

# L3 / ISI2

## Conduite de Projet

Université de Paris 1 Panthéon - Sorbonne

UFR..

Enseignant : Olivier MAQUET

Année universitaire 2005-2006

### **Préambule :**

**Ce document n'est pas un support de cours. Beaucoup de sujets ne sont pas détaillés dans ce document. Par contre, ils le seront à l'oral par l'enseignant.**

**Il est fourni aux étudiants afin de les aider dans leur prise de notes pendant les séances de cours.**

# Table des matières

<b>1. INTRODUCTION.....</b>	<b>5</b>
<b>1.1 INFORMATIONS PRELIMINAIRES .....</b>	<b>5</b>
1.1.1 Remarque concernant ce support .....	5
1.1.2 Quelles notes ? .....	5
<b>1.2 OBJECTIF DE M12 – CONDUITE DE PROJET .....</b>	<b>5</b>
1.2.1 Objectif du cours .....	5
<b>1.3 INTRODUCTION A LA DEMARCHE DE CONDUITE DE PROJET.....</b>	<b>6</b>
<b>1.4 DIFFERENTS TYPES DE PROJETS.....</b>	<b>8</b>
<b>1.5 LES ACTEURS D’UN PROJET INFORMATIQUE .....</b>	<b>9</b>
1.5.1 L’organisation d’une Direction des Systèmes d’Information (DSI) d’une entreprise .....	9
1.5.2 Les acteurs de la maîtrise d’œuvre.....	9
1.5.3 Les acteurs de la maîtrise d’ouvrage.....	9
1.5.4 Implication MOE & MOA dans un projet.....	9
1.5.5 Les instances de pilotage .....	10
<b>1.6 VOCABULAIRE (A EXPLIQUER).....</b>	<b>10</b>
1.6.1 Vocabulaire.....	10
1.6.2 Abréviations .....	10
<b>2. LES OUTILS.....</b>	<b>12</b>
<b>2.1 LES OUTILS DE L’EP JUSQU’A LA MISE EN PRODUCTION .....</b>	<b>12</b>
2.1.1 Les différents types d’outils.....	12
2.1.2 Tableau de synthèse Outils / Etapes de la démarche.....	14
<b>2.2 LES OUTILS D’ADMINISTRATION DU SYSTEME.....</b>	<b>14</b>
<b>3. RAPPELS MERISE &amp; UML (METHODOLOGIES DE MODELISATION).....</b>	<b>15</b>
<b>3.1 PRESENTATION DE LA DEMARCHE PAR NIVEAUX.....</b>	<b>15</b>
<b>3.2 UML (RAPPEL).....</b>	<b>15</b>
3.2.1 Le diagrammes des cas d’utilisation (Use case).....	15
3.2.2 Le diagrammes des classes.....	15
3.2.3 Le diagrammes d’états - transitions .....	15
3.2.4 Le diagrammes de séquences.....	15
<b>3.3 MERISE (RAPPEL).....</b>	<b>16</b>
3.3.1 Le modèle des flux (rappel).....	16
3.3.2 MCD (rappel).....	16
3.3.3 MLD relationnel (rappel).....	16
3.3.4 MCT (rappel).....	16
3.3.5 MOT (rappel).....	16
<b>4. SCHEMA DIRECTEUR .....</b>	<b>17</b>
<b>4.1 LES 2 NIVEAUX DE SD.....</b>	<b>17</b>
<b>4.2 SD STRATEGIQUE .....</b>	<b>17</b>
4.2.1 Définition .....	17
4.2.2 Les acteurs .....	17
<b>4.3 SD DU SI.....</b>	<b>18</b>
4.3.1 Définition .....	18
4.3.2 Les acteurs .....	18
<b>5. ETUDE PREALABLE.....</b>	<b>19</b>
<b>5.1 OBJECTIFS DE L’ETAPE.....</b>	<b>19</b>
<b>5.2 LES PHASES DE L’EP.....</b>	<b>19</b>
5.2.1 Lancement de l’EP (Ph1).....	19
5.2.2 Bilan de l’existant (Ph2).....	20
5.2.3 Définition des orientations (Ph3) .....	20
5.2.4 Modélisation du système cible / SER (Ph4).....	21

5.2.4.1	Modélisation du fonctionnel.....	21
5.2.4.2	Initialisation de l'architecture technique cible .....	21
5.2.5	Processus de validation et décision (Ph5).....	23
<b>5.3</b>	<b>LIVRABLES DE L'EP.....</b>	<b>25</b>
<b>5.4</b>	<b>IMPORTANCE DU PROCESSUS QUALITE.....</b>	<b>25</b>
<b>5.5</b>	<b>LES ACTEURS DE L'EP.....</b>	<b>25</b>
<b>6.</b>	<b>ETUDE DETAILLEE.....</b>	<b>27</b>
<b>6.1</b>	<b>OBJECTIFS DE L'ED .....</b>	<b>27</b>
<b>6.2</b>	<b>LES PHASES DE L'ED .....</b>	<b>27</b>
6.2.1	Lancement (Ph1).....	27
6.2.2	Complément des SFG (Ph2).....	28
6.2.3	Elaboration des SFD Données + Traitements (Ph3).....	28
6.2.3.1	Elaboration de N&S (Ph3.1) .....	29
6.2.3.2	Elaboration des SFD (Ph3.2).....	29
6.2.4	Définition de l'architecture physique (Ph4).....	30
6.2.5	Validation et décision (Ph5).....	31
<b>6.3</b>	<b>LES ACTEURS DE L'ED.....</b>	<b>31</b>
<b>6.4</b>	<b>LIVRABLES DE L'ED .....</b>	<b>32</b>
<b>7.</b>	<b>ETUDE TECHNIQUE.....</b>	<b>33</b>
<b>7.1</b>	<b>OBJECTIFS DE L'ET .....</b>	<b>33</b>
<b>7.2</b>	<b>LES PHASES .....</b>	<b>33</b>
7.2.1	Lancement (Ph1).....	33
7.2.2	Elaboration des spécifications techniques (Ph2).....	34
7.2.2.1	Préparation du cadre du développement .....	34
7.2.2.2	Identification des composants logiciels à développer .....	34
7.2.3	Mise en œuvre de la plateforme de développement (Ph3) .....	36
7.2.4	Fin de l'Etude Technique .....	38
<b>7.3</b>	<b>LES LIVRABLES.....</b>	<b>38</b>
<b>7.4</b>	<b>LES ACTEURS DE L'ET.....</b>	<b>38</b>
<b>8.</b>	<b>REALISATION ET TESTS UNITAIRES .....</b>	<b>39</b>
<b>8.1</b>	<b>OBJECTIFS DE LA REALISATION .....</b>	<b>39</b>
<b>8.2</b>	<b>LES PHASES .....</b>	<b>39</b>
8.2.1	Lancement (Ph1).....	39
8.2.2	Développement + tests unitaires (Ph2).....	40
<b>8.3</b>	<b>LES LIVRABLES.....</b>	<b>41</b>
<b>8.4</b>	<b>LES ACTEURS DE LA REALISATION.....</b>	<b>41</b>
<b>9.</b>	<b>QUALIFICATION MOE .....</b>	<b>42</b>
<b>9.1</b>	<b>OBJECTIFS .....</b>	<b>42</b>
<b>9.2</b>	<b>LES PHASES .....</b>	<b>42</b>
9.2.1	Lancement (Ph1).....	43
9.2.2	Préparation de la Qualification (Ph2).....	43
9.2.3	Exécution de la qualification (Ph3).....	44
9.2.4	Préparation de la livraison à la MOA (Ph4).....	44
<b>9.3</b>	<b>LES LIVRABLES.....</b>	<b>45</b>
<b>9.4</b>	<b>LES ACTEURS DE LA QUALIFICATION .....</b>	<b>45</b>
<b>10.</b>	<b>RECETTE .....</b>	<b>46</b>
<b>10.1</b>	<b>OBJECTIFS .....</b>	<b>46</b>
<b>10.2</b>	<b>LES PHASES .....</b>	<b>46</b>
10.2.1	Lancement (Ph1).....	46
10.2.2	Préparation des cas de tests et des jeux d'essais de recette (Ph2).....	47
10.2.3	Exécution de la recette (Ph3) .....	47
<b>10.3</b>	<b>LES 2 TYPES DE RECETTE.....</b>	<b>48</b>
10.3.1	La VA .....	49
10.3.2	La VSR.....	49

<b>10.4</b>	<b>LES LIVRABLES (PRODUITS PAR LA MOA)</b> .....	<b>49</b>
<b>10.5</b>	<b>LES ACTEURS DE LA RECETTE</b> .....	<b>49</b>
<b>11.</b>	<b>PREPARATION A LA MISE EN OEUVRE</b> .....	<b>51</b>
<b>11.1</b>	<b>LES PHASES</b> .....	<b>51</b>
11.1.1	<i>Conduite du changement (Ph1)</i> .....	51
11.1.2	<i>Préparation du déploiement (Ph2)</i> .....	51
11.1.3	<i>Les acteurs</i> .....	51
11.1.4	<i>Livrables</i> .....	52
<b>12.</b>	<b>MISE EN ŒUVRE / DEPLOIEMENT</b> .....	<b>53</b>
<b>12.1</b>	<b>LES PHASES</b> .....	<b>53</b>
<b>12.2</b>	<b>DEPLOIEMENT</b> .....	<b>53</b>
12.2.1	<i>Déploiement de l'architecture technique (Ph1)</i> .....	53
12.2.2	<i>Déploiement de l'application (Ph2)</i> .....	53
12.2.3	<i>Reprise des données (Ph3)</i> .....	54
12.2.4	<i>Compléments à faire l'environnement de production (Ph4)</i> .....	54
<b>12.3</b>	<b>LOTISSEMENT</b> .....	<b>54</b>
<b>13.</b>	<b>BILAN</b> .....	<b>55</b>
<b>14.</b>	<b>GARANTIE</b> .....	<b>56</b>
<b>15.</b>	<b>MAINTENANCE / TMA</b> .....	<b>57</b>
<b>15.1</b>	<b>ORGANISATION</b> .....	<b>57</b>
<b>15.2</b>	<b>OUTILS</b> .....	<b>57</b>
<b>16.</b>	<b>LA QUALITE DANS UN PROJET INFORMATIQUE</b> .....	<b>58</b>
<b>16.1</b>	<b>OBJECTIFS D'UN PAQ</b> .....	<b>58</b>
<b>16.2</b>	<b>DANS QUELS CAS FAIRE UN PAQ ?</b> .....	<b>58</b>
<b>16.3</b>	<b>QUELS ACTEURS PARTICIPENT A LA REDACTION D'UN PAQ ?</b> .....	<b>58</b>
<b>16.4</b>	<b>CONTENU DU PAQ</b> .....	<b>58</b>
<b>17.</b>	<b>LES ADAPTATIONS DE LA DEMARCHE</b> .....	<b>60</b>
<b>17.1</b>	<b>ADAPTATION DE LA DEMARCHE DE CONDUITE STANDARD</b> .....	<b>60</b>
<b>17.2</b>	<b>AUTRES DEMARCHES / METHODES DE CONDUITE DE PROJET</b> .....	<b>60</b>
17.2.1	<i>RAD</i> .....	60
17.2.2	<i>SDM/S</i> .....	60
17.2.3	<i>Autres démarches</i> .....	60
<b>18.</b>	<b>QU'EST-CE QU'UN CHEF DE PROJET ?</b> .....	<b>61</b>
<b>18.1</b>	<b>ROLE DU CPI</b> .....	<b>61</b>
<b>18.2</b>	<b>QUALITES DU CPI</b> .....	<b>61</b>
<b>19.</b>	<b>METHODE D'EVALUATION DES CHARGES</b> .....	<b>62</b>
<b>19.1</b>	<b>RATIOS DES ETAPES</b> .....	<b>62</b>
<b>19.2</b>	<b>METHODOLOGIES DE CHIFFRAGE DES CHARGES</b> .....	<b>62</b>

# 1. Introduction

## 1.1 Informations préliminaires

- ✦ Présentation O. MAQUET.
- ✦ Présentation de Steria (Steria : acteur de recrutement de nb stagiaires de Master 2<sup>ème</sup> année) ;

Tour de table des étudiants :

- ✦ lesquels sont issus d'un IUT ? d'un BTS ? de l'IUP 1<sup>ère</sup> année ? ..
- ✦ Quelles méthodes connues : UML ? Merise ? Remora ? ...
- ✦ Quelles expériences en utilisation de méthodes ?

### 1.1.1 Remarque concernant ce support

**Ce document n'est pas un support de cours. Beaucoup de sujets ne sont pas détaillés dans ce document. Par contre, ils le seront à l'oral par l'enseignant.**

**Il est fourni aux étudiants afin de les aider dans leur prise de notes pendant les séances de cours.**

### 1.1.2 Quelles notes ?

Deux notes, dont :

- ✦ 1 note pour le projet de fin de valeur (env. 4 semaines pour faire le devoir par groupe de 3 ou 4 étudiants max.),
- ✦ 1 note pour un partiel.

## 1.2 Objectif de M12 – Conduite de projet

### 1.2.1 Objectif du cours

#### ❶ Introduction :

- Présenter le programme des 15 séances : présentation du planning (vérifier les dates des 15 séances : 30 heures).

#### ❷ Objectifs du programme des 15 séances :

- Présentation de la démarche de projet : il s'agit d'une démarche « **générique** », à adapter si nécessaire aux projets pilotés (Gros / Petit ; Forfait / Régie ; Risques élevés / Risques faibles ; Développement spécifique / Intégration d'un progiciel ; etc...).
- La démarche est décrite en termes de :
  - Organisation MOE / MOA,
  - Tâches à réaliser,
  - Livrables à produire,
  - processus qualité,
- Présenter le rôle d'un CP ou DP d'une SSII sur les projets qui doit :
  - connaître les étapes d'un projet,
  - manager un projet (chef de projet),
  - planifier et organiser les travaux,

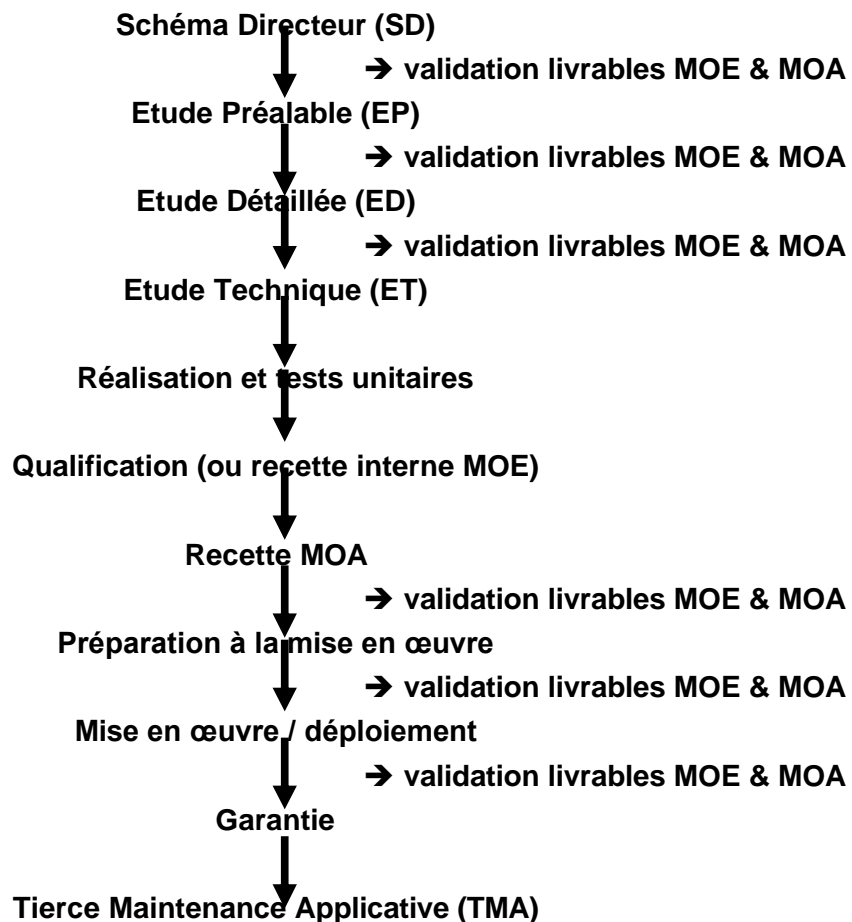
## 1.3 Introduction à la démarche de conduite de projet

Présentation d'une démarche **standard (ou « générique »)** de conduite de projet : méthode très complète, ... à adapter en fonction de la taille du projet, des contraintes, (voir le §1.4 « Différents types de projets), de la problématique de chaque projet.... Il s'agit d'une démarche prévisionnelle élaborée par le CP de la MOE.

Elle sert au CP à identifier les tâches de la MOE (voire aussi celles de la MOA), son planning, son plan de charge (entrées / sorties) sur le projet, etc...

La démarche finalisée est figée dans un PAQ du projet, qui doit faire obligatoirement l'objet d'une validation.

### Les étapes de la démarche « standard (ou « générique ») :



### Remarques sur la démarche type :

❶ La démarche de Conduite de Projet est structurée ainsi :

- X étapes,
- chaque étape est décomposée en Y phases.

Chaque étape est définie par :

- des missions à réaliser (livrables à produire),
- une organisation spécifique, côté MOA et côté MOE,
- des process, outils particuliers, ....

② Evolution de la conduite de projet depuis 10 ans :

- Evolution des méthodes de conception (des outils de + en + automatisés, ...)
- Rôle majeur des utilisateurs,
- Evolution des architectures techniques,
- Les directions Utilisatrices (la MOA), qui financent les projets, veulent y participer, ...

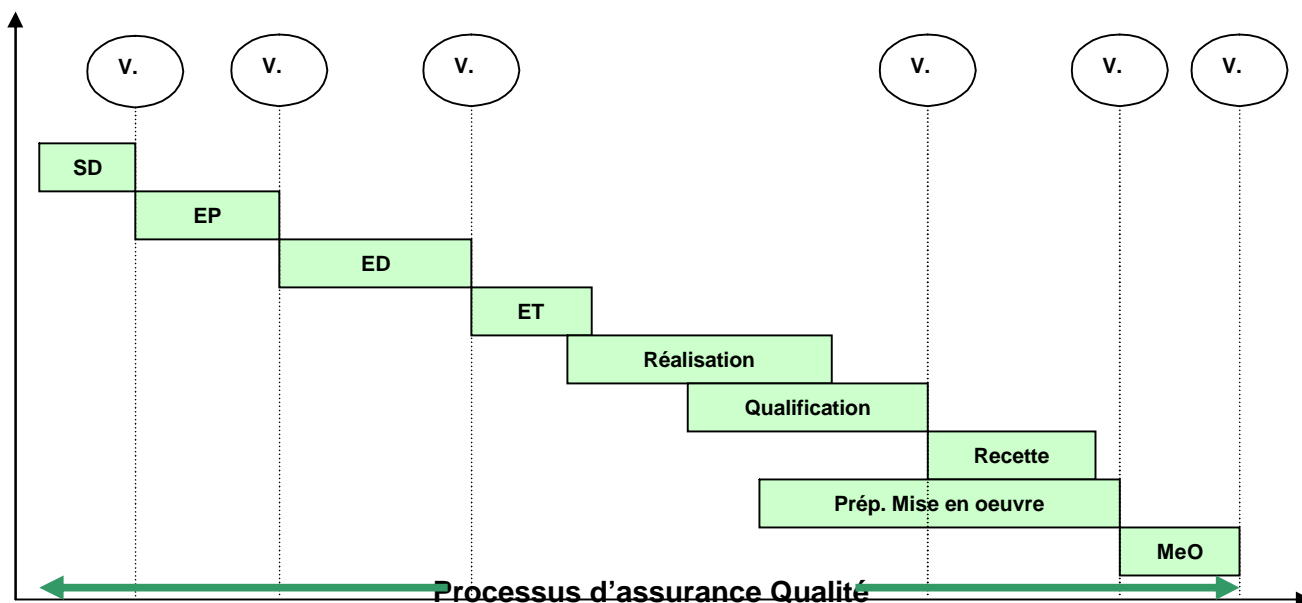
③ Il s'agit d'une démarche standard définie pour un projet complet à mener depuis un SD jusqu'au déploiement, en passant par un développement spécifique.

La démarche doit être adaptée à la taille du projet, le type de prestations (développement spécifique / intégration d'un progiciel), l'organisation (par exemple, MOE de type SSII ou interne, le montage : forfait ou régie, etc, ...). L'adaptation est une des 1ères tâches du CP sur son projet.

④ Chaque étape fait l'objet d'un ou plusieurs livrables, qui doivent être validés conjointement par la maîtrise d'oeuvre et la maîtrise d'ouvrage.

⑤ L'ordre des étapes est théorique : dans un planning, les phases, voire les étapes peuvent se chevaucher.

**Exemple : Planning PERT montrant les étapes qui peuvent se chevaucher :**



**Exemple : description succincte des travaux par étape de la démarche :**

Illustrer la démarche telle que décrite ci-dessus à l'aide d'un exemple : Bouygues Telecom

Etape	Illustration de l'exemple
Schéma Directeur	Bouygues se lance dans les années 80 dans de nvx métiers : la téléphonie, d'où de nvx besoins en Système d'Information. Elaboration d'un plan d'évolution du SI sur 5 ans. Définition d'un SI après une fusion (Air France / Air Inter).
Etude Préalable	Lancement du projet « Facturation » des services de téléphonie mobile aux client. Besoin d'une application Intranet et Internet. Elaboration de plusieurs scénarii fonctionnels & techniques. Choix en fin d'EP de lancer un projet de développement Web.
Etude Détaillée	Elaboration des spécifications (emploi d'UML, Merise ou d'une autre méthode telle Merise ou Remora). Elaboration d'une maquette, voire d'un prototype.

Etude Technique	Spécification des composants logiciels à développer dans l'architecture définie par les architectes, sur la base des spécifications réalisées en ED.
Dév. & tests unitaires	Développement & tests unitaires (chaque développeur teste les composants qu'il développe)
Qualif. & intégration	Assemblage et tests fonctionnels de « bout en bout » de l'application. Tests techniques (montée en charge)
Recette	Tests par le client de l'application qu'il réceptionne.
Pré. de la mise en œuvre	Formation des utilisateurs Migration des données à blanc
Mise en œuvre	Mise en production de l'application qui devient accessible par les utilisateurs + arrêt de l'ancienne application.

## 1.4 Différents types de projets

La démarche de conduite de projet présentée dans ce document est « dimensionnée » pour des projets de développement spécifique.

Les projets différents (exemple : projets d'intégration de progiciels, ERP, ...) nécessitent une autre démarche de conduite de projet, qui n'est pas décrit dans ce document.

### Principales caractéristiques d'un projet :

**La démarche doit être adaptée aux caractéristiques suivantes :**

#### 1) Type

- projet de développement
- projet d'intégration de progiciel ou ERP
- projet de migration : fonctionnelle [migration de données] et / ou technique [chgt de version d'Oracle de serveur J2EE, ...]

#### 2) Architecture

- mainframe
- client / serveur
- client léger (Web)

#### 3) Taille du projet

- Grand projet (Ex : > 5000 jrs\*h)
- Projet moyen (Ex : compris entre 1500 et 5000 jrs\*h)
- Petit projet (Ex : < 1500 jrs\*h)

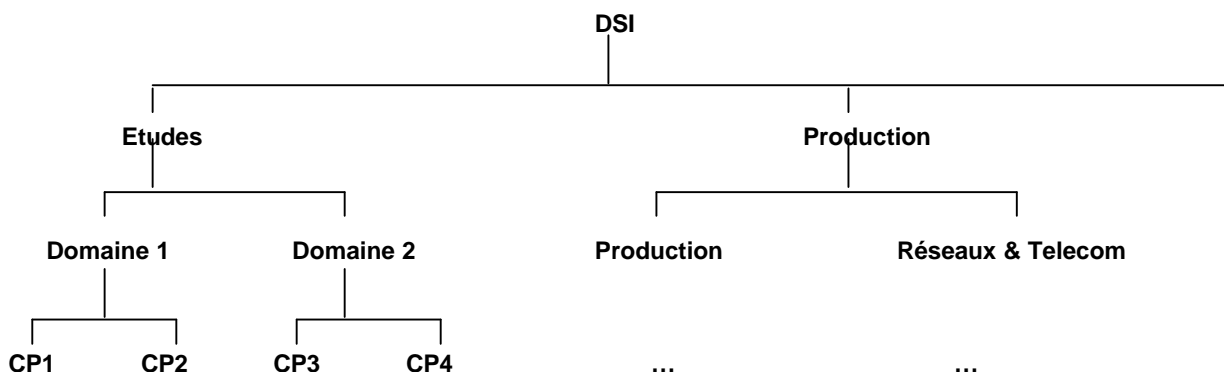
#### 4) Fonctionnel

- application de gestion, compta, RH
- Décisionnel, infocentre, datawarehouse, ...
- industriel (informatique embarqué)
- etc ...

La définition d'une démarche adaptée au contexte, objectifs, ...du projet est une tâche du Chef de projet ou Directeur de projet MOE.

## 1.5 Les acteurs d'un projet informatique

### 1.5.1 L'organisation d'une Direction des Systèmes d'Information (DSI) d'une entreprise



### 1.5.2 Les acteurs de la maîtrise d'œuvre

Maîtrise d'œuvre (MOE principale / déléguée) :

- Cpi,
- Concepteurs,
- Développeurs,
- Experts (réseaux, architectures, produits tels Oracle ou WebLogic, ....), ...

### 1.5.3 Les acteurs de la maîtrise d'ouvrage

Maîtrise d'ouvrage :

- Direction utilisatrice d'une société (direction commerciale, DRH, DAF, ...),
- CPU,
- Utilisateurs finaux, ...

### 1.5.4 Implication MOE & MOA dans un projet

**Les poids des responsabilités MOE & MOA dans les étapes de la démarche de Conduite de projet :**

	<u>Poids MOE</u>	<u>Poids MOA</u>
→ Schéma directeur . . . . .	***	***
→ Etude Préalable . . . . .	***	***
→ Etude Détaillée . . . . .	***	***
→ Etude Technique . . . . .	***	
→ Réalisation et tests unitaires . . . . .	***	
→ Intégration / Qualification . . . . .	***	
→ Recette : recette MOE + recette MOA + bench (si nécessaire) . . . . .	*	***
→ Préparation de la mise en œuvre . . . . .	***	***
→ Mise en œuvre ( déploiement, ...) . . . . .	***	
→ Garantie . . . . .	***	
→ TMA : tierce Maintenance Applicative . . . . .	***	**

## 1.5.5 Les instances de pilotage

Instances de décision et d'arbitrage :

- Comité de Pilotage
- Comité de Projet / Cellule de Suivi / Point hebdomadaire

Préciser :

- leur rôles,
- les fréquences de réunions,
- les acteurs MOE et MOA impliqués,
- l'importance de CR de réunions.

## 1.6 Vocabulaire (à expliquer)

### 1.6.1 Vocabulaire

Expliciter le vocabulaire suivant :

- Système d'Information (définies par les SFG),
- Système informatique (définies par les SFD),
- Architecture physique / logicielle,
- conduite de projet,
- maîtrise d'œuvre (principale / déléguée),
- maîtrise d'ouvrage / d'oeuvre
- assistance maîtrise d'ouvrage / d'oeuvre
- chef de projet utilisateur (CPu),
- chef de projet informatique (CPI),
- instances de pilotage (comité de pilotage & comité de suivi),
- régie (= engagement de moyens),
- forfait (= engagement de résultats),
- co-traitance,
- sous-traitance,
- livrable,
- assurance qualité,
- Plan d'assurance Qualité (PAQ),
- Maquette,
- Prototype.

### 1.6.2 Abréviations

- SD : Schéma Directeur
- EP : Etude Préalable
- ED : Etude Détaillée
- ET : Etude Technique
- PAQ : Plan d'Assurance Qualité
- CP : Chef de projet
- Cpi : Chef de projet Informatique (ou chef de projet MOE)
- Cpu : Chef de projet Utilisateurs (ou chef de projet MOA)
- DP : Directeur de projet

- MOE : Maîtrise d'Oeuvre
- MOA : Maîtrise d'Ouvrage
- SFG : Spécifications Fonctionnelles Générales
- SFD : Spécifications Fonctionnelles Détaillées
- TMA : Tierce Maintenance Applicative

## 2. Les outils

---

Importance des outils dans un projet.

### 2.1 Les outils de l'EP jusqu'à la mise en production

#### 2.1.1 Les différents types d'outils

##### ❶ Les outils bureautiques :

- Word,
- Excel,
- PowerPoint.

##### ❷ Les outils de conception :

- outils de modélisation Merise,
- outil de modélisation UML, ... (ex : Together de Borland ; Rose de Rational Rose, Power-AMC de Sybase, ...)

##### ❸ Les outils de développement :

Les principaux middleware, serveurs Web et serveurs d'applications :

- TUXEDO (architectures UNIX pour C/S),
- WebLogic (architecture J2EE à base d'EJB),
- WebSphere (architecture J2EE à base d'EJB),
- Apache, Tomcat, J-Boss ou Jonaz (open source – serveur Web et serveur d'applications pour une architecture),
- CICS (IBM),
- TDS (Bull).

Les principaux outils de développement :

- J-builder, WebGain pour les développements Java (J2EE),
- Développer 2000,
- PowerBuilder,
- Visual Basic,
- NATSTAR, (en perte de vitesse)
- PROLIFICS, (en perte de vitesse)

Les framework (bcp d'open source) utilisés en développement d'un projet J2EE :

- Struts,
- Spring,
- Hivernate,
- EjbGen,
- Ant, ...

Les principaux outils de développement (éditeurs):

- Eclipse (éditeur pour code Java),

- TextPad (open source),
- WorkShop (outil de BEA à utiliser notamment pour développer des webservice).

Les compilateurs.

Les principaux outils de gestion de configuration :

- CVS (open source) Et/ou Suversion,
- PVCS.

Les principaux outils de gestion des demandes (ou Faits techniques), principalement utilisés en TMA :

- MANTIS (open source),
- Bugzilla (open source)
- etc.

#### 4 Les outils de tests (qualification, recette) :

- Outil de gestion des plans, des scenarii et des cas de tests (Test Director, Mercury)Outils permettant d'automatiser les tests de non-régression.

Les principaux outils de tests :

- QA-Run de Compuware,
- ????? de Mercury.

#### 5 Les outils de bench :

- outils permettant de simuler la montée en charge d'une application (ex : simulation d'un nb d'internautes de + en + nombreux). Utilisation d'injecteurs.

Les principaux outils de bench :

- QA-Load de Compuware.
- ????????? de Mercury.

#### 6 Les outils de pilotage & planification :

- outils de planification et de suivi des charges (ex : MS-Project, PMW)

#### 7 Les outils de qualimétrie (essentiellement utilisés en TMA) :

- Ces produits reconnaissent la syntaxe de certains langages de programmation. Ils peuvent servir à mesurer la qualité du code source, identifier du code mort, faire des analyses d'impacts, ...

Les principaux outils de qualimétrie :

- CAST de Cast SoftWare.
- System Code d'ACFO.

## 2.1.2 Tableau de synthèse Outils / Etapes de la démarche

### Tableau de synthèse Outils / Etapes de la démarche :

	Outils de conception	Outils de développement	Outils de tests	Outils de bench	Outils de pilotage et planification + Outils bureautiques
Schéma Directeur	<b>(x)</b>				<b>x</b>
Etude Préalable	<b>x</b>				<b>x</b>
Etude Détaillée	<b>x</b>				<b>x</b>
Etude Technique	<b>x</b>	<b>x</b>			<b>x</b>
Dév. & tests unitaires		<b>x</b>			<b>x</b>
Qualif. & intégration		<b>x</b>	<b>x</b>	<b>x</b>	<b>x</b>
Recette		<b>x</b>	<b>x</b>	<b>x</b>	<b>x</b>
Pré. de la mise en œuvre			<b>x</b>	<b>x</b>	<b>x</b>
Mise en oeuvre					<b>x</b>

Un CP doit vérifier d'avoir les bons outils au bon moment, avec les ressources humaines maîtrisant ces outils ...

## 2.2 Les outils d'administration du système

Les outils nécessaires, une fois l'application finale mise en exploitation sont les suivants :

- sauvegarde / restauration,
- ordonnancement des batch (exemple : \$UNIVERSE),
- Outil de surveillance système (exemple : PATROL).

### 3. Rappels MERISE & UML (méthodologies de modélisation)

Ce chapitre n'est pas au programme. Il pourra toutefois être réalisé à la demande des étudiants.

#### 3.1 Présentation de la démarche par niveaux

Présentation des 3 niveaux de la méthode MERISE.

Présentation des rôles des niveaux conceptuel, ... physique.

Les modèles MERISE.

	Données	Traitements	Flux
Conceptuel	MCD	MCT, CVO	MCC
Organisationnel	MOD	MOT	MOC
Logique	MLD (+ réparti)	MLT (+ réparti)	
Physique	MPD	MPT	

Rappel de l'historique de MERISE (année 78 à ce jour).

Principe de modélisation :

- séparation données / traitements,
- préoccupations distinctes suivant le cycle d'abstraction.

MERISE = Modèles + démarche de Conduite de projet.

Les évolutions de Merise : Merise/OO, Merise/2, ...

#### 3.2 UML (rappel)

##### 3.2.1 Le diagrammes des cas d'utilisation (Use case)

Présentation des concepts du diagramme UML.

##### 3.2.2 Le diagrammes des classes

Présentation des concepts du diagramme UML.

##### 3.2.3 Le diagrammes d'états - transitions

Présentation des concepts du diagramme UML.

##### 3.2.4 Le diagrammes de séquences

Présentation des concepts du diagramme UML.

### **3.3 MERISE (rappel)**

#### **3.3.1 Le modèle des flux (rappel)**

Présentation des concepts du MCC :

- ◆ domaine,
- ◆ acteur,
- ◆ flux d'information,
- ◆ propriété.

Présentation du cas « Compagnie d'assurance IARD – traitement d'un sinistre auto ».

#### **3.3.2 MCD (rappel)**

Présentation des concepts du MCD :

- ◆ entité,
- ◆ relation,
- ◆ cardinalités,
- ◆ propriété.

Exercice : cas Hotel ou autre.

#### **3.3.3 MLD relationnel (rappel)**

Rappel de la théorie.

Présentation de règles de migration d'un MCD en un MLD/Rel.

Exercice : cas Hotel ou autre.

#### **3.3.4 MCT (rappel)**

Rappel de la théorie.

Présentation du cas « Compagnie d'assurance IARD – traitement d'un sinistre auto ».

#### **3.3.5 MOT (rappel)**

Rappel de la théorie.

Présentation du cas « Compagnie d'assurance IARD – traitement d'un sinistre auto ».

## 4. Schéma Directeur

### 4.1 Les 2 niveaux de SD

2 types de SD : le schéma Directeur stratégique et le schéma Directeur opérationnel du SI.

Schéma Directeur stratégique



Schéma Directeur des SI

### 4.2 SD stratégique

#### 4.2.1 Définition

**Schéma Directeur Stratégique :** définition de la stratégie de l'entreprise à 5 ans, voire jusqu'à 10 ans. Très important pour les entreprises pour les métiers évoluent, ou suite à une fusion, ...

#### Exemples de cas ou des SD stratégiques ont été réalisés :

- fusion Air France + Air Inter,
- fusion Carrefour + Promodès.
- Bouygues : évolution du BTP vers les télécoms,
- Vivendi Universal : évolution du métier du distributeur d'eau (ex-CGE) aux métiers du multimédia (CEGETEL/SFR, ...).

Dans ce 1<sup>er</sup> type de SD, on traite tous les impacts.

#### Exemple des aspects à traiter dans un SD stratégique :

Dans la fusion Air France + Air Inter, on élabore des macros plan d'actions concernant :

- fusion la fusion des divisions de 2 sociétés (fusion des 2 directions financières, des 2 directions des RH, des 2 directions du personnel navigant, ...). A la clé, prévision de plan d'embauche et/ou licenciement,
- délocalisation de certaines structures. par exemple, le siège complet à Charles de Gaulle, réduction des effectifs basés à Orly,
- homogénéisation du parc d'avions (par exemple, prévoir sur un horizon de qqes années d'homogénéiser le parc des avions : tout Airbus ou tout Boeing, afin de réduire les coûts de maintenance, les coûts de formation des pilotes, etc...)
- restructuration des lignes desservies : prévision d'élimination des lignes redondantes + créations de nouvelles lignes à desservir,
- etc...

#### 4.2.2 Les acteurs

- Directions Générales,
- Sociétés de Conseil ou Consultants « Top manager » (ex : Alain MINC, ex-PDG, ...).

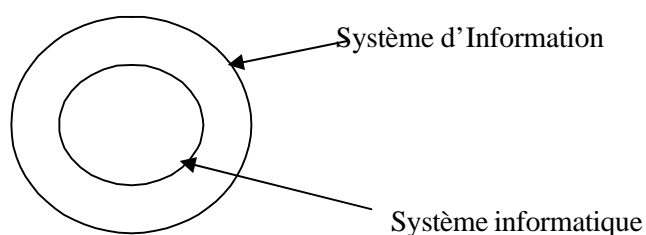
## 4.3 SD du SI

### 4.3.1 Définition

Mise en œuvre du SD stratégique de l'entreprise sur le SI ;

- ◆ Planification des projets à lancer à un horizon de 2/3 ans ;
- ◆ Définition des moyens nécessaires à la réalisation :
  - périmètre des projets,
  - architecture physique du SI,
  - budgets,
  - ressources,
  - macro-planning, ...

**Rq** : Définir les 2 termes : Système d'Information et Système Informatique



### Exemple des aspects à traiter dans un SD du SI :

Dans l'évolution de Bouygues vers la téléphonie mobile, il faut identifier :

- les projets informatiques à lancer avec une 1<sup>er</sup> ébauche de leur contenu fonctionnel et de leur architecture respectifs + définir un macro planning de ces projets sur les 2/3 ans à venir,
- les budgets de la DSI nécessaires pour réaliser les projets sur les 2/3 ans à venir,
- envisager des embauches au sein de la DSI,
- etc...

### 4.3.2 Les acteurs

- ◆ DSI
- ◆ Préciser le rôle des cabinets de conseil (Price WaterHouse, Ernst & Young, Andersen Consulting, Orgaconseil, ...).

## 5. Etude Préalable

### 5.1 Objectifs de l'étape

Objectifs de l'étape pour la MOE :

- ◆ comprendre les besoins de la MOA et élaborer une de plusieurs solutions fonctionnelles et techniques chiffrées (coûts de mise en œuvre + coûts de fonctionnement jusqu'à 5 ans, ...). Par exemple :
  - développement spécifiques / Intégration d'un progiciel ou d'un ERP (PeopleSoft, Oracle Applications, SAP R/3, Oneworld de JD Edwards, ....) ;
  - architecture X-Net / architecture C-S, ...
- ◆ choix obligatoire d'une solution unique pour poursuivre vers l'Etude Détaillée, parmi les plusieurs solutions proposées. Chacune des solutions proposées aux instances de pilotage pour décision doit être accompagnée du chiffrage des charges, des coûts, d'un planning et d'un recensement des avantages & inconvénients.

### 5.2 Les phases de l'EP

Les phases de l'EP :

- Lancement (Ph1),
- Bilan de l'existant (Ph2),
- Définition des orientations (Ph3),
- Modélisation du système cible / SER (Ph4),
- Validation et décision (Ph5).

#### 5.2.1 Lancement de l'EP (Ph1)

##### ① Mise en place de l'organisation du projet :

- Organisation MOE + organisation MOA + instances de pilotage du projet :
  - Maîtrise d'œuvre - Maîtrise d'ouvrage,
  - Assistance Maîtrise d'œuvre (si nécessaire) - Assistance maîtrise d'ouvrage (si nécessaire),
  - Chef de projet informatique (CPI) - Chef de projet utilisateur (CPU),
  - Les profils de la MOE les + courants pour une EP (sous la responsabilité du CPI) :
    - Concepteurs,
    - Architectes,
    - Experts possibles : administration des données, expert technique, ergonomes (si maquettage à faire en EP), responsable Qualité (éventuellement pour le PAQ), ...
  - Mise en place des instances de pilotage du projet (les + courantes)
    - Comité de projet,
    - Comité pilotage,
- Charges / Planning / Coûts :
  - Charges et planning précis pour l'EP
  - Charges et planning estimés pour les étapes suivantes : à actualiser au fur et à mesure de l'avancement du projet.

Planning des instances de pilotage à définir

Planning réunions avec les utilisateurs à définir

## Remarque :

Dans le planning, indiquer la durée de chaque tâche à réaliser, les dates prévisionnelles de livraison des livrables, les dates des instances de pilotage du projet, les dates prévisionnelles des réunions de travail avec la MOA

### ② **Choix des méthodes de conception :**

- pour les architectures logicielles non objet (mainframe ou C/S), privilégier Merise,
- pour les architectures logicielles objet (Web de type J2EE), privilégier :
  - Merise « données » pour la modélisation du MCD au MPD,
  - UML pour la modélisation de la partie dynamique (diagr. de classes, de cas d'utilisation, d'états-transitions, de séquences),

### ③ **Mise en place de la plate-forme de conception :**

- un environnement bureautique complet (OFFICE),
- outil de suivi de projet (PSN6, MS-Project, PMW),
- outil de conception (MEGA, POWER\*AMC),
- outil de maquettage (a priori l'outil de développement).

### ④ **Rédaction et validation du PAQ par les 2 parties :**

Le PAQ doit être validé immédiatement (c-à-d dès la fin de la phase de Lancement) par la MOE et la MOA → document contractuel entre les 2 parties (il engage les 2 parties).

## 5.2.2 Bilan de l'existant (Ph2)

Modélisation limitée aux SFG par rétro-ingénierie de l'application existante limitée le + souvent à un SER → Bilan fonctionnel et technique : a minima modélisation des données du système existant, afin de construire un nouveau modèle des données dans lequel on sera migrer les données de l'application existante.

Expliquer SFG et SFD.

Préciser comment définir un Sous Ensemble Représentatif (SER).

**Rq** : Expliquer la différence entre SFG et SFD.

## 5.2.3 Définition des orientations (Ph3)

Suivant la taille du projet, les enjeux, les risques, ... possibilité de prévoir plusieurs scénarios pour la cible :

Scénario(s) fonctionnel(s) + scénario(s) technique(s) classés du « plus complet » (donc le + cher) au « plus sobre » (donc le – cher). En règle générale, la MOE propose 3 scénarios max.

Chacun des scénarios va être spécifié sur un SER, les avantages & inconvénients vont être identifiés, les charges & coûts de chacun seront mesurés, ...

Exemple, pour un projet de GRH :

- Possibilité de revamper (ré-habillage) d'une « vieille » application mainframe, pour la rendre accessible via un intranet,

- Possibilité de mettre en œuvre un progiciel. Après étude du marché, on retiendra 2 ou 3 progiciels cible à étudier,
- Possibilité de ré-écrire une application « from scratch ».

## 5.2.4 Modélisation du système cible / SER (Ph4)

### 5.2.4.1 Modélisation du fonctionnel

Sur les projets « Nouvelles Technologies » (NTIC), privilégier l'emploi d'UML pour la modélisation des traitements du système cible, à la méthode MERISE. Toutefois, il est recommandé de modéliser le MCD/MLD en plus du diagramme de classes UML, pour les raisons suivantes :

- MCD/MLD : Modélisation du système de stockage des données (modèle statique).
- diagramme de classes UML : Modèles métiers, servant à la définition des classes métiers d'une architecture applicative.

Modélisation du système cible, éventuellement sur un SER.

- Modélisation avec une méthode (Merise, UML). Possibilité de mixer les 2 pour les projets X-Net.
- Production des SFG (Vision Système d'Information) : à élaborer avec Merise et/ou UML.

**Rq** : Faire des rappels MERISE / UML.

La modélisation peut se faire sur ou plusieurs scénarios identifiés en Ph3.

#### ❶ Cas d'une recherche d'un progiciel ou ERP :

1. Analyse de l'offre des progiciels du marché. Sélection de 2 à 5 progiciels max. à étudier.
2. Analyse des fonctionnalités du progiciel pour déterminer le taux de couverture du progiciel des besoins exprimés par la MOA.
3. Identification des fonctionnalités qui devront aux côtés du progiciel.

### 5.2.4.2 Initialisation de l'architecture technique cible

#### ❶ Types d'architectures physiques :

**Architecture physique** : ensemble des moyens comprenant les serveurs, les PC, logiciels de base (OS, SGBD, serveurs d'applications), les réseaux WAN et LAN nécessaires au fonctionnement de l'application cible

Les types d'architectures physiques possibles :

- architecture mainframe avec PC en émulation (mode caractère),
- architecture client – serveur (client "lourd"),
- architecture Web : Intranet / Extranet / Intranet (client "léger").

Initialiser l'architecture physique cible :

- OS du serveur (Ex : UNIX ou Linux ou NT),
- SGBD (Ex : Oracle, SQL-Server, ou MySQL)
- Serveur d'application (Ex si J2EE : WebLogic, WebSphere ou JBOSS, ...)

Pour un projet avec une architecture J2EE, on peut proposer plusieurs variantes :

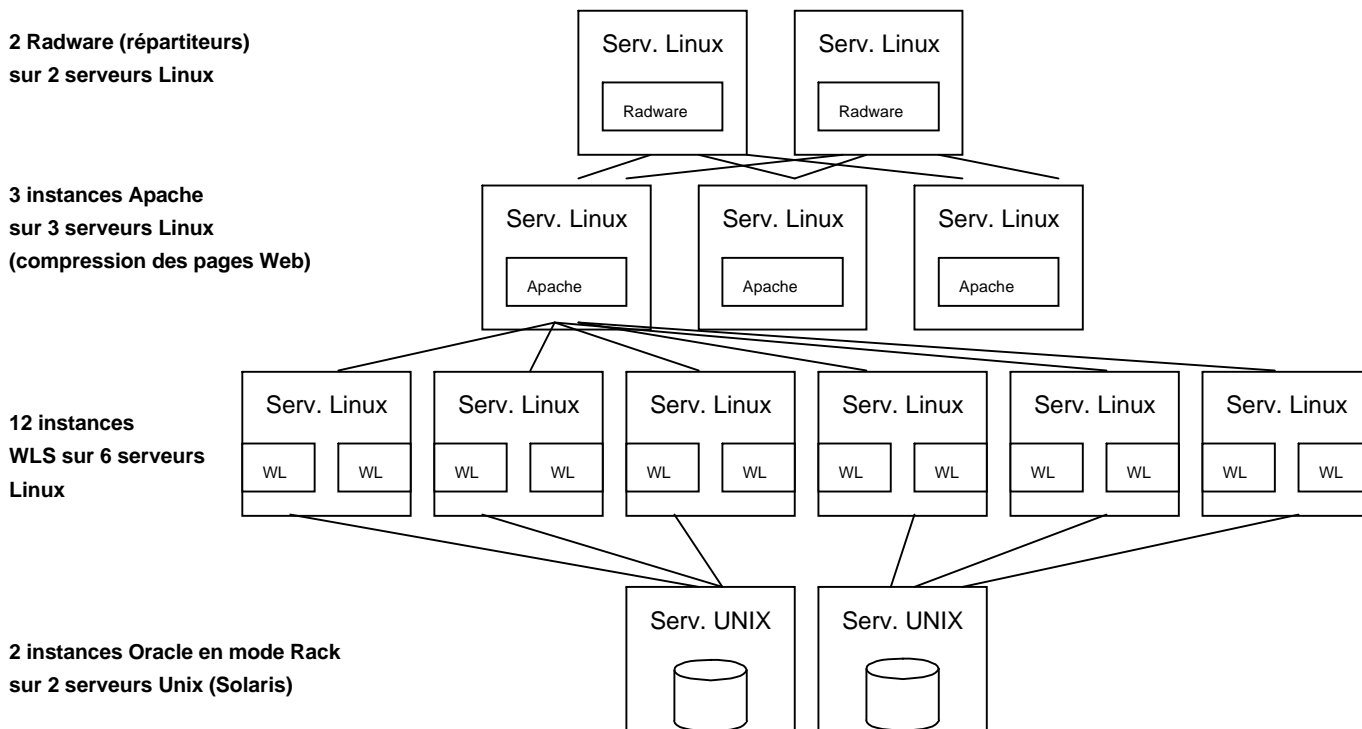
- WebLogic + Oracle,

➤ ou Jboss (voire Jonaz) + Oracle,

➔ Utilisation d'une architecture open source

Exemple :

Schéma de l'architecture physique de [www.anpe.fr](http://www.anpe.fr)



Exemple :

Télé-IR : + 200 serveurs physiques dans la période saisie de l'IR.

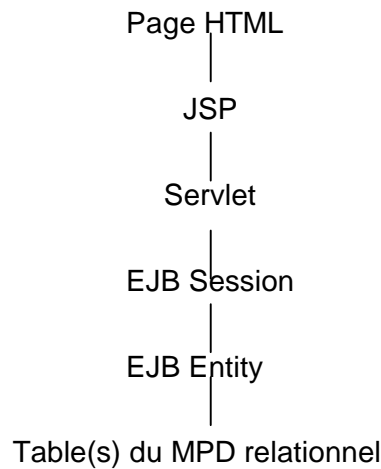
### 🕒 Types d'architectures logicielles :

**Architecture logicielle** : Structure interne d'une application, 'c-à-d l'ensemble des composants logiciels à développer.

→ Exemple : Pour une architecture logicielle J2EE, c'est l'ensemble des pages HTML, JSP, servlets, EJB, Shell, procédures stockées, ... composant l'application. Les composants à développer doivent respecter la norme J2EE en vigueur (en , J2EE v1.1).

Exemple :

Schéma d'une architecture logicielle du type J2EE (norme v2.0 , modèle MVC, ...) :



Définir :

- Les avantages / inconvénients de chaque scénario,
- Les charges / délais (planning) de chaque scénario,
- Les coûts de chaque scénario (préciser les coûts d'investissement et les coûts de fonctionnement (par exemple, sur 5 ans)

On pourra utiliser une grille avec différents critères à noter, en s'appuyant sur des coefficients pour calculer une note finale et donc justifier la préférence de la MOE / ou de la MOA.

Pour les développement en architecture J2EE, on pourra utiliser des **open source** (de + en + nombreux) :

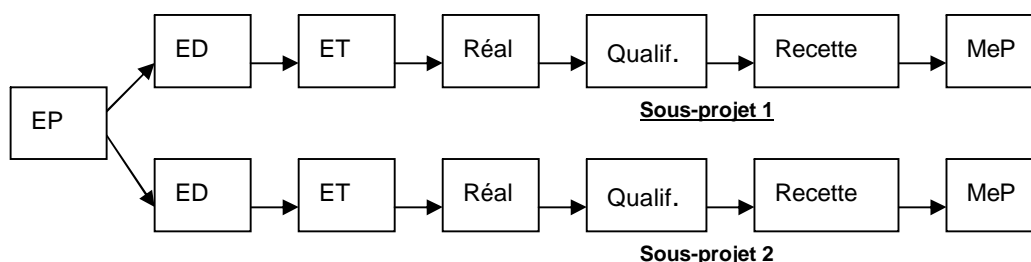
- CVS, subversion pour la gestion de config.,
- ant ou spring pour la compilation (\*.jar, \*.war, \*.ear),
- ejbgen pour la génération dses squelettes des EJB,
- Struts, pour le dév. de la couche présentation (framework), ...
- Hibernate, ...

### 5.2.5 Processus de validation et décision (Ph5)

#### ❶ Finalisation de la production des livrables de l'EP :

Finaliser la rédaction des documents, notamment en complétant les parties suivantes :

- Découpage (si nécessaire) en sous-projets pour les étapes à suivre ; découpage conseillé pour les grands projets (> 2000 j\*h).



- Actualisation des avantages et inconvénients de chacun des scénarios proposés.
- Actualisation des charges de la MOE et des délais (planning) pour poursuivre le projet jusqu'à sa fin.

### ② **Processus de validation et de décision :**

Processus de choix d'une solution en fin d'étape: comité de projet + comité de pilotage (voire du comité de direction de l'entreprise). 3 décisions possibles :

- poursuite sans réserve,
- restriction du périmètre,
- arrêt du projet (en général si impossibilité avec le budget prévu).

Pour le CPi chargé de cette présentation, il est important d'être synthétique, de montrer les avantages et inconvénients de chaque solution, de présenter les budgets prévisionnels associés à chacune des solutions et (éventuellement) de présenter sa préférence.

Pour un Comité de Pilotage, on prévoira des slides PowerPoint, en accompagnement des livrables.

Après que le Comité de Pilotage ait validé la solution, on peut lancer, avec son accord l'ED (ou les ED en cas de lotissement en plusieurs sous-projets).

## 5.3 Livrables de l'EP

Les livrables de l'Etude Préalable :

- Note de lancement de l'EP (facultatif) → (issue de Ph1),
- Plan d'Assurance Qualité (PAQ) → (issue de Ph1).
- Dossier de fin d'Etude Préalable (aspects fonctionnels) (\*)
- Dossier d'architecture technique :
  - Architecture physique (config. des machines, réseaux, ...)
  - Architecture logicielle (organisation des composants logiciels par rapport à un schéma type).

(\*) Le dossier de fin d'Etude Préalable contient a minima (en général sur un SER) les modèles MERISE ou UML suivants :

- un MCD,
- un diagramme des cas d'utilisation (ou Use case),
- un diagramme des classes,

Ces diagrammes UML et modèles MERISE ne sont pas finalisés. A titre d'exemple, le MCD contient 90%-95% des entités et associations et (par exemple) 50% des attributs des entités et associations.

## 5.4 Importance du processus Qualité

Le PAQ est un contrat qualité entre le maître d'oeuvre et le maître d'ouvrage :

- démarche de conduite de projet (avec ses livrables) adaptée à la problématique du projet,
- définition de l'organisation du projet : MOE, MOA et instances de pilotage
- description du périmètre du projet,
- description des ressources (effectifs, charges, plannings, outils, ...),

Les SSII se font certifier par l'AFAQ (ISO9001).

**Normes** : ISO9001, CMMI, ITIL.

## 5.5 Les acteurs de l'EP

### ❶ Les acteurs d'un projet en étape d'EP :

- maîtrise d'oeuvre (voir les acteurs) + assistance MOE,
- maîtrise d'ouvrage (voir les acteurs) + assistance MOA.

Expliquer les différents profils des acteurs MOE & MOA pour une Etude Préalable.

Préciser les profils des acteurs de la MOE et de la MOA pour une EP. Pour la MOE, on citera plus en détails :

- le rôle du **concepteur** : bonne connaissance des méthodes de modélisation, capable d'être « force de proposition » auprès de la MOA, ...
- le rôle de l'**architecte** : élaboration de l'architecture logicielle et de l'architecture physique, élaboration des N&S de développement, garant de l'utilisation des N&S, ...
- le rôle de l'**administrateur des données** : garant de la cohérence des données, optimisation des modèles conceptuels et logiques), consolidation des modèles multi-projets, ...
- le rôle de l'**ergonome** (assimilée MOE ou MOA),

- **le rôle de la MOE ayant une bonne connaissance de l'ancien système informatique** (celui qui sera arrêté à l'issue de la mise en production de la nouvelle application) : rétro-documentation, préparation à l'élaboration des SFG/SFD du processus de migration des données, ...

② **Les instances de pilotage en étape d'EP :**

- réunions mensuelles pour le comité de pilotage,
- réunions hebdomadaires pour le comité de projet.

Préciser le rôle, les objectifs, les participants aux instances de pilotage.

## 6. Etude Détaillée

Introduire le vocabulaire suivant :

- ✦ SFG,
- ✦ SFD,
- ✦ maquettage,
- ✦ prototypage.

### 6.1 Objectifs de l'ED

Dans l'hypothèse d'une solution cible à développer en spécifique, les objectifs de l'étape pour la MOE sont les suivants :

- ✦ Compléter les SFG initialisées en EP & élaborer les SFD avec la collaboration de la MOA sur la base de la solution validée en fin d'EP,
- ✦ Préparer le lancement de la conception technique (passage d'une préoccupation « Système d'information » à « Système informatique »).
- ✦ Faire valider tous les livrables de l'ED avant de passer à l'étape suivante. La solution spécifiée en ED doit être accompagnée du chiffrage des charges, des coûts, d'un planning pour être présentée aux instances de pilotage.

### 6.2 Les phases de l'ED

Les phases de l'ED sont :

- Lancement (Ph1)
- Compléments des SFG (Ph2)
- Elaboration des SFD Données + Traitements (Ph3),
- Définition de l'architecture technique (Ph4),
- Validation et décision (Ph5).

#### 6.2.1 Lancement (Ph1)

##### ❶ Adaptation de l'organisation du projet aux préoccupations de l'ED :

- Organisation MOE + organisation MOA + instances de pilotage du projet :
  - Maîtrise d'œuvre
  - Assistance Maîtrise d'œuvre (si nécessaire)
  - Chef de projet informatique (CPI)
  - Equipe MOE (concepteurs, architecte, ...)
  - Maîtrise d'ouvrage,
  - Assistance maîtrise d'ouvrage (si nécessaire),
  - Chef de projet utilisateur (CPU),
  - Utilisateurs

Les profils de la MOE les + courants pour une ED (sous la responsabilité du CPi) :

- Concepteurs (avec des compétences en maquettage et/ou prototypage),
  - Architectes,
  - Experts possibles : administration des données, expert technique, ergonomiste (si nécessaire), responsable Qualité (éventuellement pour la MàJ du PAQ), ...
  - Mise en place des instances de pilotage du projet (les + courantes)
    - Comité de projet,
    - Comité pilotage,
- Charges / Planning / Coûts

Planning des instances à définir

- Charges et planning précis pour l'ED
- Charges et planning estimés pour les étapes suivantes : à actualiser au fur et à mesure de l'avancement du projet.

#### ② **Mise à niveau (si nécessaire) de la plate-forme de conception :**

- par exemple, outil de conception Merise &/ou UML,
- par exemple, installation d'outil de maquettage (a priori compatible avec l'outil de développement).

#### ④ **Actualisation du PAQ et validation par les 2 parties :**

Dans la plupart des cas, il s'agit de la mise à jour du PAQ initialisé en début d'EP.

Importance du rôle d'un PAQ à faire dans l'étape de lancement.

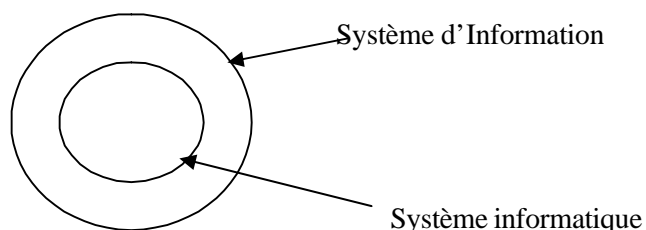
Le PAQ devra décrire et valider les procédures visant à garantir la qualité du processus.

- l'organisation de la MOE, de la MOA, des instances de pilotage : fréquence des réunions, rédacteur de l'OJ, rédacteur du CR, ...
- les livrables à produire,
- les ressources nécessaires à la bonne réalisation de l'étape (serveur, postes, licences, ...),
- le planning (mettre en évidence le chemin critique, expliciter les tâches de la MOE, de la MOA, ...)
- les modalités de validation d'un livrable,
- les modalités de livraison d'une nouvelle version, etc...

Le PAQ doit être validé immédiatement (c-à-d dès la fin de la phase de Lancement) par la MOE et la MOA → document contractuel entre les 2 parties (il engage les 2 parties).

### 6.2.2 Complément des SFG (Ph2)

Rappel : SFG → Vision « Système d'Information ».



Réaliser les compléments des SFG ébauchées en EP, **c-à-d sur 100% du domaine étudié**, et non plus par rapport au SER.

Les SFG finales décriront la cible à atteindre. Cette cible sera décrite avec :

- les diagrammes UML : Diagr. des cas d'utilisation, diagr. de classes,
- les modèles MERISE : MCD, etc...

### 6.2.3 Elaboration des SFD Données + Traitements (Ph3)

Rappel : SFD → Vision « Système informatique ».

### 6.2.3.1 *Elaboration de N&S (Ph3.1)*

Elaboration des normes et standards d'un projet (elles sont élaborées par la MOE, mais elles peuvent aussi être fournies par la MOA) :

- standards ergonomiques (charte graphique, feuilles de style, ...),
- standards de développement (règles de nommage des données, des fichiers, ... règles de développement, ...)
- standards d'utilisation du SGBD.

### 6.2.3.2 *Elaboration des SFD (Ph3.2)*

Elaboration des SFD :

- des données : MCD complet + MPD.
- des traitements de la cible :
  - traitements transactionnels (TP) :  
(Nécessité d'une charte d'ergonomie, charte graphique, feuille(s) de style)
    - ❖ application TP Cible → Maquette ou prototype à développer,
  - traitements batch :
    - ❖ des traitements d'interfaces
    - ❖ des traitements de migration des données (programmes de déchargement et de rechargement),
  - de manière facultative :
    - ❖ des systèmes transitoires (expliquer le fonctionnement d'un lotissement),  
application TP / système transitoire → Maquette ou prototype à développer,
    - ❖ des processus dégradés.  
application TP / système dégradé → Maquette ou prototype à développer,

Pour l'ensemble des SFD, on peut modéliser les processus à l'aide des méthodes Merises ou UML.

### **Maquettage / prototypage :**

Importance :

- ♣ du **maquettage** (si possible avec l'outil de développement, afin de ré-utiliser). Objectif du maquettage : valider l'IHM et l'ergonomie par les utilisaux finaux.
- ♣ du **prototypage**. Objectif : valider certains aspects techniques du développement (ex. développement en JAVA sur LINUX, afin de vérifier la faisabilité et les performances futures)

**Dans la mesure du possible, maquetter 100% des écrans de l'application cible.**

### **Système dégradés :**

3 types de systèmes dégradés (objectif : pallier à un dysfonctionnement temporaire du système informatique « normal », tel qu'une panne d'un serveur, une panne du LAN ou du WAN, ...

- ❖ Fonctionnement manuel (sans ressources informatique)
- ❖ Hardware : redondance des équipements informatiques (matériels et réseaux),  
Indispensable pour une architecture sécurisée (disponibilité = 99,9% du temps)
- ❖ Software : traitements de reprise nécessitant des développements TP & batch.

Exemples de conception de **systèmes dégradés** :

Solutions hardware (la majorité des cas) :

- redondance des équipements informatiques (matériels et réseaux) qui permettent de maintenir une activité permanente d'une application en production, y compris lorsque une partie des équipements sont en panne ou arrêtés pour raisons de maintenance technique ou de déploiement d'une nouvelle version logicielle :
  - Redondance des instances WebLogic, des instances Apache dans une application J2EE,
  - Utilisation de l'option Oracle Parallel Server (OPS) d'Oracle.
- Sites de backup (exemple : rappeler l'incendie du siège du Crédit Lyonnais, et sa faculté à redémarrer les applications en quelques heures) → sites « en doublon » dont le lieu est tenu secret.

Solutions software (la minorité des cas) :

- processus dégradés (salle de marché dans une banque, contrôle aérien, salle de contrôle d'une centrale nucléaire) : solutions TP et/ou batch à imaginer.

### **Système transitoires** :

Définition d'un lotissement + impacts (charges, contraintes pour les utilisateurs, ...).

Exemple de **système transitoires** :

Dans le cadre d'un lotissement du système de paie de tous les fonctionnaires du Ministère de l'Education Nationale, il faut définir le système de paie dans l'ancien ou le nouveau système, selon si la catégorie des fonctionnaires à payer fait partie du lot à développer (ou pas).

## **6.2.4 Définition de l'architecture physique (Ph4)**

Objectif de cette phase :

- définir les moyens nécessaires à l'architecture technique (architecture physique + architecture réseaux) et aux environnements de développement,
- définir les moyens nécessaires à l'architecture technique et aux environnements de tests MOE, de recette MOA, de bench ; voire également de formation
- définir les moyens nécessaires à l'architecture technique et à l'environnement de déploiement.

Définir pour chacun des serveurs utilisés en dev, recette, bench, production ... :

- La RAM,
- l'espace disque utile,
- les licences SGBD, serveurs d'applications,
- les machines sur lesquelles il faut installer des compilateurs, un outil de gestion de conf., ...
- des licences de bench, etc...

Les différents types de clients - serveurs : faire schéma du Gartner Group :

- architecture mainframe,
- revamping,
- client-serveur (client « lourd »),
- architecture Web (client « léger »),
- architecture centralisée / distribuée,
- etc ...

Choix de l'architecture technique cible :

- Réseau Wan et LAN. Par exemple, IP indispensable pour application X-Net,
- serveur UNIX ou NT + définition de sa configuration (RAM, CPU, espaces disques, RAID1/5, machine de backup, ...)
- client - serveur (2 ou 3 tier) ou X-Net (INTRANET, INTERNET),
- PC « léger » ou lourd »,
- SGBD : ORACLE, INFORMIX, SYBASE, ...
- choix des outils de conception / développement / tests / etc...
- sauvegarde / restauration,
- ordonnancement des batch,

Sécurité de l'architecture dans l'environnement de production → redondance des serveurs, des LS, utilisation du système RACK (oracle), ...

### 6.2.5 Validation et décision (Ph5)

**Importance de la validation du dossier de fin d'Etude Détaillée = contrat entre la maîtrise d'œuvre (MOE) et la maîtrise d'ouvrage (MOA).**

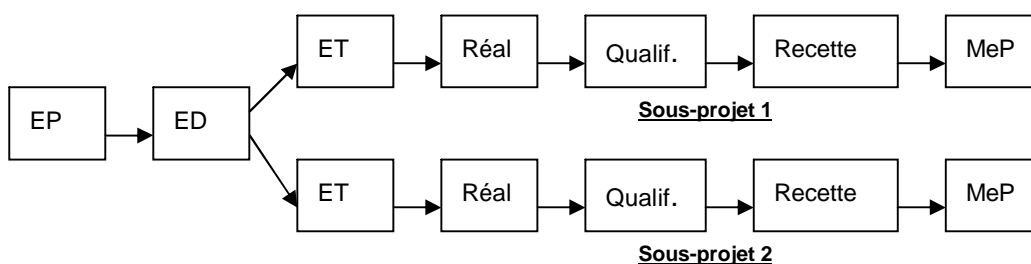
Pour le CPi chargé de cette présentation, il est important d'être synthétique, de montrer la maîtrise technique et fonctionnelle du projet.

Pour un Comité de Pilotage, on prévoira des slides PowerPoint.

Plusieurs décisions possibles :

- Validation des SFD réalisées, de l'architecture décrite ; ainsi que l'enveloppe des charges et le planning proposé ;
- Demande de révision des SFD, par exemple si dépassement du budget prévu...

**En cas de projet trop gros, possibilité de lotissement des travaux jusqu'à la mise en production incluse :**



Après que le Comité de Pilotage ait validé la solution, on peut lancer, avec son accord l'ET.

## 6.3 Les acteurs de l'ED

### ❶ Les acteurs d'un projet en étape d'ED :

- maîtrise d'œuvre (voir les acteurs) + assistance MOE,
- maîtrise d'ouvrage (voir les acteurs) + assistance MOA.

Préciser les profils des acteurs de la MOE et de la MOA pour une ED. Pour la MOE, on citera plus en détails :

- le rôle du **concepteur** : bonne connaissance des méthodes de modélisation, capable d'être « force de proposition » auprès de la MOA, ...
- le rôle de l'**architecte** : élaboration de l'architecture logicielle et de l'architecture physique, élaboration des N&S de développement, garant de l'utilisation des N&S, ...
- le rôle de l'**administrateur des données** : garant de la cohérence des données, optimisation des modèles conceptuels et logiques), consolidation des modèles multi-projets, ...
- le rôle de l'**ergonome** (assimilée MOE ou MOA),
- le rôle du **maquetteur** (ou développeur HTML pour une maquette Web).
- le rôle de quelques **experts** : expert SGBD, expert en architecture, ...

#### ② Les instances de pilotage en étape d'ED :

- réunions mensuelles pour le comité de pilotage,
- réunions hebdomadaires pour le comité de projet.

Préciser le rôle, les objectifs, les participants aux instances de pilotage.

## 6.4 Livrables de l'ED

Les livrables de l'Etude Détaillée :

- Note de lancement de l'ED (facultatif),
- Plan d'Assurance Qualité (si mis à jour)
- Dossier de fin d'Etude Détaillée (\*),
- Dossier d'architecture technique (\*\*),

A faire valider en fin de phase Ph1 par la MOA

(\*) -- Le dossier de fin d'Etude Détaillée contient les SFG & SFD décrivant le système cible. A minima, on trouve :

- MCD,
- un diagramme des cas d'utilisation (ou Use case),
- un diagramme d'états transitions,
- un diagramme des classes,
- une maquette ou un prototype

(\*\*) Le dossier d'architecture technique contient, a minima. :

- l'ébauche de l'architecture logicielle cible (qui sera le « socle » pour faire l'ET)
- une description de l'architecture physique pour les environnements de développements de tests et de production,

## 7. Etude Technique

### 7.1 Objectifs de l'ET

L'Etude Technique est une **étape charnière** entre les étapes de conception précédentes et l'étape de réalisation qui suit (passage d'une « préoccupation Système d'information » à une « préoccupation Système informatique »).

L'objet de l'Etude Technique est de préparer le travail des développeurs, en décrivant (spécifications techniques) de tous les composants logiciels à développer (pour une architecture J2EE, il s'agit de décrire les spécifications techniques des EJB Entity et Session, des servlet, JSP, des procédures stockées (si Oracle), etc...).

L'Etude Technique est une **étape à 100% MOE**. La MOA est donc peu active pendant cette étape. Elle participe toutefois aux instances de pilotage (travail de suivi des travaux de la MOE).

En entrée :

- Les SFD (issus de l'ED).

En sortie :

- Les normes et standards de développement,
- Les spécifications techniques (véritables dossiers de programmation pour les développeurs),
- Scripts de création du schéma de la base de données (script SQL).

### 7.2 Les phases

Les phases :

- Lancement (Ph1)
- Elaboration des spécifications techniques (Ph2)
- Mise en œuvre de la plateforme de développement (Ph3).

#### 7.2.1 Lancement (Ph1)

##### ❶ **Adaptation de l'organisation du projet aux préoccupations de l'ET :**

→ Organisation MOE + organisation MOA + instances de pilotage du projet :

- Maîtrise d'œuvre
- Assistance Maîtrise d'œuvre (si nécessaire)
- Chef de projet informatique (CPI)
- Les profils de la MOE les + courants pour une ET (sous la responsabilité du CPI) :
  - Concepteurs/développeurs (avec des compétences en dev. dans l'architecture logicielle cible prévue ;),
  - Architectes,
  - Experts possibles : administration des données, expert technique, ergonomes (si nécessaire), responsable Qualité (éventuellement pour la MàJ du PAQ), ...
- Planification des instances de pilotage du projet pour la durée de l'ET :
  - Comité de projet,
  - Comité pilotage,

Planning des instances à définir

→ Charges / Planning / Coûts

- Charges et planning précis pour l'ET
- Charges et planning estimés pour les étapes suivantes : à actualiser au fur et à mesure de l'avancement du projet.

#### ② **Mise à niveau (si nécessaire) de la plate-forme de conception technique :**

- par exemple, outil de conception Merise &/ou UML,
- par exemple, installation d'outil de maquettage (obligatoirement compatible avec l'outil de développement).

#### ④ **Actualisation du PAQ et validation par les 2 parties :**

Dans la plupart des cas, il s'agit de la mise à jour du PAQ initialisé en début d'EP et complété en ED.

Importance du rôle d'un PAQ à faire dans l'étape de lancement.

Le PAQ devra décrire valider les procédures visant à garantir la qualité du processus.

- l'organisation de la MOE, de la MOA, des instances de pilotage : fréquence des réunions, rédacteur de l'OJ, rédacteur du CR, ...
- les livrables à produire,
- les ressources nécessaires à la bonne réalisation de l'étape (serveur, postes, licences, ...),
- le planning (mettre en évidence le chemin critique, expliciter les tâches de la MOE, de la MOA, ...)
- les modalités de validation d'un livrable,
- les modalités de livraison d'une nouvelle version, etc...

Le PAQ doit être validé immédiatement (c-à-d dès la fin de la phase de Lancement) par la MOE et la MOA → document contractuel entre les 2 parties (il engage les 2 parties).

## 7.2.2 Elaboration des spécifications techniques (Ph2)

### 7.2.2.1 Préparation du cadre du développement

Préparer le cadre du développement :

- Rédiger le dossier N & S de développement (« Guide du développeur »),
- Rédiger le dossier « Guide de gestion de configuration » (avec CVS/Subversion, PVCS, ...),
- Rédiger le dossier « Norme de livraison »,

### Gestion de configuration des composants logiciels

La Gestion de config. consiste à gérer les versions d'un même composant logiciel. Par exemple, connaître le source :

- ◆ La version du composant source mis en production (dernière version du source mis en production),
- ◆ La version du composant source en cours de recette,
- ◆ La version du composant source en cours de qualification,
- ◆ La version du composant source en cours de développement.

Indispensable sur les gros projets (un même source peut être modifié par X personnes), en TMA, ...

### 7.2.2.2 Identification des composants logiciels à développer

Identifier les composants logiciels à développer à partir des SFD, sur la base d'un dossier de l'architecture logicielle type (à rédiger dans cette étape).

En entrée :

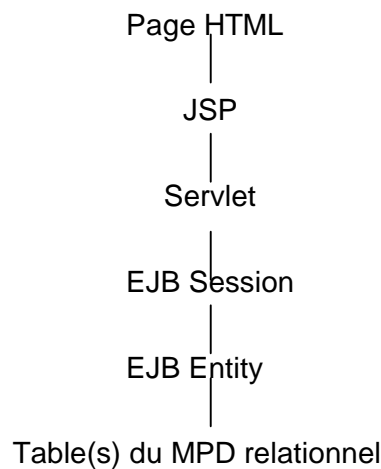
- ❖ Les SFD,
- ❖ Les N& S de développement (les règles de nommage, les règles du développement regroupés éventuellement dans un dossier appelé le « Guide du développeur »).

En sortie :

- ❖ Identification des composants logiciels à développer → Dossiers de programmation (pour les traitements),
- ❖ Scripts de création du schéma de la base de données (script SQL)

### Exemple n°1 :

Schéma d'une architecture logicielle du type J2EE



Identifier les composants logiciels à développer par rapport à l'architecture logicielle J2EE type décrite ci-dessus.

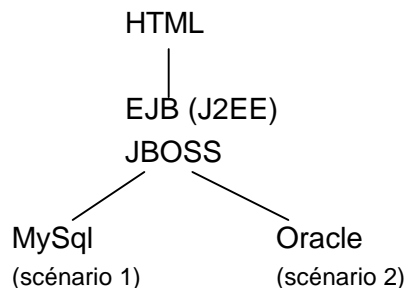
### Exemple n°2 :

Choisir l'architecture système adaptée à la taille du projet (nb accès simultanés, volumes de données stockées en BS, etc ...) pour l'environnement de production.

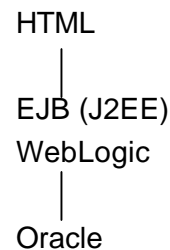
#### Petit projet



#### Projet moyen



#### Grand projet



MySql : open Source

JBOSS : open source

→ Pour les “petites” architectures, privilégier les open source ou les produits libres.

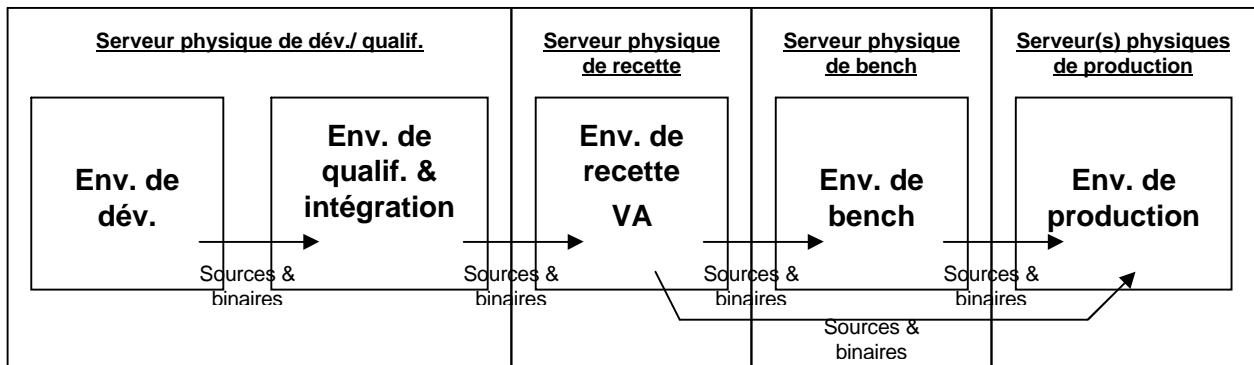
### 7.2.3 Mise en œuvre de la plateforme de développement (Ph3)

#### ❶ Mise en place des plateformes nécessaires au bon déroulement du projet :

- ❖ Installation du serveur et des postes de dev,
- ❖ Installation des licences de développement (framework, serveur d'application, SGBD, etc...)
- ❖ Installation des compilateurs. Par exemple, compilateurs C, Java, ... pour un ou plusieurs OS cible (par exemple AIX, HP-UX, Solaris, ...) (cas de l'AP-HP).

#### ❷ Création des N environnements nécessaires au bon déroulement du projet :

- ❖ Création des environnements dont la MOE aura besoin pour toutes les étapes à venir jusqu'à la fin de la recette + La MOA devra également définir les environnements dont elle a besoin notamment pour la recette.
- ❖ Exemple : utilisation de 4 serveurs physiques pour héberger 5 environnements nécessaires au projet :



#### Remarques :

- ❶ Prévoir un environnement de bench seulement si c'est nécessaire.
- ❷ Indiquer quelle population utilise chacun des environnements :

Tableau : Environnements nécessaires / types d'acteurs :

	MOE	MOA
Env. de dev.	Equipe de projet MOE	
Env. de qualif. & intégration	Equipe de projet MOE	
Env. de recette (VA)		Equipe de recetteurs
Env. de formation		Formateurs
Env. de bench	Equipe de Projet MOE + Production	
Env. de production	Production	

② **Indiquer quel(s) environnement(s) utilisés à chaque étape de la démarche :**

Tableau : Environnements nécessaires / Etapes de la démarche :

	Env. de dev.	Env. de qualif. & intégration	Env. de recette (VA)	Env. de bench	Env. de formation	Env. de production
Nb à prévoir	Un ou plusieurs, suivant le projet	Un ou plusieurs, suivant le projet	Un ou plusieurs, suivant le projet	Un	Un ou plusieurs, suivant le projet	Un
Obligatoire / facultatif	Obligatoire	Obligatoire	Obligatoire	Facultatif	Facultatif	Obligatoire
Schéma Directeur						
Etude Préalable						
Etude Détaillée						
Etude Technique	X					
Dév. & tests unitaires	X	(X)				
Qualif. & intégration	X	X		(X)		
Recette	X	X	X	(X)		
Pré. de la mise en œuvre					X	
Mise en œuvre						X
T.M.A.	X	X	X	X	X	X

③ **Installation des logiciels appropriés pour chaque environnement :**

- Logiciels à installer dans tout environnement :
  - ◆ Instance du SGBD + création du schéma de la base de données
  - ◆ Instance du moniteur transactionnel (ou serveur d'application),
- Autres logiciels à installer dans l'environnement de Développement :
  - ◆ compilateurs
  - ◆ framework
  - ◆ outils de développement
- Autres logiciels à installer dans les environnements de tests (qualif & recette VA) :
  - ◆ Outils de tests non-régression
- Autres logiciels à installer dans les environnements de tests de bench :
  - ◆ Outils de tests de bench
- Autres logiciels à installer dans les environnements de tests de production :
  - ◆ Automate de production (ex : \$Universe)
  - ◆ outil de surveillance (ex : Patrol)
  - ◆ Automate de sauvegarde

④ **Création de la base de données, sous le SGBD prévu (Oracle ou autre).**

- Création d'une instance de la base de données dans l'env. de développement ;
- initialisation des données (=chgt des tables de référence et création de petits jeux de données pour les tables 'Métiers').

## 7.2.4 Fin de l'Etude Technique

Pas de validation comme dans les étapes précédentes (une MOA est rarement capable de valider les résultats de cette étape, totalement interne à la MOE).

## 7.3 Les livrables

- Dossiers de conception technique (identifiant et spécifiant l'ensemble des composants logiciels à développer).
- Dossier N & S de développement (« Guide du développeur »),
- Dossier « Guide de gestion de configuration »,
- Dossier « Norme de livraison »,

## 7.4 Les acteurs de l'ET

### ❶ Les acteurs d'un projet en étape d'ET :

- maîtrise d'œuvre (voir les acteurs) + assistance MOE,
- maîtrise d'ouvrage (voir les acteurs) + assistance MOA

**Rôle limité en ET.**

Préciser les profils des acteurs de la MOE et de la MOA pour une ET. Pour la MOE, on citera plus en détails :

- le rôle de **l'architecte** : élaboration des N&S de développement, élaboration de l'architecture logicielle de tous les processus à développer, garant de l'utilisation des N&S, ...
- le rôle **du DBA** : optimisation du schéma de la base de données, et des requêtes (SQL), , ...
- le rôle du **développeur** (ou développeur HTML pour une maquette Web). En ET, il assiste l'architecte pour la préparation de l'architecture logicielle des composants à développer. En ET, on fait donc appel à des développeurs « seniors » (expérimentés).
- le rôle de quelques **experts** : expert SGBD, expert en architecture, ...

### ❷ Les instances de pilotage en étape d'ET :

- réunions mensuelles pour le comité de pilotage,
- réunions hebdomadaires pour le comité de projet.

Préciser le rôle, les objectifs, les participants aux instances de pilotage.

## 8. Réalisation et tests unitaires

### 8.1 Objectifs de la réalisation

L'étape « Réalisation et tests unitaires » a pour objet de développer tous les composants logiciels de l'application cible.

Cela comprend tous les composants logiciels (progr. Java, progr. C, Shells, procédures SQL, procédures Shells, etc...) suivants :

- des traitements transactionnels (TP) :
  - ❖ application TP Cible,
  - ❖ application TP / système transitoire,
  - ❖ application TP / système dégradé, ...
- des traitements batch :
  - ❖ des traitements d'interfaces
  - ❖ des traitements de migration des données (programmes de déchargement et de rechargement), ...

L'étape « Réalisation et tests unitaires » est une **étape à 100% MOE**. La MOA est donc peu active pendant cette étape. Elle participe toutefois aux instances de pilotage (travail de suivi des travaux de la MOE).

### 8.2 Les phases

- ♦ Lancement (Ph1),
- ♦ Développement et tests unitaires (Ph2).

Préciser les différence entre « test unitaire », « test d'intégration » (ou de qualification) et « test de recette ».

Rq : pas de phase de validation pour passer à l'étape suivante, car il s'agit de 2 phases sous la maîtrise 100% de la MOE.

#### 8.2.1 Lancement (Ph1)

- Mise en place de l'organisation
- Rédaction d'un PAQ dédié au développement.

#### ❶ Organisation de la MOE :

La structure MOE comprend :

- ♦ une équipe de développeurs (éventuellement spécialisée par domaines fonctionnels),
- ♦ une équipe architecture (N & S, traitements communs, optimisation du modèle des données, migration des données, ...)

Plusieurs possibilités d'organiser l'équipe MOE lorsqu'on développe une application dans une architecture N-tier :

- ♦ 1<sup>ère</sup> possibilité : chaque programmeur développe à la fois la partie serveur et la partie cliente.
- ♦ 2<sup>ème</sup> possibilité : constituer une équipe qui développe la partie cliente, et une équipe qui développe la partie serveur.

Le rôle des experts :

- oracle : tuning BD, optimisation des requêtes SQL, ...
- DBA : optimiser les modèles relationnels, partitionnement physique des disques, ...
- moniteur transactionnel : optimisation des requêtes, ...
- déploiement : préparation du déploiement dans l'architecture de production. Pour une appli C/S, prévoir et organiser le déploiement de l'appli "cliente",
- réseau,
- etc...

### ❷ Mise à jour du PAQ :

Dans le PAQ (mise à jour par rapport aux versions précédentes), décrire :

- ✦ L'organisation du projet, dont les instances de pilotage,
- ✦ les responsabilités des parties, y compris des éventuels sous-traitants et co-traitants.

Importance du système qualité pendant tout le projet :

- ✦ PAQ de conception / PAQ de développement : aspect contractuel entre MOE et MOA,
- ✦ revue des livrables (programmes ou documents) avant livraison à la MOA,
- ✦ suivi des charges (consommé, RàF),
- ✦ suivi des bordereaux de livraison / bordereaux de réception.

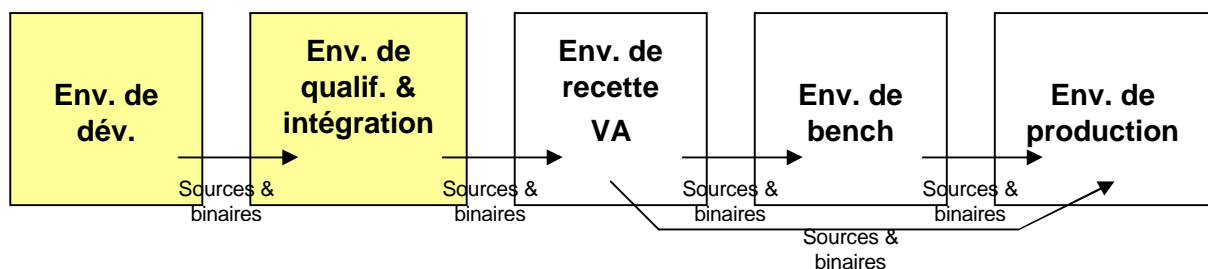
## 8.2.2 Développement + tests unitaires (Ph2)

- Développement de tous les composants logiciels (tels que prévus en fin d'Etude Technique).
- Tests unitaires (chaque composant logiciel doit être testé unitairement par l'ingénieur qu'il l'a développé : il s'agit de vérifier sa conformité aux SFD prises).

### Remarques :

#### ❶ Les différents environnements utilisés :

- environnement de développement,
- environnement de qualification & intégration (les composants une fois testés, sont déployés dans cet environnement pour être qualifiés → voire l'étape suivante).



#### ❷ Faire la documentation associée à l'application :

- ✦ Documentation des programmes sources ( exemple : JavaDoc)
- ✦ Dossier d'installation (explique la procédure à suivre pour installer l'application dans un nouvel environnement)
- ✦ Dossier d'exploitation (Explique à l'équipe de Production le schéma, l'ordonnancement des batch à lancer, leur fréquence, les input/output nécessaires à chaque traitement, etc...).

## 8.3 Les livrables

- ✦ Script SQL (Oracle) de création du schéma de la base de données,
- ✦ Programmes sources & binaires,
- ✦ Documentation des programmes sources,
- ✦ Dossier d'installation : comment déployer l'application dans un environnement ?
  - ✦ Déploiement à chaud : sans arrêt de l'application
  - ✦ Déploiement à froid : avec arrêt de l'application.
- ✦ Dossier d'exploitation.

Attention :

- 1) Ne pas livrer de « dossier des tests unitaires » → Piège !!!
- 2) Attention à éviter les migrations techniques pendant les développements du type :
  - migration oracle 8.1.7 vers Oracle 9.i,
  - migration WebLogic 6.1 vers WebLogic 8.1.

## 8.4 Les acteurs de la Réalisation

### ① Les acteurs d'un projet en étape de réalisation :

- maîtrise d'œuvre (voir les acteurs) + assistance MOE,
- maîtrise d'ouvrage (voir les acteurs) + assistance MOA

**Rôle limité en ET.**

Préciser les profils des acteurs de la MOE et de la MOA pour l'étape de Réalisation. Pour la MOE, on citera plus en détails :

- le rôle de **l'architecte** :, garant de l'utilisation des N&S auprès des développeurs, optimisation du code, ...
- le rôle **du DBA** : optimisation du schéma de la base de données, et des requêtes (SQL), , ...
- le rôle du **développeur**. Il développe suivant les spécif. de l'ET et réalise les tests unitaires,
- le rôle de quelques **experts** : expert SGBD, expert en architecture, ...

### ② Les instances de pilotage en étape de réalisation :

- réunions mensuelles pour le comité de pilotage,
- réunions hebdomadaires pour le comité de projet.

Préciser le rôle, les objectifs, les participants aux instances de pilotage.

## 9. Qualification MOE

### Autres termes pour désigner cette étape :

- Tests d'Intégration,
- Recette MOE.

### 9.1 Objectifs

La MOE doit faire des tests complets avant de livrer l'application à la MOA pour recette. C'est une étape très lourde, estimée à environ 30% des charges de « Réalisation et tests unitaires ».

Ces tests sont de 2 natures :

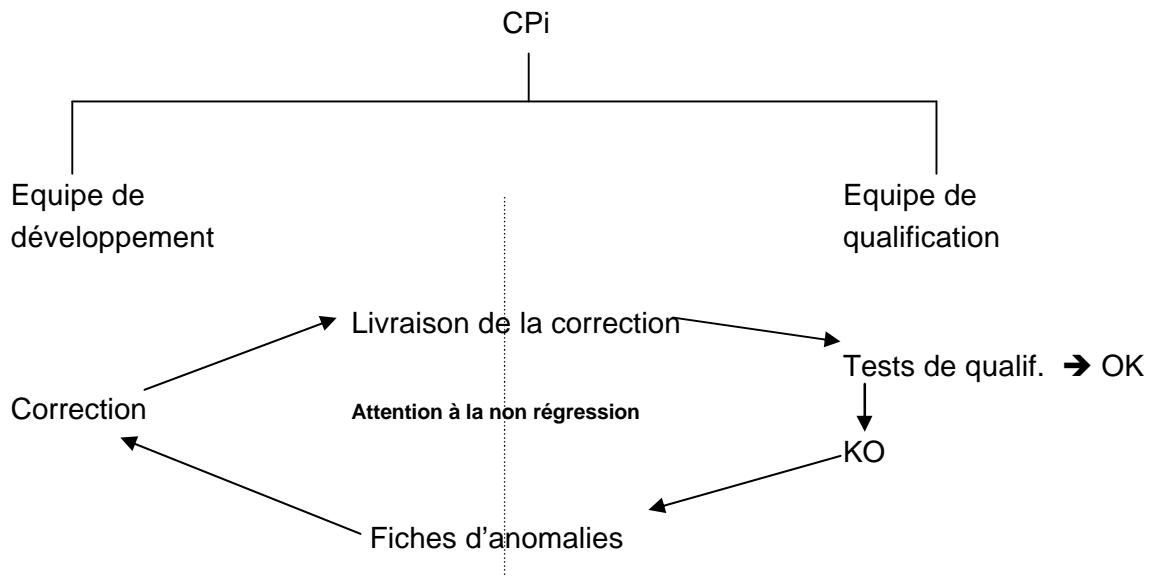
- ◆ des **tests fonctionnels (obligatoire !)** :
  - Intégrer & qualifier les programmes transactionnels et batch de la cible,
  - Intégrer & qualifier la reprise des données,
  - Intégrer & qualifier les procédures d'installation et d'exploitation (création d'un schéma, chgt des tables de référence, etc...),
- ◆ des **tests techniques (facultatif, suivant la criticité de l'application développée !)** :
  - tests de performances (Bench) tant sur le transactionnel (exemple : site internet [www.anpe.fr](http://www.anpe.fr)) que sur le batch (exemple : traitement de la paie en fin de mois, traitement des dossiers IJM à la CANAM).
  - Tests d'exécution des traitements batch dans une plage horaire prédéfinie.  
Par exemple, entre 20h et 6h.
  - tests de montée en charge d'un nb d'utilisateurs de + en +.  
Par exemple, temps de réponse TP < 1s si moins de 5000 connexions simultanées.
  - tests avec un volume de données croissant dans les tables de la base de données,  
Par exemple, temps d'exécution d'une requête SQL si 1 million de lignes, 10 millions, 100 millions de lignes (ex : Renault, application d'incidentologie).
  - tests avec indisponibilité partielle d'une partie des ressources (par exemple, simuler le fonctionnement d'un gros site Web si certaines instances WebLogic de production sont « tombées »), ...

### 9.2 Les phases

- ◆ Lancement (Ph1),
- ◆ Préparation de la qualification (Ph2),
- ◆ Exécution des tests d'intégration (Ph3),
- ◆ Préparation de la livraison auprès de la MOA (Ph4).

## 9.2.1 Lancement (Ph1)

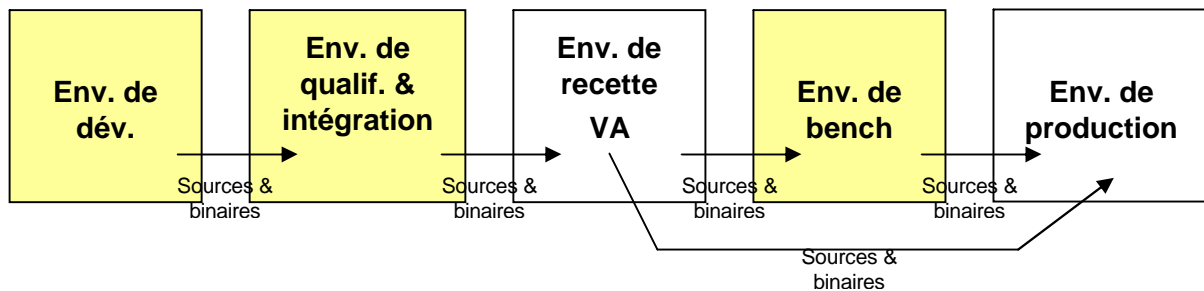
Importance de distinguer au sein de la MOE l'équipe de développement de l'équipe d'intégration.



Dans le PAQ, on fixera la durée de la qualification (notamment le nombre d'itérations).

### Les différents environnements utilisés :

- environnement de développement,
- environnement de qualification & intégration,
- environnement de bench (si nécessaire).



### Possibilité d'automatisation des tests :

- ◆ tests fonctionnels (tests pour vérifier la conformité aux SFD),
- ◆ tests techniques :
  - Installer les outils de tests, si nécessaire.

## 9.2.2 Préparation de la Qualification (Ph2)

**Cas de test (déf.) :** Description d'un test (fonctionnel) à réaliser.  
Par exemple, tester la page de chgt d'adresse d'un client.

**Jeux de données (déf.) :** Description du jeu de données à utiliser pour exécuter un cas de test. On décrit les données réelles à saisir en entrée du cas de test (input), et le résultat escompté (output).

Préparation des cas de tests et des jeux par la MOE :

- ✦ Préparation des jeux d'essais (jeux d'essais fonctionnels et jeux d'essais techniques).
- ✦ Rédaction du "plan de qualification" : jeux d'essais à exécuter, résultats attendus.

### 2 types de tests de qualification :

- **tests fonctionnels** visant à vérifier la conformité du codage des règles de gestion dans les programmes, à vérifier le bon fonctionnement de l'application dans le SI (échanges avec les appli interfacées) + vérifier la non-régression.
- **tests techniques** : tests de performance, de temps de réponse, de montée en charge.

### 9.2.3 Exécution de la qualification (Ph3)

Itération des 5 tâches suivantes :

	Equipe MOE de dev.	Equipe MOE de qualif.
Exécution de la qualification (=Passage des jeux d'essais fonctionnels et/ou techniques)		X
Identification des dysfonctionnements (rédaction des fiches d'anomalies)		X
Correction + tests unitaires, après instruction des fiches de développement	X	
Livraison de la correction dans l'env. de qualification	X	
Nouvelle qualification (2 <sup>ème</sup> passage des jeux d'essais) = tests de non-régression		X

### Remarques :

#### ❶ **Remarques sur les tests de non régression :**

Les tests de non-régression doivent être réalisées sur les composants logiciels qui ont été corrigés par l'équipe de dev., mais aussi sur la totalité des autres processus (y compris ceux qui ont déjà fait l'objet d'une 1<sup>ère</sup> campagne de tests de qualification).

Itération de l'exécution de la totalité des cas de tests suivants :

- Tests fonctionnels,
- Tests techniques.

❷ **Classer les anomalies par gravité** (classification à utiliser également en recette – classification à mettre dans le PAQ, car contractuel entre les 2 parties MOE et MOA) :

- ✦ ano grave / bloquante,
- ✦ ano gênante / majeure,
- ✦ ano non gênante / mineure,
- ✦ amélioration / évolution.

❸ **Rédaction du « dossier de qualification »** : il décrit les anomalies rencontrées, les corrections réalisées, ... il trace tous les tests réalisés par l'équipe de qualification.

### 9.2.4 Préparation de la livraison à la MOA (Ph4)

Suivant les conditions prévues au PAQ, on peut livrer – en règle générale – l'application (sources & exécutables) lorsque les conditions suivantes sont respectées :

- Zéro anomalie bloquante,
- Nb anomalies majeures < seuil-1 (défini dans le PAQ) ou le contrat,
- Nb anomalies mineures < seuil-2 (défini dans le PAQ) ou le contrat.

Une fois, l'application livrée en recette, l'équipe MOE est réduite. Quelques développeurs (MOE) resteront toutefois sur le projet pendant la recette, afin de corriger les anomalies identifiées par la MOA en recette.

### 9.3 Les livrables

- ◆ **Plan de la qualification** (à faire par la MOE dès la fin de l'ED) : il décrit les cas de tests et les jeux de données prévues pour la qualification de l'application,
- ◆ **Dossier de qualification** : il trace l'exécution de la qualification MOE : il contient donc tous les cas de tests qui ont été exécutés et les résultats constatés (il contient donc les fiches d'anomalies). Il est à rédiger au fur et à mesure de l'exécution des campagnes de tests.

### 9.4 Les acteurs de la qualification

#### ① Les acteurs d'un projet en étape de qualification :

- maîtrise d'œuvre (voir les acteurs) + assistance MOE,
- maîtrise d'ouvrage (voir les acteurs) + assistance MOA

**Rôle limité en ET.**

Préciser les profils des acteurs de la MOE et de la MOA pour une étape de qualification. Pour la MOE, on citera plus en détails :

- le rôle du **développeur**. Il corrige les anomalies identifiées par le qualificateur.
- le rôle du **qualificateur** : il prépare le plan de qualification (à partir des SFD), puis exécute les cas de tests, et rédige – si nécessaire les fiches d'anomalies, ...

Si nécessaire, peuvent encore intervenir :

- le rôle de **l'architecte** : , garant de l'utilisation des N&S auprès des développeurs, optimisation du code, ...
- le rôle **du DBA** : optimisation du schéma de la base de données, et des requêtes (SQL), , ...

#### ② Les instances de pilotage en étape de qualification :

- réunions mensuelles pour le comité de pilotage,
- réunions hebdomadaires pour le comité de projet.

Préciser le rôle, les objectifs, les participants aux instances de pilotage.

## 10. Recette

### 10.1 Objectifs

La MOA doit faire des tests complets avant de mettre en production l'application.

Dans le même temps, la MOE maintient une équipe de développeurs réduite pour assurer la correction des anomalies identifiées pendant la recette par la MOA.

Ces tests couvrent :

- ◆ Recette fonctionnelle des programmes TP / batch de la cible,
- ◆ Recette fonctionnelle de la reprise des données.

NB : le périmètre de la recette est différents des tests de qualification effectués par la MOE.

De plus, les tests de la MOA peuvent être accompagnés d'autres tâches, telles des tests auprès d'un échantillon de futurs utilisateurs de l'application (échantillon représentatif).

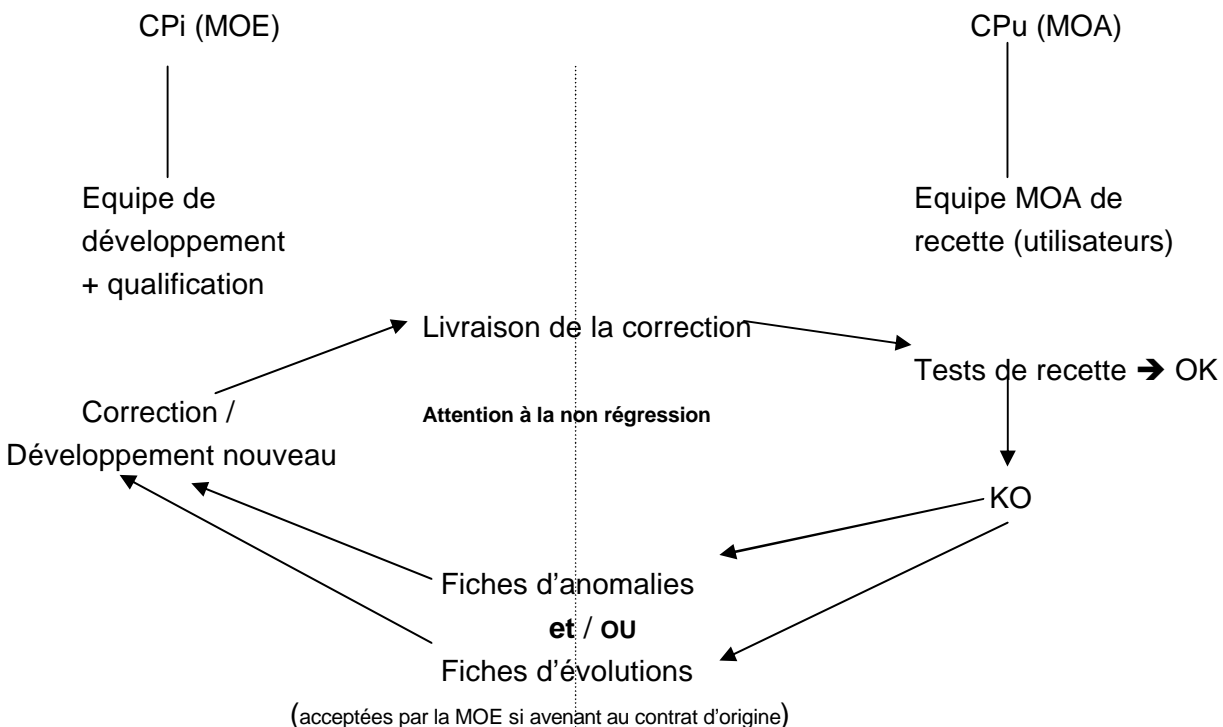
### 10.2 Les phases

- ◆ Lancement (Ph1),
- ◆ Préparation des jeux d'essais de recette (Ph2),
- ◆ Exécution de la recette (Ph3).

#### 10.2.1 Lancement (Ph1)

Une organisation particulière au sein de la MOA: une équipe dédiée au tests de recette.

Préciser le rôle du MOE et du MOA durant la recette.

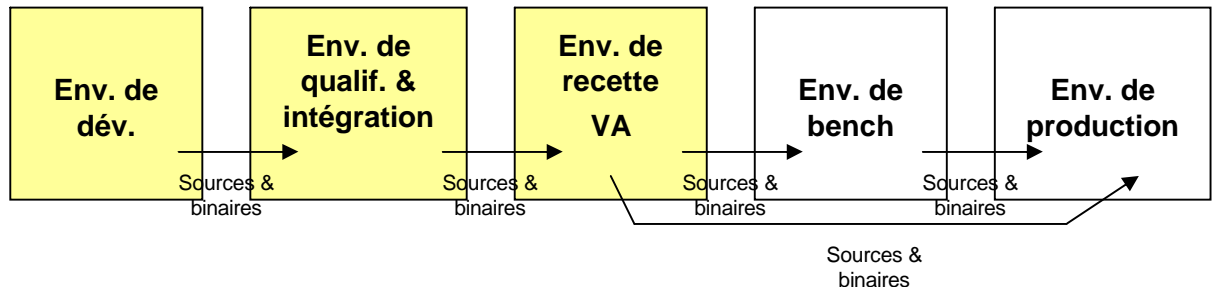


Traitement de correction par la MOE variable, suivant la gravité des anomalies identifiées par la MOA :

- Si fiche d'anomalie 'bloquante' → correction + relivraison à la MOA sous 48 ou 72 heures (suivant contrat passé)
- Dans les autres cas (gravité = 'majeur' ou 'mineur') → regroupement des anomalies dans des lots.

### Les différents environnements utilisés par la MOE :

- environnement de développement,
- environnement de qualification & intégration.



### Les différents environnements utilisés par la MOA :

- environnement de recette.

Dans le PAQ, on fixera la durée de la recette (la durée de la recette génère des charges dans la MOE, que la MOE doit connaître au préalable).

## 10.2.2 Préparation des cas de tests et des jeux d'essais de recette (Ph2)

**Cas de test (déf.) :** Description d'un test fonctionnel à réaliser. Par exemple, tester la page de chgt d'adresse d'un client.

**Jeux de données (déf.) :** Description du jeu de données à utiliser pour exécuter un cas de test. On décrit les données les données réelles à saisir en entrée du cas de test (input), et le résultat escompté (output).

Préparation des cas de tests et des jeux d'essais (par la MOA) et non par la MOE :

- ♦ jeux d'essais fonctionnels TP et bach,
- ♦ jeux d'essais fonctionnels TP et bach de reprise des données, ...

Rq : La MOE peut fournir son assistance à la MOA dans la préparation des cas de tests et des jeux de données.

## 10.2.3 Exécution de la recette (Ph3)

### Une recette se déroule en 2 temps :

- recette dans un environnement de tests (**VA ou VABF**),  
= Recette dans un environnement spécifique, par une équipe MOA du projet.
- recette dans un environnement de production (**VSR**).  
= Recette dans un environnement réel de production (conditions réelles d'utilisation), mais limité à un site pilote.

Plusieurs parties à recetter :

- **recette fonctionnelle** (applicatifs et/ou progiciels): vérification de la conformité des règles développées avec les SFD,
  - **recette de la reprise des données** : vérification de la cohérence et de l'intégrité des données migrées de l'ancien système, vérification que l'épuration effectuée est OK, ...
  - **recette technique** : vérification des performances, de la solidité de l'appli lors d'une montée en charge, de l'implémentation conforme des N&S, bench de performances, ...
- **Rarement en recette !!**

**Déroulement du processus de recette : itération des 7 tâches suivantes :**

	Equipe MOE de dév.	Equipe MOE de qualif.	Equipe MOA de recette
Exécution de la recette VA (= passage des jeux d'essais fonctionnels et/ou techniques)			<b>x</b>
Identification des dysfonctionnements (rédaction des fiches d'anomalies)			<b>x</b>
Correction + tests unitaires, après instruction des fiches de développement	<b>x</b>		
Livraison de la correction dans l'env. de qualification	<b>x</b>		
Qualification de la correction		<b>x</b>	
Livraison de la correction dans l'env. de recette		<b>x</b>	
Nouvelle recette (2 <sup>ème</sup> passage des jeux d'essais) = tests de non-régression des corrections réalisées			<b>x</b>

### Remarques :

#### ❶ **Remarques sur les tests de non régression :**

Les tests de non-régression doivent être réalisés sur les composants logiciels qui ont été corrigés par la MOE, mais aussi sur la totalité des autres processus (y compris ceux qui ont déjà fait l'objet d'une 1<sup>ère</sup> campagne de tests de recette).

#### ❷ **Remarques sur les fiches d'anomalies :**

Les fiches d'anomalies sont rédigées par la MOA et adressées à la MOE pour correction. Les anomalies doivent être reproductibles par la MOE.

#### ❸ **Classer les anomalies par gravité (classification à utiliser également en recette) :**

- ✦ ano grave / bloquante,
- ✦ ano majeure,
- ✦ ano mineure,
- ✦ amélioration / évolution.

Complexité lorsque 2 systèmes informatiques doivent co-exister.

## 10.3 Les 2 types de recette

Les 2 types de recette sont :

- La Vérification d'Aptitude (VA), aussi appelée Vérification d'Aptitude au Bon Fonctionnement (VABF),
- La Vérification de Service Régulier (VSR)

### 10.3.1 La VA

#### La VA – ses caractéristiques :

- Elle consiste à exécuter tous les cas de tests fonctionnels (et techniques si nécessaire) de manière exhaustive avec des jeux d'essais à blanc,
- Elle dure – en règle générale - de 15 jrs à 3 mois, suivant la taille du projet,
- Elle s'exécute dans un environnement de recette.
- Le « dossier de recette VA » trace toutes cas à passer et les résultats des tests constatés.

### 10.3.2 La VSR

#### La VSR – ses caractéristiques :

- Elle se déroule sur un site pilote, avant généralisation du déploiement, donc l'environnement de production,
- Le site pilote a un fonctionnement normal. Il utilise des données réelles.
- Elle dure – en règle générale - de 2 à 6 mois, suivant la taille du projet,
- Il n'y a pas de dossier de recette.

**Site pilote :** site qui utilise avant les autres sites (ou services) de l'entreprise la nouvelle application qui vient d'être recettée. Les autres sites (ou services) continuent – en règle générale – à utiliser l'ancienne application.

## 10.4 Les livrables (produits par la MOA)

- ◆ **Plan de la recette** (à faire par la MOA dès la fin de l'ED) : il décrit les cas de tests et les jeux de données prévues pour la recette,
- ◆ **Dossier de recette** : il trace l'exécution de la recette VA : il contient donc tous les cas de tests qui ont été exécutés et les résultats constatés (il contient donc les fiches d'anomalies),
- ◆ **Dossier « bilan de la recette VA & VSR »** (à rédiger au fur et à mesure de l'exécution des campagnes de tests),
- ◆ **PV de recette** (PV de VA & PV de VSR). Ces documents sont des documents contractuels prévus pour les projets au forfait (il déclenche – en règle générale – le paiement d'une tranche des prestations d'un projet au forfait).

## 10.5 Les acteurs de la recette

### ❶ Les acteurs d'un projet en étape de qualification :

- maîtrise d'œuvre (voir les acteurs) + assistance MOE,
- maîtrise d'ouvrage (voir les acteurs) + assistance MOA.

Préciser les profils des acteurs de la MOE et de la MOA pour une étape de recette. Pour la MOE, on citera plus en détails :

- le rôle du **développeur**. Il corrige les anomalies identifiées par le qualificateur.

- le rôle du **qualificateur** : il requalifie les corrections réalisées suite aux anomalies détectées par la MOA, ...

② **Les instances de pilotage en étape de qualification :**

- réunions mensuelles pour le comité de pilotage,
- réunions hebdomadaires pour le comité de projet.

Préciser le rôle, les objectifs, les participants aux instances de pilotage.

## 11. Préparation à la mise en oeuvre

### 11.1 Les phases

Les phases :

- ◆ Conduite du changement (Ph1) → Phase exclusivement à la charge de la MOA
- ◆ Préparation du déploiement (Ph2) → Phase exclusivement à la charge de la Production

#### 11.1.1 Conduite du changement (Ph1)

Conduite du changement, à la charge de la MOA exclusivement :

- mesures d'accompagnement (exemple : évolution des tâches du personnel d'une entreprise, reconversion en cas de licenciement, ...).
- Communication, formation (éviter les "effets tunnel").
- Information vers les utilisateurs finals (journaux, séminaires de communication, ....).
- transfert de compétences (entre un éditeur de progiciel et une société cliente, entre un SSII maître d'oeuvre et une DSI, ...).

Exemples de conduite du changement :

##### Ex. n°1 :

Entreprise X → décision de la DG de retarder la mise en production d'une nouvelle version devant des risques de grèves du personnel qui considèrent que leur métier évolue et qu'il est sous-formé → La MOA (en accord avec la DG de l'entreprise) décide de faire des actions de communication (séminaires) auprès de toutes les directions régionales.

##### Ex. n°2 :

La MOA (en accord avec la DG de l'entreprise) a mis en place une structure permettant de répondre aux 500 emails quotidiens d'informations d'internautes candidats ou employeurs sur le site Web de l'entreprise X.

#### 11.1.2 Préparation du déploiement (Ph2)

Préparer le plan de déploiement dans l'architecture de production.

**Plan déploiement :** Dossier qui décrit :

- les modalités du déploiement de l'architecture physique (serveurs et PC). Opération complexe notamment pour les applications C/S avec un client "lourd".
- les modalités de la migration des données. Opération complexe, lorsque le volume des données est très important, lorsque le service est ouvert 7j/7j et 24h/24h). Cas des sites Internet, ou des applications utilisées par des sites réparti dans le monde entier.

#### 11.1.3 Les acteurs

Les acteurs :

- La production, pour la préparation de l'architecture technique physique,
- La direction de la formation, pour les plans de formation,
- La direction du personnel, pour des mesures possibles de reconversion/licenciement, ...

➤ La MOA, ...

#### **11.1.4 Livrables**

Les livrables de l'étape :

- support de communication,
- guide utilisateur,
- supports de formation,
- ....

## 12. Mise en œuvre / Déploiement

### 12.1 Les phases

Les phases :

- Déploiement de l'architecture technique (Ph1)
- Déploiement de l'application (Ph2)
- Reprise des données (Ph3)
- Automatisation de l'environnement de production (Ph4)

Important :

- Rôle de la MOA : néant
- Rôle de la MOE : équipe de la production + assistance de la MOE.

### 12.2 Déploiement

#### 12.2.1 Déploiement de l'architecture technique (Ph1)

Il s'agit de déployer l'architecture technique (ou architecture physique) de l'environnement de production.

##### ❶ Mise en place des serveurs de production :

Achat des matériels, logiciels, ... mise en place des serveurs de production, installation, paramétrage, mise en place de mirroring, de RAID5, de sites avec bascules automatiques (citer l'exemple de l'incendie du Crédit Lyonnais en 1999).

##### ❷ Déploiement des licences nécessaires au fonctionnement de l'application :

- des licences tels Oracle (noyau), WebLogic (licence serveur), etc... au niveau des serveurs,
- des éventuelles licences sur les postes PC des utilisateurs.

##### ❸ Mise en place, si nécessaire, d'une architecture redondée :

- ✦ N serveurs d'applications J2EE (WebLogic) dans l'env. de production,
- ✦ Utilisation de l'option 'Oracle Parallel Serveur' (OPS) d'Oracle ou Oracle Rack,
- ✦ Redondance des lignes réseau, ...
- ✦ Etc...

##### ❹ Préparer le déploiement (ou l'update) des PC :

Préparer le déploiement sur les PC, voire le remplacement des PC, la mise à niveau du réseau, ... notamment pour les applications C/S.

##### ❺ Mise à niveau du réseau WAN et LAN :

- ✦ Notamment passage à IP sur le WAN,
- ✦ augmentation de la bande passante du WAN, ...

#### 12.2.2 Déploiement de l'application (Ph2)

Déploiement des progiciels et des applications développées en spécifique (binaires).

### 12.2.3 Reprise des données (Ph3)

Migration des données dans l'architecture cible, avec les données réelles de l'ancien système arrêté (par exemple, la veille). Au quel cas, cette opération est souvent à réaliser pdt un WE, ou une nuit.

Pour le site pilote (VSR) :

→ Reprise des données en début de VSR.

Pour les autres sites :

→ Reprise des données en fin de VSR.

### 12.2.4 Compléments à faire l'environnement de production (Ph4)

#### ❶ Mise en place des outils d'exploitation et d'administration du système :

Les outils nécessaires, une fois l'application finale mise en exploitation sont les suivants :

- sauvegarde / restauration,
- ordonnancement des batch (exemple : \$UNIVERSE),
- Outil de surveillance système (exemple : PATROL).

#### ❷ Mise en place éventuellement d'une machine de backup, ...

## 12.3 Lotissement

Risques liés à une mise en oeuvre par lots successifs :

- 2 systèmes informatiques à faire cohabiter,
- interfaces et systèmes transitoires,
- inconvénients pour les utilisateurs à utiliser 2 systèmes.

## 13. Bilan

---

Importance du bilan dans une SSII.

- Capitaliser sur le chiffreage des UO,
- Capitaliser sur les métiers du client,
- Capitaliser sur l'architecture logicielle,
- Capitaliser sur les contacts clients, etc...

## 14. Garantie

---

Garantie = correction des anomalies par la MOE non détectées en Qualification et en Recette ; et souvent due au titre d'un marché au FORFAIT entre une SSII et une société cliente.

Elle démarre, en général, à la fin de la recette ou à la fin du déploiement.

Procédures contractuelles à mettre en œuvre applicables à l'issue du déploiement :

- délai d'intervention en cas d'anomalie bloquante sur problème software ou hardware
- délai d'intervention en cas d'anomalie majeure (ou mineure) sur problème software ou hardware

En règle générale, la SSII n'a plus d'équipe MOE affecté au projet. Elle doit toutefois être à même d'intervenir suivant les délais définis au contrat.

## 15. Maintenance / TMA

---

Introduire le vocabulaire suivant :

- ♦ TMA : Tierce Maintenance Applicative,

Les 4 types de maintenance :

- maintenance corrective,
- maintenance préventive (exemple : faire évoluer un séquence de code ... mal écrite, peu performante, ....)
- maintenance évolutive (exemple : évolution de la réglementation).
- Maintenance adaptative (exemple : chgt de version de l'OS, du SGBD, ...).

Pour la TMA, faire un PAQ dédié aux spécificités de ce type d'activités.

### 15.1 Organisation

- Evolution de l'organisation : équipe Front office + équipe Back office.  
Font office : métier, à proximité du client,  
Back office : éventuellement distant, orienté technique.
- Evolution vers du near-shore (Espagne, Maroc, ...) ou du off-shore (Inde, Roumanie, Chine, ...).

### 15.2 Outils

Outils impératifs dans une TMA :

- Outil de gestion des demandes (faits techniques) : Mantis, Bugzilla, XAGA,
- Outil de gestion de config. CVS (Subversion) : indispensable, qd développement de plisieurs versions du logiciel simultanément.
- Outil de qualimétrie, analyse d'impacts : CAST, System Code (ACFO). Ces produits reconnaissent la syntaxe de certains langages de programmation. Ils peuvent servir à mesurer la qualité du code source, identifier du code mort, faire des analyses d'impacts, ...

## 16. La qualité dans un projet informatique

### 16.1 Objectifs d'un PAQ

Le PAQ décrit les modalités d'assurance qualité informatique entre une MOE et une MOA pour l'ensemble des étapes d'un projet.

Le PAQ est rédigé en début du projet. Il est alors validé par les 2 parties MOE & MOA. Il est applicable par les 2 parties pendant tout le projet.

### 16.2 Dans quels cas faire un PAQ ?

Le PAQ est « obligatoire » (indispensable) :

- dans tout projet au forfait entre une SSII et un client,
- dans tout projet au-delà d'une certaine taille (par exemple > 400 / 500 j\*h)

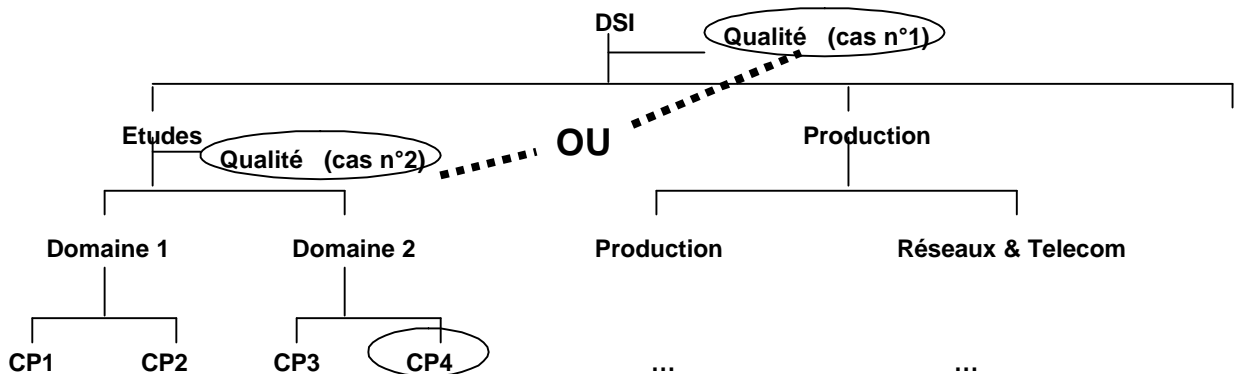
Le PAQ est inutile dans les cas suivants :

- prestations d'une SSII en régie,
- petit projet (par exemple < 200 j \*h).

### 16.3 Quels acteurs participent à la rédaction d'un PAQ ?

#### Cas n°1 – projet interne :

Dans ce cas, il est rédigé par le CPi, avec l'assistance de la Direction Qualité de sa DSI.



#### Cas n°2 – projet confié à une SSII :

Dans ce cas, il est rédigé par le CPi de la SSII, avec la collaboration du CPi du client et l'assistance des Directions Qualité de la SSII et du Client.

### 16.4 Contenu du PAQ

Le sommaire d'un PAQ est toujours le même. Ce sommaire type est défini par la norme AFNOR.

Voir un exemple du PAQ fourni par l'enseignant.

Les principaux sur lesquels il faut porter son attention :

- l'organisation du projet (et donc la disponibilité des ressources MOE & MOA),
- les procédures de livraison
- les procédures de réception et de validation des livrables
- les indicateurs et métriques qualité

## 17. Les adaptations de la démarche

---

### 17.1 Adaptation de la démarche de Conduite standard

Adaptation de la démarche de Conduite de projet si :

- grand projet (prévoir un découpage en sous-projets + lotissement éventuel),
- moyen projet,
- petit projet (moins d'étapes, moins de livrables, moins de points de validation, ...).

Autres adaptations suivant la nature du projet :

- projet de migration,
- projet d'intégration (ERP par exemple).

### 17.2 Autres démarches / méthodes de conduite de projet

#### 17.2.1 RAD

Rapid Application Développement (RAD).

#### 17.2.2 SDM/S

SDM/S (9 étapes, chacune décomposée en 9 phases) :

- DBS,
- CAS
- SES,
- SIS,
- PRG,
- TST,
- CONV,
- INST

#### 17.2.3 Autres démarches

Démarche itérative (en spirale).

Pilotage par les enjeux.

## **18. Qu'est-ce qu'un chef de projet ?**

---

### **18.1 Rôle du CPi**

Importance du rôle du CPi : véritable chef d'orchestre.

### **18.2 Qualités du CPi**

Les 7 qualités du CP : voir note OM.

## 19. Méthode d'évaluation des charges

---

Donner des métriques pour calculer les charges brutes par unités d'oeuvre, en fonction de la complexité TS, S, M, C, TC.

Cette méthode permet de calculer des charges brutes, auxquelles il faut rajouter des charges d'encadrement, des charges d'assurance qualité, d'administration des données, ...

### 19.1 Ratios des étapes

Rappeler le poids de chaque étape :

→ Schéma directeur	
→ Etude Préalable	5%
→ Etude Détaillée	15%
→ Etude Technique	10%
→ Réalisation	35%
→ Intégration	15%
→ Recette	10%
→ Préparation de la mise en oeuvre	
→ Mise en oeuvre	5%
→ Bilan	

2 méthodes :

- PERT,
- GANTT.

3 préoccupations :

- évaluer les charges et les planifier,
- gérer le suivi et les écarts,
- prise de décisions (si retards sur le chemin critique).

### 19.2 Méthodologies de chiffrage des charges

Décrire les méthodologies de chiffrage des charges sur un projet :

- ◆ recensement des unités d'œuvre,
- ◆ attribution d'un facteur de complexité : TS, S, M, C, TC
- ◆ calcul des charges brutes,
- ◆ ajout des charges d'encadrement (+15% chez SOPRA),