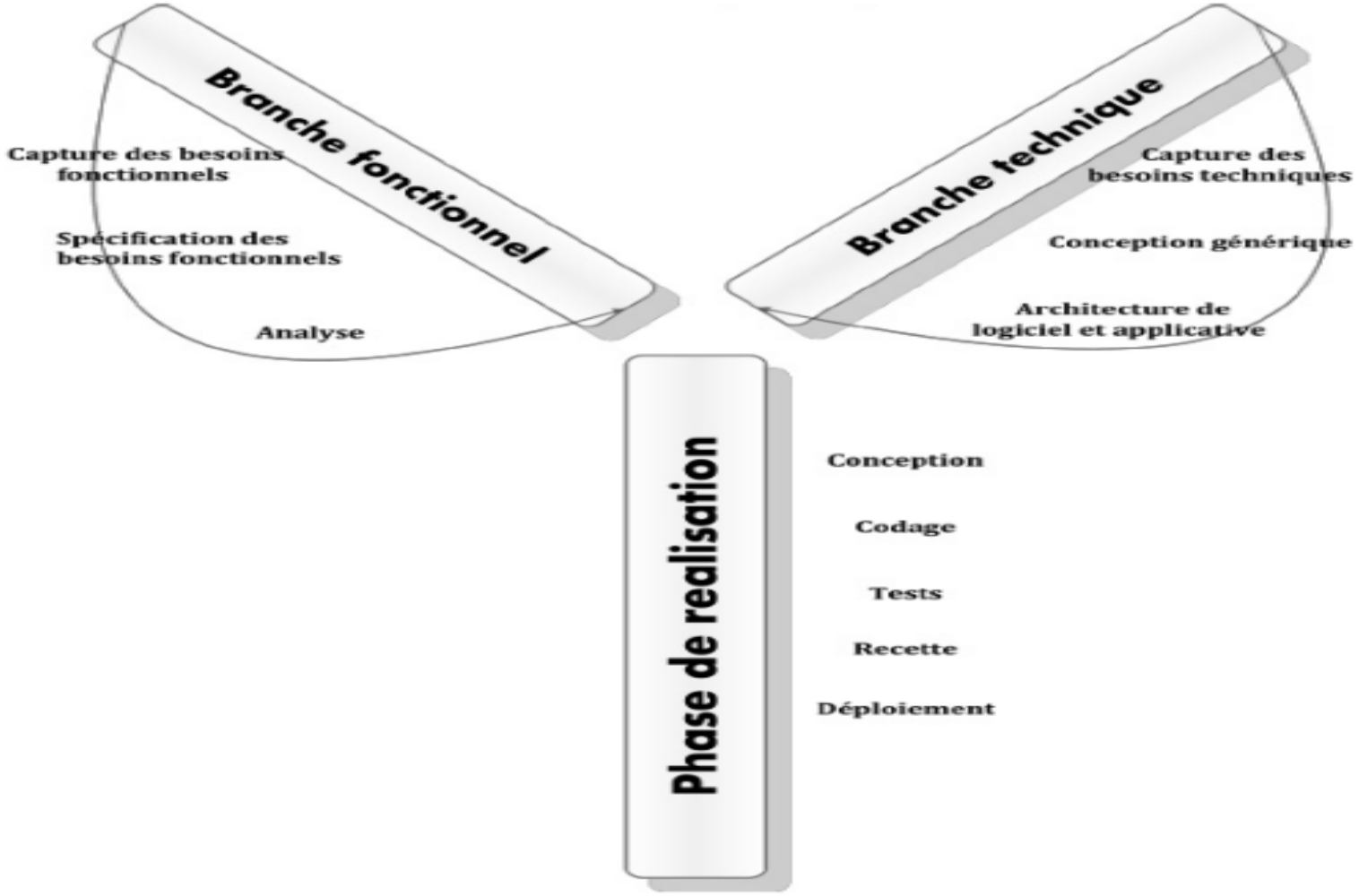
Exemple

- **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 ;**
- **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 ;**

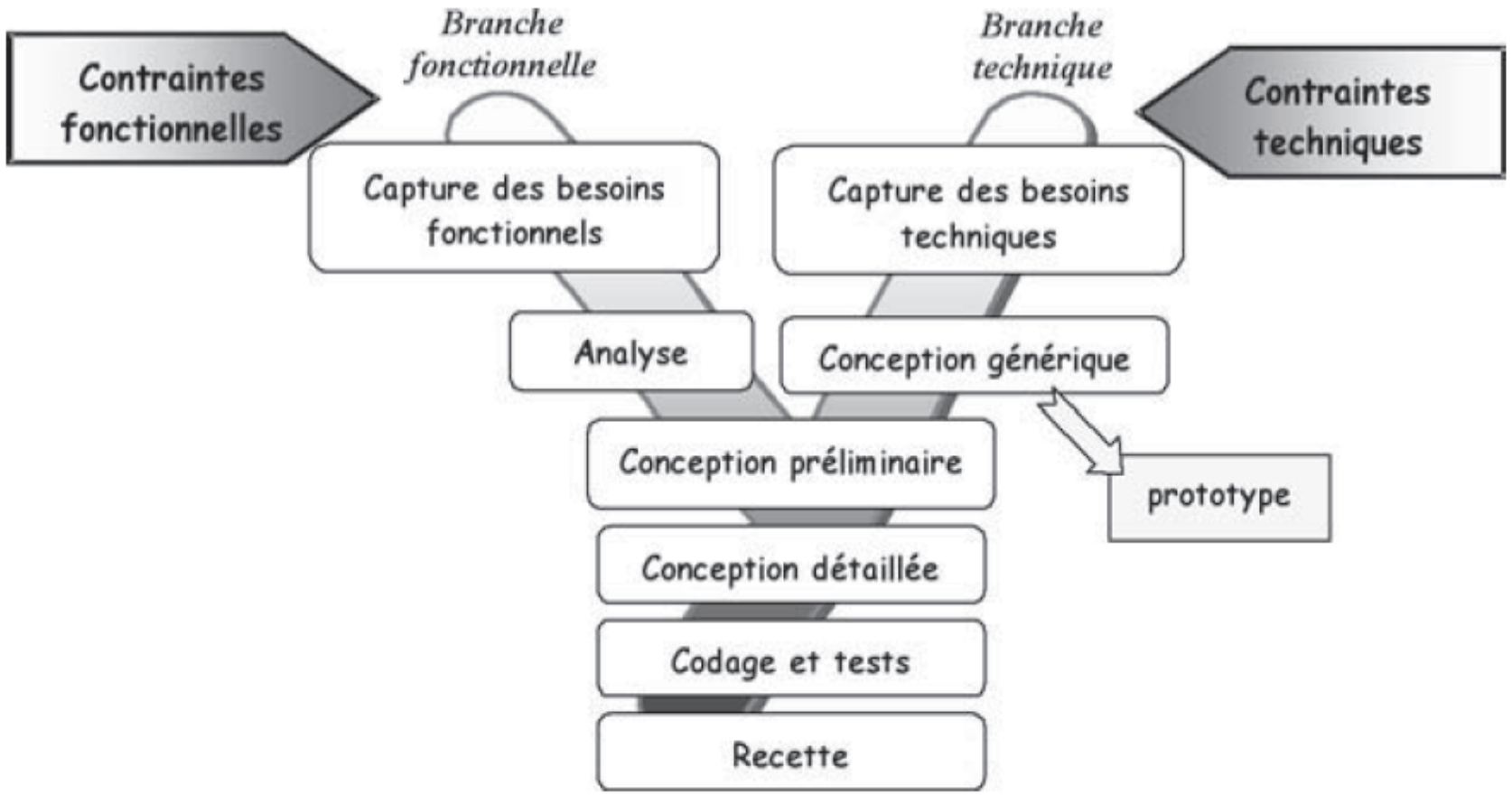
Caractéristiques du 2 TUP

- **Itératif et incrémental piloté par les risques**
- **Piloté par les exigences des utilisateurs**
- **Modélisé avec UML**
- **Centré sur l'architecture.**

Processus 2TUP



Processus 2TUP



Branche fonctionnelle

- **Les principales étapes de la branche fonctionnelle se présentent comme suit :**
  - **L'étape capture des besoins fonctionnels: Cette phase a pour objectif de définir :**
    - **La frontière fonctionnelle entre le système et son environnement.**
    - **Les activités attendues des différents utilisateurs par rapport au système.**
  - **L'étape d'analyse: consiste à étudier précisément les spécifications fonctionnelles de manière à obtenir une idée de ce que va réaliser le système en terme de métier**

Branche technique

- **Les principales étapes de la branche technique se présentent comme suit**
  - **L'étape capture des besoins techniques : Cette étape recense toutes les contraintes sur les choix de technologies pour la conception du système. Les outils (et les Frameworks)et le matériel sélectionnés ainsi que la prise en compte des contraintes d'intégration avec l'existant (pré requis d'architecture technique)**

Branche technique

- **Conception générique: Il s'agit du découpage en composants nécessaires à la construction de l'architecture technique. Il est généralement conseillé de réaliser un prototype pour assurer la validité de l'architecture.**

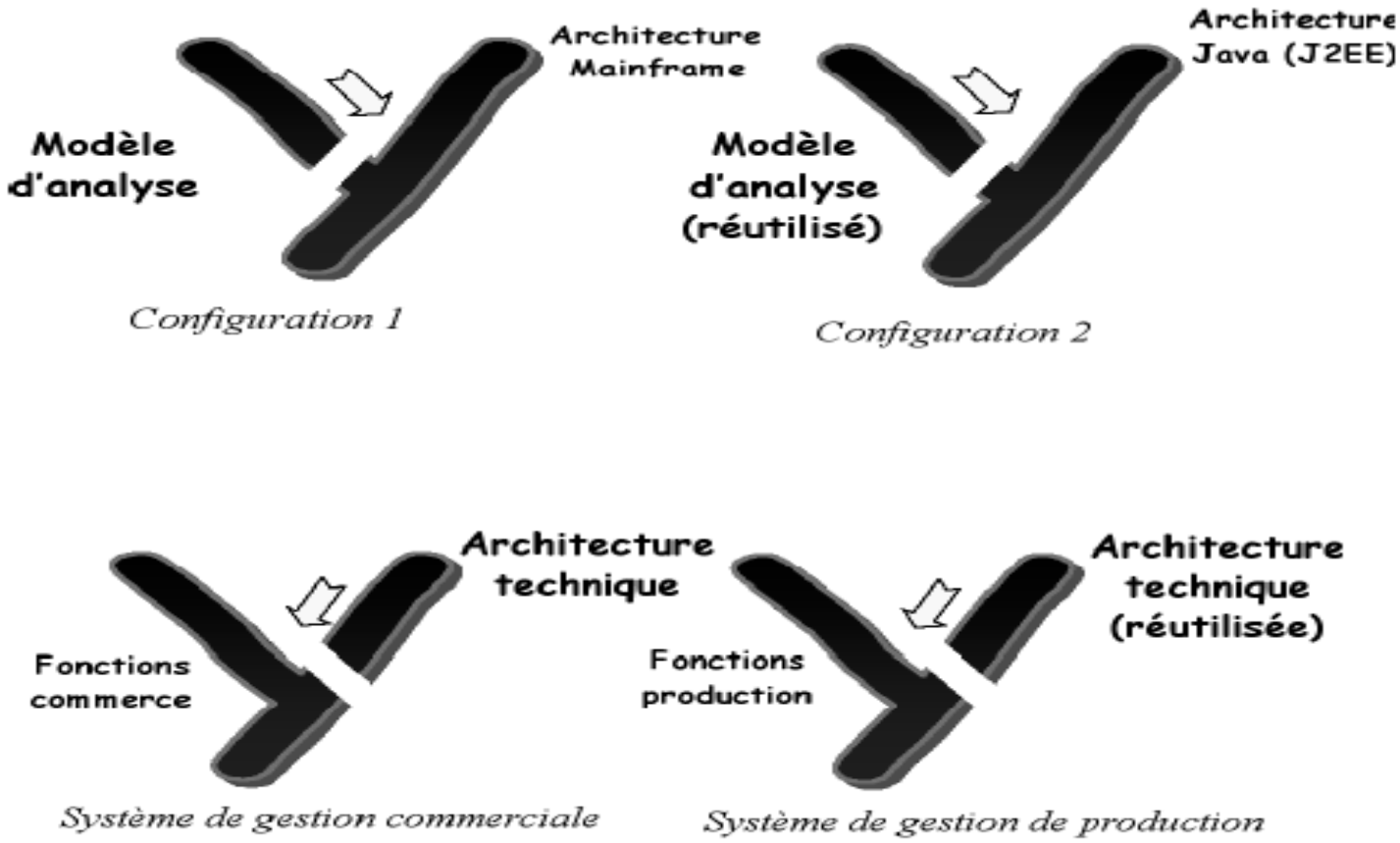
Phase conception - réalisation

- **Les principales étapes de cette branche se présentent comme suit :**
  - **L'étape conception préliminaire : Cette étape permet de produire le modèle de conception système. Ce dernier organise le système en composants, délivrant les services techniques et fonctionnels, Ce qui induit le regroupement des informations des branches technique et fonctionnelle.**

Phase conception - réalisation

- **L'étape conception détaillée: permet d'étudier comment réaliser chaque composant. le résultat fournit l'image prête { fabriquer du système complet.**
- **L'étape de codage : permet d'effectuer la production des composants et les tests des unités de code au fur et à mesure de leur réalisation.**
- **L'étape de recette : consiste à valider les fonctionnalités du système développé.**

2TUP et la réutilisabilité

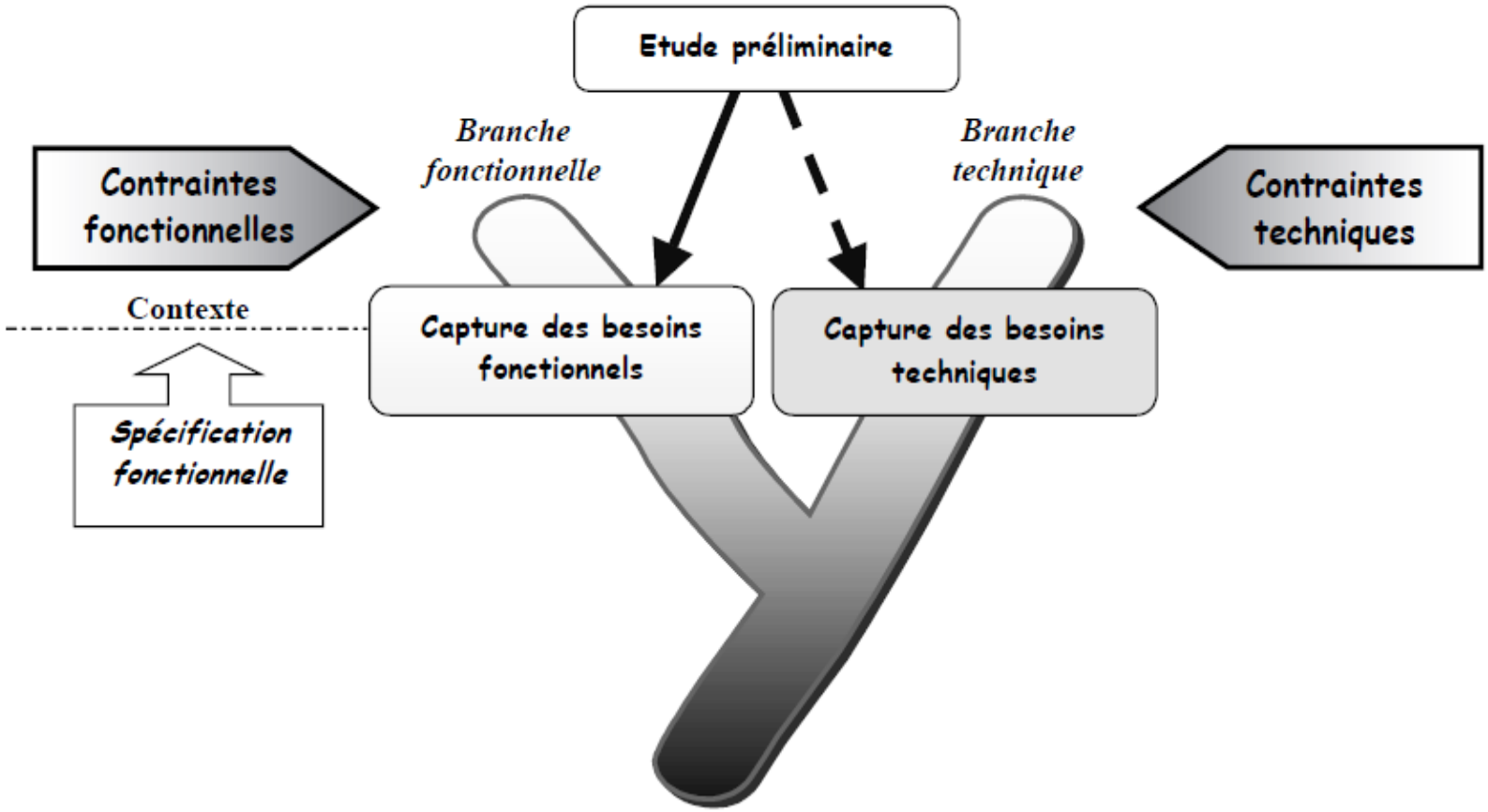


Etude de cas

Etude de Cas- 2TUP

ESSABBAR DRISS
essabbar.emsi@gmail.com

Etude préliminaire



Etude préliminaire

- **L'étude préliminaire (ou préétude) est la toute première étape de notre processus de développement.**
- **Elle survient à la suite d'une décision de démarrage de projet, et consiste à effectuer un premier repérage des besoins fonctionnels et opérationnels, en utilisant principalement le texte, ou des diagrammes très simples.**
- **Elle prépare les étapes plus formelles de capture des besoins fonctionnels et de capture des besoins techniques**

Etude préliminaire

- **Une fois ce premier recueil de besoins effectué, la description du contexte du système peut commencer. Elle consiste en trois activités successives :**
  - **l'identification des acteurs,**
  - **l'identification des messages,**
  - **la réalisation des diagrammes de contexte.**