REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE « SAAD DAHLEB » DE BLIDA

Faculté des sciences
Département de l'informatique

Mémoire du projet de fin d'étude

Pour l'obtention du diplôme d'ingénieur d'état

Spécialité : Informatique

Option : Système d'Information Avancé

Thème :

La gestion de la qualité du logiciel pour la sécurité des technologies de l'information

Proposé et encadré par:

- M^me^.Admane
- M^me^. Targui
- M^elle^. Khelifa

Etudié par :

- M^elle^. Chemlal wassila
- M^r^. Haddoum karim

Devant le jury :

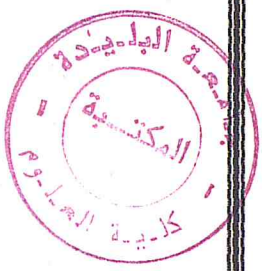
- **Président** : M^r^.Boukhlef
- **Examinateur** : M^me^.Boumahdi
- **Promoteur** : M^me^.Benstiti

Promotion : 2003-2004

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE « SAAD DAHLEB» DE BLIDA



Faculté des sciences
Département de l'informatique

Mémoire du projet de fin d'étude

Pour l'obtention du diplôme d'ingénieur d'état

***Spécialité** : Informatique*

***Option :** Système d'Information Avancé*

Thème :

La gestion de la qualité du logiciel pour la sécurité des technologies de l'information

Proposé et encadré par:

- M^me^.Admane
- M^me^. Targui
- M^elle^. Khelifa

Etudié par :

- M^elle^. Chemlal wassila
- M^r^. Haddoum karim

Devant le jury :

- **Président** : M^r^.Boukhlef
- **Examinateur :** M^me^.Boumahdi
- **Promoteur :** M^me^.Benstiti

Promotion : 2003-2004

Remerciements

Nous remercions avant tout le bon dieu qui nous a aidé à réaliser ce modeste travail.

Nous tenons à remercier nos encadreurs Melle.khelifa , Mme .targui et Mme. Admane pour leur aide, leur patience, leur disponibilité et leur compréhensibilité.

Nous remercions les membres du jury pour nous avoir fait l'honneur de juger notre travail.

Nous adressons nos sincères remerciements à Mme.Benstiti.

Nous tenons à exprimer nos profondes gratitudes et nos vifs remerciements à ma mère, M'.Boutheldja MOHAMED, M' .YOUCEF,FETHI RIGHI ET M'.Hantabli Hamza pour leurs aides, leurs conseils et leurs préoccupations, durant notre projet, ainsi leurs aides morales.

Nous remercions ceux qui ont bien voulu nous accueillir au sein de la Division Système d'Information(DSI) de CERIST

Nous remercions tous les enseignants de la faculté des sciences de BLIDA et surtout ceux du département informatique.

Nous remercions, de tout coeur, tous ceux qui ont contribué de prés ou de loin à la réalisation de ce travail.

Dédicace

Ce mémoire est dédié :

A mon très cher regretté père et à ma très

Chère mère pour leur soutien durant

toutes mes études,

Pour leur bienveillance, leurs efforts constants dans mes études,

et pour leurs encouragements,

A mes frères kamel,Nacer,Zohier,A.Rezak et à ma sœur Ratiba

A mon fiancé Mohamed pour m'avoir soutenu durant mes études,

et toute sa famille surtout Nassira, Assoum et Ahmed.

A mon beau père Yahia, ma belle mère Fatma.

A toutes mes belles soeurs, mes nièces, et mes neveux

A toute la famille CHEMLAL, BOUTHELDJA ET RIGHI

A mon binôme karim et à toute sa famille

A Fadhila, Fatima, Nadia, Nassima, SELMA,

Djaouida, Karima et Messouda.

A tous mes ami(e) s qui j'aime et qui m'aiment.

C.Wassila

Dédicace

Ce mémoire est dédié :

A mes très chers parents pour leur soutien durant

toute ma carrière,

Pour leur bienveillance, leurs efforts constants dans mes études,

Et Pour leur encouragements,

A mon frère RABIE

A mes sœurs Assia, Mounia, Sabrina et Lamia

A toute ma famille,

A mon binôme Wassila et à toute sa famille

A Mohamed, Fateh, Hamza, Selma, Nassima et Nadia

A tous mes ami(e) s.

H.Karim

TABLE DES MATIERES

Chapitre 3.Le modèle CMM

Chapitre 4.Système de gestion de la qualité

Chapitre 5.Développement du système de gestion de la qualité

LISTE DES FIGURES

Chapitre 1

Chapitre 2

Chapitre 3

Chapitre 4

Chapitre 5

La gestion de la qualité du logiciel pour la sécurité des technologies de l'information

Résumé : La sécurité des technologies d'informations (TI) vise à garantir l'intégrité, la disponibilité et la confidentialité de l'information. Elle inclut aussi la sécurité des matériels, des logiciels et des réseaux.

Pour cela, l'entreprise doit définir et implanter des programmes qualité dans le secteur des TI, visant l'amélioration de développement des logiciels.

Les spécialistes du domaine de logiciels ont proposé des démarches de certification et d'évaluation des processus.

Notre travail est intéressé par une phase de projet de la gestion de la qualité :

C'est la définition et la conception d'un système de gestion de la qualité du logiciel. Durant cette phase, il faut définir tout ce qui est lié au concept de qualité du logiciel.

Au cours de notre projet, nous avons développé un outil de gestion de la qualité du logiciel, simple d'utilisation qui offre au manager de structurer les processus de développement du logiciels. On a adapté dans notre projet, le modèle de maturité CMM, comme référence.

Mots clés : Qualité logiciel, certification, évaluation des processus, CMM, UML.

The software's quality management for information technologies's security

Abstract: The information technology's security aims at guarantee the integrity, availability and the confidentiality of information. It includes the hardware's, software's and network's security. That is why societies should define and implement quality's in the sector of TI.

The software's experts have proposed a certification and process's value models.

Our work constitutes one phase of quality's management:

It is the definition and implementation of software's quality management system. During this phase, it is necessary to define all what concerns the concept of software's quality.

During our project, we developed a simple tool of software's quality management; allowing managers to structure the software's development process. We have adapted in our study the CMM as model of process's value.

Key words: software's quality, certification, process's value, CMM, UML.

Introduction générale

Introduction générale

Ses dernières années, les technologies de l'information sont devenues une partie prenante de notre vie, nous les trouvons dans tous les secteurs d'activités dans le monde. Malheureusement, l'augmentation des problèmes liés à la sécurité informatique a entaché son émergence, des pertes économiques dues à des interruptions de services (bogue 2000) et des interruptions dans les systèmes se chiffrent en millions de dollars.

Quand la sécurité, les biens et le confort de l'utilisateur sont en jeu, celui-ci exprime son intérêt par des exigences de qualité de plus en plus fortes. De ce fait, il est plus avantageux pour les organismes présentant dans le développement logiciel de définir et d'implémenter des programmes qualité, visant à l'amélioration des processus liés au développement logiciel.

Des référentiels font office de guide pour la sécurité, tel que COBIT, dont l'un de ses processus est destiné à la gestion de la qualité. Pour mettre en oeuvre cette dernière, des référentiels qualité existent pour réponde aux besoins organisationnels des entreprises.

Ces référentiels sont des normes et des modèles. La famille de la norme ISO9000 constitue des lignes directrices permettant la standardisation des pratiques dans une organisation.

Le modèle propre au développement logiciel le plus répandu est le Capability Maturity Model (CMM) qui fournit un cheminement logique d'amélioration de processus.

Le travail que nous allons présente rentre dans un projet lancé par le Centre de Recherche sur Informatique Scientifique et Technique (CERIST), qui consiste à développer un système de sécurité basé sur le modèle COBIT. Dans ce cadre est intégré notre projet.

Notre travail consiste à développer un outil de gestion de la qualité logiciel, afin de permettre au manager d'organiser et maîtriser ses processus logiciels.

Afin d'atteindre notre principale but, nous avons organisé notre mémoire comme suit :

Le premier chapitre sera consacré à l'étude du concept qualité, ainsi que les différentes politiques qualité existantes (maîtrise, contrôle, assurance et management).

Dans le deuxième chapitre, une brève présentation des différentes démarches qualité, ainsi qu'une comparaison entre les plus connus (ISO 9000 et CMM).

Dans le troisième chapitre, nous présenterons la démarche choisie (évaluation de capacité CMM), regroupant son historique, ses concepts de base.

Dans le quatrième chapitre, nous décrivons le système de gestion de la qualité logiciel basé sur CMM. Une brève présentation du langage de modélisation UML, afin de l'utiliser pour le développement de notre outil.

Dans le cinquième chapitre, nous aborderons le développement de l'outil, suivant un cycle de vie en V.

Pour en finir, une conclusion regroupera l'importance de notre outil pour la gestion de la qualité, ainsi que les perspectives envisagées.

1. La problématique

Ce présent travail consiste à résoudre un certain nombre de problèmes, qui sont résumés en :

- L'absence des notions du management de la qualité dans les sociétés.
- La qualité du logiciel est traitée du point de vue économique et non organisationnel.
- Les sociétés ne disposent pas de cadres structurés dans lequel le développement logiciel doit progresser de manière hiérarchique et effective.

2. Les objectifs

Face à ces différents problèmes, nous avons mis en œuvre notre outil, dont l'objectif est d'apporter des solutions pertinentes aux problèmes énoncés.

Notre objectif est de développer un outil de gestion de la qualité du logiciel, simple et convivial, permettant une utilisation facile. Cet outil offre les fonctionnalités suivantes :

- Notre système contient tous les concepts liés à la qualité du logiciel en se basant sur le modèle CMM.
- Il maintient une compréhension des exigences des clients.
- Il permet aux managers qualités de faire un bon suivi du projet logiciel.
- Il permet de découvrir les problèmes à temps dans le cycle de développement.
- C'est un outil qui permet d'avoir des logiciels de qualité ; des logiciels qui offrent une bonne sécurité des technologies d'informations.

CHAPITRE I

Concept qualité

1. Introduction

La qualité est aujourd'hui un enjeu clé de la survie et du développement des entreprises. Afin de la gérer, l'entreprise doit se doter de politiques qualité efficaces (contrôle qualité, assurance qualité, gestion de la qualité, etc.). L'ensemble des moyens utilisés pour la mise en application d'une politique qualité constitue, pour une entreprise donnée, son système qualité. Ce dernier est mis en place par l'application de différentes démarches qualité.

Dans ce chapitre, on va définir les divers concepts liés à la qualité ainsi les différentes politiques qualité existantes.

2. Concept qualité

▪ **Définition 1 :** Qualité vient du latin *qualitas* de *qualis* qui signifie notamment la manière d'être, bonne ou mauvaise, de quelque chose, état caractéristique, supériorité, excellence en quelque chose [Larousse 01].

▪ **Définition 2 :** Qualité est un ensemble des traits et des caractéristiques d'un produit ou d'un service qui leur confèrent l'aptitude à satisfaire des besoins exprimés ou implicites (ISO (1) 8402) [PINET 02].

▪ **Définition 3 :** Qualité est l'aptitude d'un ensemble de caractéristiques intrinsèques d'un produit, d'un système ou d'un processus à satisfaire les exigences des clients et autres parties intéressées (ISO 9000 (version2000)) [PINET 02].

2.1. Evolution du concept qualité

La figure 1-1 montre l'évolution du concept qualité. Il met en évidence les trois grandes étapes qui sont:

▪ le début formel des travaux liés à la qualité et l'introduction du concept de **contrôle qualité** à partir des travaux de l'Américain Walter Shewhart dans le cadre de la Western Electric Company;

▪ **l'assurance qualité**, qui permet au client d'obtenir, de la part de son fournisseur, une formulation écrite des mesures prises pour assurer la livraison d'un produit correspondant aux spécifications convenues;

(1). **En anglais** : International Organisation for Standardisation

▪ la vision moderne de la gestion de la qualité, le **management total de qualité (MTQ (2))**, issu en partie des travaux liés à la systématique. Il privilégie l'assignation des objectifs de qualité au système représentant l'organisme dans sa totalité [Sunier 95].

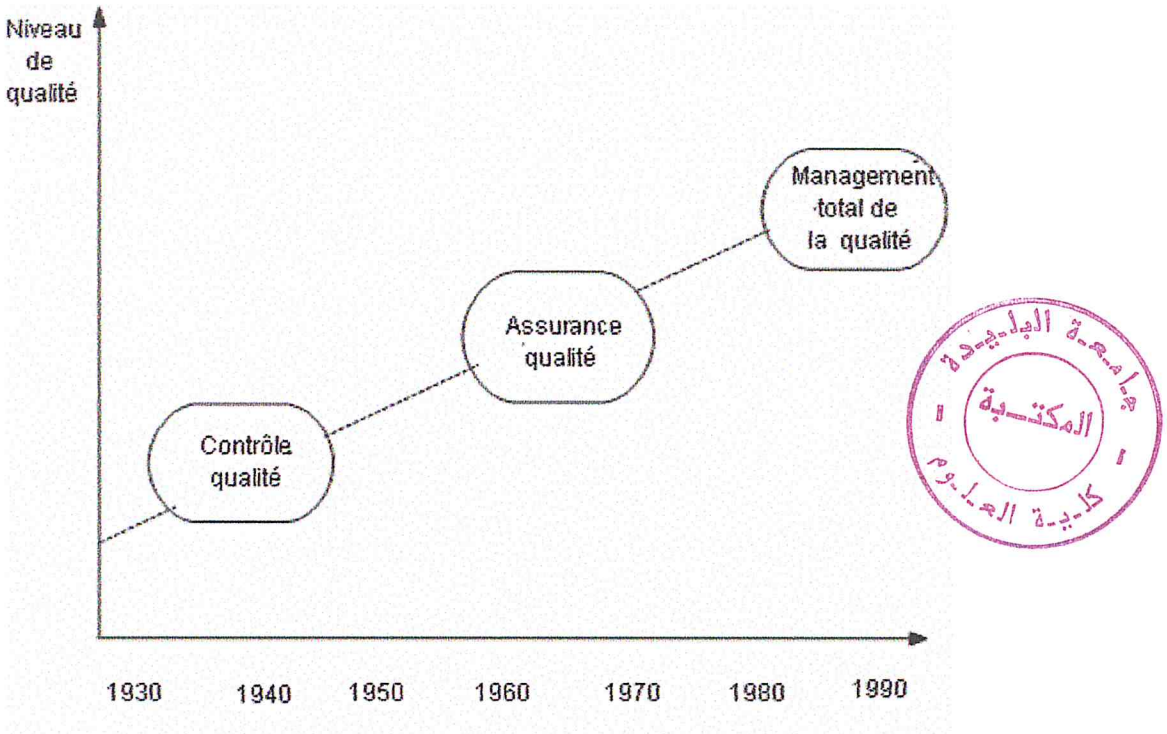


Figure 1-1 : Evolution du concept qualité [Sunier 95]

3. Management de la qualité

Le management de la qualité est l'ensemble des activités coordonnées pour orienter et contrôler un organisme en matière de qualité. Il inclut l'établissement d'une politique qualité et d'objectifs qualité.

Le management comprend quatre actions fondamentales : planifier, maîtriser, prouver et améliorer (figure 1-2) [PINET 02].

Le management de la qualité implique de :

- Choisir ou construire un référentiel.
- Le mettre en place et le pratiquer.
- Contrôler l'utilisation qui en est faite.
- Mesurer les écarts.
- Rechercher à s'améliorer en permanence.

(2). **En anglais** : Management Total Quality

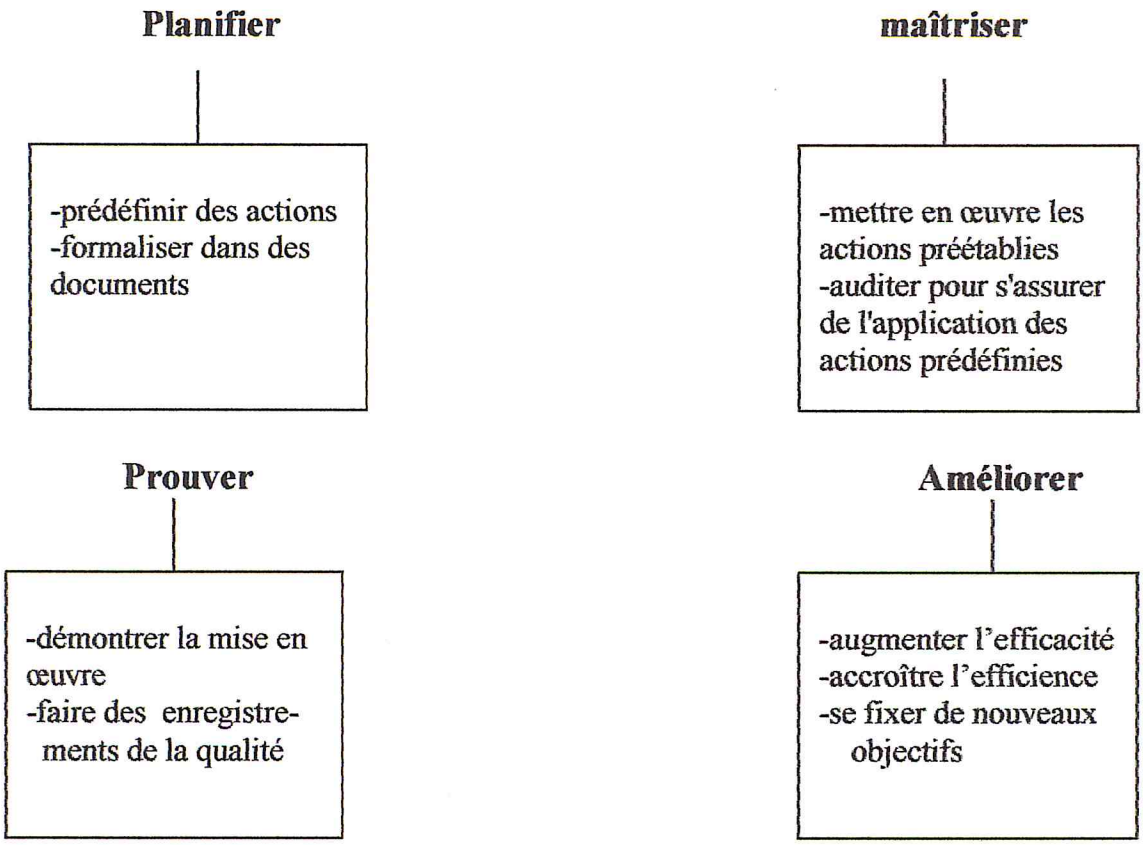


Figure 1-2 : Les maîtres mots [PINET 02]

- **Roue de Deming**

Les quatre concepts de management de la qualité ne sont pas récents, on les trouve avec des terminologies similaires, dans le fameux cycle PDCA (plan, do, check, act) (figure 1-3) imaginé par Deming comportant quatre étapes : prévoir, faire, vérifier et corriger.

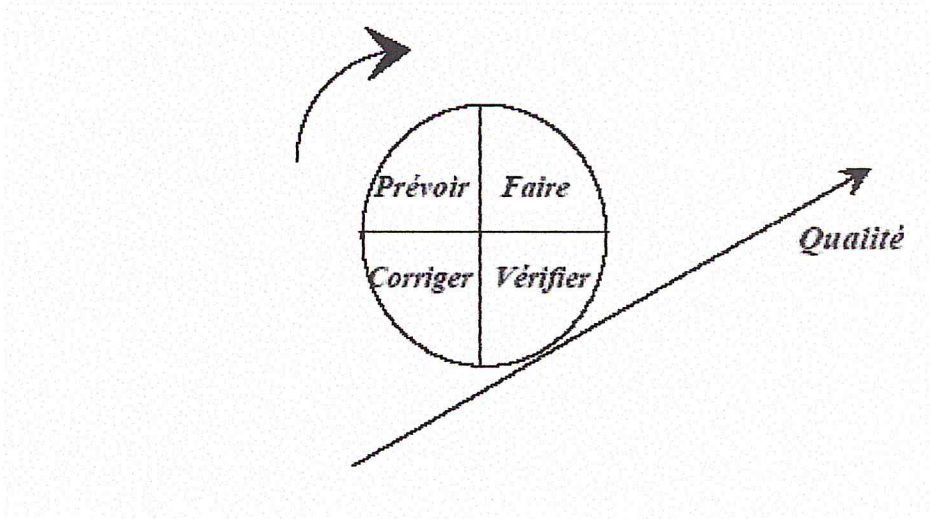


Figure 1-3 : la roue de Deming [Fleurquin 96]

L'étape « **Prévoir** » consiste à :

- Identifier les objectifs possibles d'amélioration ;
- Classer par priorités ces objectifs ;
- Etablir le plan d'action jugé nécessaire pour atteindre les objectifs choisis.

L'étape « **Faire** » consiste à :

- Déterminer les moyens nécessaires à la réalisation des actions ;
- Allouer les moyens ;
- Déclencher les actions ;

L'étape « **Vérifier** » consiste à :

- Mesurer les résultats ;
- Vérifier que les actions mises en place sont efficaces et atteignent l'objectif défini.

L'étape « **Corriger** » consiste à :

- Comparer les résultats obtenus aux objectifs ;
- En fonction des écarts et de l'expérience acquise, établir un plan d'actions correctif et déclencher les actions.

4. Politique qualité

▪ **Définition 1**:la politique qualité est l'ensemble d'orientations et d'objectifs généraux d'un organisme concernant la qualité, tels qu'ils sont exprimés formellement par la direction au plus haut niveau (ISO 8402).

▪ **Définition 2**: la politique qualité doit être définie formellement par la direction de plus haut niveau. elle doit être un élément de la politique générale de l'organisme; elle servira de cadre à toutes les actions entreprises dans l'organisme (ISO 9000 v2000).

Une politique qualité doit fixer:

- la réglementation sur laquelle elle s'appuie (par ex: ISO 9001);
- l'attitude adoptée par chaque collaborateur face au concept de la qualité;
- les paramètres de contrôle déterminants;
- le niveau d'exigence de la qualité.

Nous allons maintenant décrire les différentes politiques qualité (figure 1-4) existantes.

4.1. Management de la qualité

Le management de la qualité est l'ensemble des activités de la fonction générale du management. Il détermine la politique qualité, les objectifs et les responsabilités.

Ces derniers sont mis en oeuvre par des moyens tels que la planification de la qualité, la maîtrise de la qualité et l'amélioration de la qualité dans le cadre du système qualité (ISO 9000).

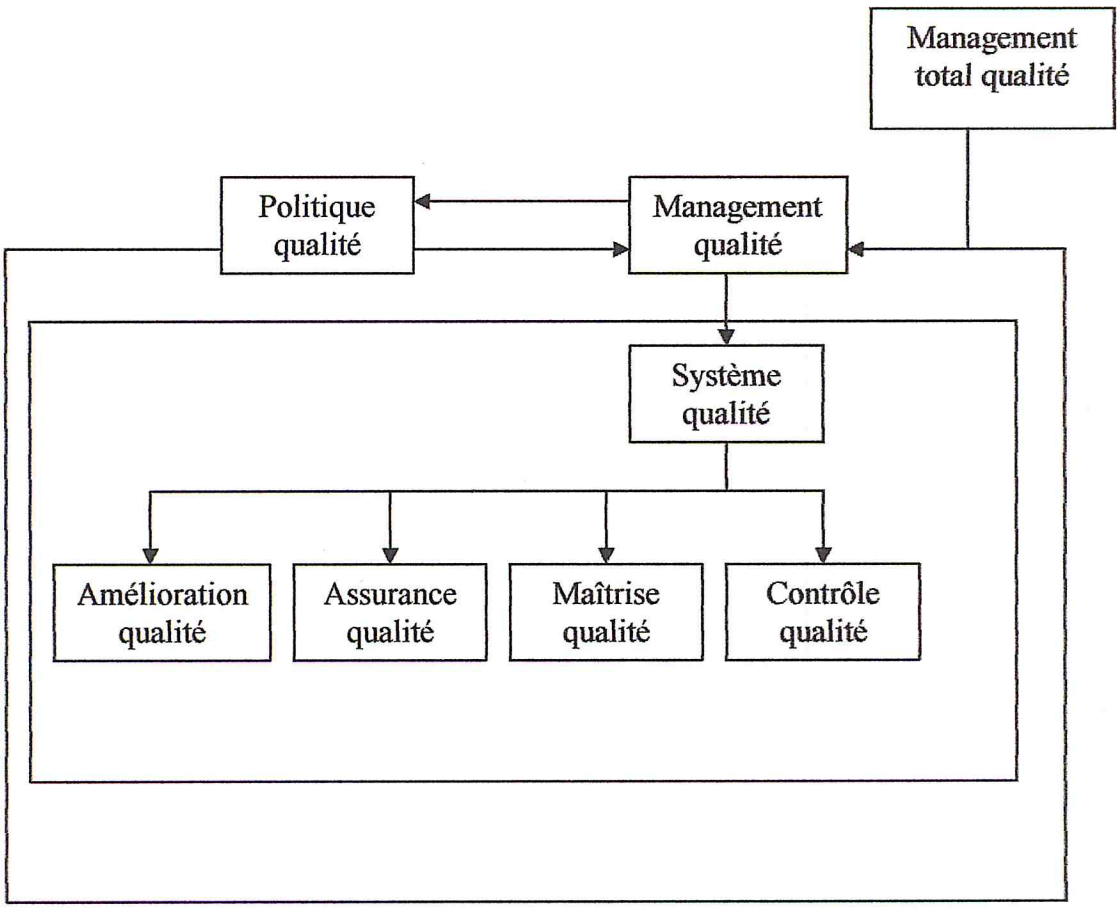


Figure 1-4 : les différentes politiques qualité

4.2. Contrôle qualité

Le contrôle qualité vise à évaluer la qualité des produits, à identifier leurs éventuels défauts et à les corriger si nécessaire. Le processus de développement est reconnu comme une entité non définie. Le processus est considéré comme satisfaisant mais on sait qu'il présente des dysfonctionnements sur lesquels nous n'avons pas prise (génération de défauts). La qualité du produit n'est donc considérée qu'une fois le produit crée .Cette approche s'est révélée peu satisfaisante et très coûteuse (ISO 9000) [Lemoine 02].

4.3. Maîtrise qualité

La maîtrise qualité est l'ensemble de techniques et d'activités à caractère opérationnel utilisées pour satisfaire aux exigences pour la qualité.

La maîtrise qualité doit passer par :

- La définition d'un cycle de vie, suite d'étapes bien identifiées (bien et/ou mal) maîtrisées.
- La définition d'un processus, instance d'un processus standard qui est une définition opérationnelle du processus de base guidant la mise sur pied d'un processus commun à tous les projets logiciel entrepris par une organisation.
- L'Amélioration de l'instance d'un processus standard choisi et maîtrisé.
- La maîtrise de la qualité comprend des techniques et des activités à caractère opérationnel .Ils ont pour but à la fois de piloter un processus et d'éliminer les causes de fonctionnement non satisfaisant à toutes les phases de la boucle de la qualité en vue d'atteindre la meilleure efficacité économique (ISO 8402)

4.4. Assurance qualité

L'assurance qualité est l'ensemble des activités préétablies et systématiquement mises en oeuvre dans le cadre du système qualité, et démontrées en tant que besoin, pour donner la confiance appropriée en ce qu'une entité satisfera aux exigences pour la qualité (ISO 8402).

L'assurance de la qualité vise à la fois des objectifs internes et externes:

- Au sein de l'organisme, elle sert à donner confiance à la direction;
- Dans des conditions contractuelles ou autres, elle sert à donner confiance aux clients ou autres.

Dans l'assurance qualité, il faut suivre ses règles :

- Ecrire ce que vous faites ;
- Faire ce qui est écrit ;
- Prouver que vous le faite.

4.5. Amélioration qualité

L'amélioration qualité est l'ensemble d'actions entreprises dans tout l'organisme, en vue d'accroître l'efficacité et le rendement des activités et des processus pour apporter des avantages accrus à la fois à l'organisme et à ses clients.

L'amélioration de la qualité se fait essentiellement au travers des actions d'apprentissage tel que la boucle de qualité (on passe de détection/correction vers prévention des fautes) (ISO 8402).

4.6. Management total de la qualité

Le management total de la qualité est le mode de management d'un organisme, centré sur la qualité, basé sur la participation de tous ses membres et visant au succès à long terme par la satisfaction du client, et à des avantages pour tous les membres de l'organisme et pour la société (ISO8402).

Le management total de la qualité, et sa définition selon ISO, met en évidence certaines phrases-clés :

- la qualité est l'affaire de tous;
- la qualité est un état d'esprit;
- la direction est responsable à 100% des problèmes de qualité et de leur persistance.

La politique qualité est mise en oeuvre au sein d'une entreprise par un **système qualité**.

5. Système qualité

L'ensemble de la structure organisationnelle, des responsabilités, des procédures, des procédés et des ressources mises en oeuvre au sein d'une entreprise pour gérer la qualité.

Le but d'un système qualité est d'atteindre le plus précisément les objectifs de qualité définis dans le cadre de la politique qualité (ISO 9000).

6. Documents requis pour le système qualité

6.1. Introduction

L'organisme désirant mettre en oeuvre un système qualité doit le définir par écrit en référençant l'ensemble des règles et procédures à appliquer.

Cette étape est impérative pour donner la confiance voulue à la direction et aux clients.

6.2. Types de documents

Quelques sortes de documents distincts permettant de décrire le système qualité sont à prendre en compte sont: (Figure 1-5).

- **Manuel qualité (MQ)**

Un manuel qualité est un document décrivant le système de management de la qualité d'un organisme. La forme du manuel varie selon la complexité de l'entreprise. Ce document est à usage strictement interne .il décrit la politique qualité de l'entreprise. Il décrit aussi explicitement l'organisation mise en place pour maîtriser la qualité (ISO 9000).

- **Plan qualité (PQ)**

Un plan qualité est un document spécifiant les éléments du système de management de la qualité et les ressources à appliquer dans un cas spécifique .dans ce document, on va trouver les pratiques qualités relatives à un produit, un service, un contrat ou un projet. Le plan qualité contient la liste des activités qui s'y rapportent et l'attribution des responsabilités correspondantes (ISO 9000).

- **Manuel des procédures**

Une procédure est une manière spécifique d'effectuer une activité ou un processus. Les procédures sont matérialisées par des documents écrits et formalisés. Elles visent à décrire l'organisation mise en place et son mode de fonctionnement.

Une procédure décrit dans l'ordre chronologique l'organisation des activités d'un processus dans le but de satisfaire à un objectif. La description doit préciser <<Quoi>> doit être fait <<Qui>> doit le faire, <<Quand>>, <<Ou>> et <<comment>> il faut le faire (règle QQQOC).les procédures sont répertoriées dans un manuel des procédures(ISO 9000).

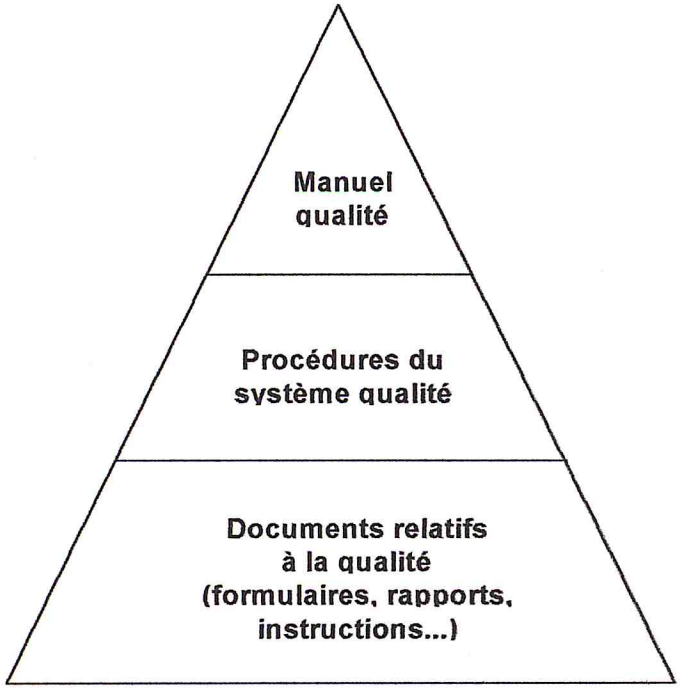


Figure 1-5 : Hiérarchie des documents qualité [Sunier 95]

• Fiches /formulaires

Les fiches sont des documents spécifiques peu volumineux, elles tiennent généralement sur une seule feuille de papier. L'objectif de ces fiches est de véhiculer et de conserver des informations de nature homogène. Exemples : fiche de tâches, fiche d'anomalies, formulaires de saisie, etc. les fiches sont des bons moyens de saisie et de conservation d'informations. Le rassemblement de plusieurs fiches dans un même endroit constitue un fichier. L'accès aux informations sera facile et rapide [PINET 02].

• Les non-conformités

Une non-conformité est une non satisfaction à une exigence .cette dernière est un besoin ou une attente formulé dans un document. Lors de la vérification (inspection, revue, validation), le vérificateur compare ce qu'il observe sur le terrain par rapport au référentiel défini (ISO 9000). Après l'examen et le constat du vérificateur, deux cas se présentent :

- Le vérificateur n'a pas trouver un écart par rapport au référentiel, il y a une conformité entre le prescrit et le réalisé.
- Le vérificateur a trouvé un écart par rapport au référentiel, il n'y a pas une adéquation entre le prescrit et le réalisé. Suivant l'importance de l'écart, le vérificateur qualifiera l'écart de non-conformité ou de seulement de remarque.

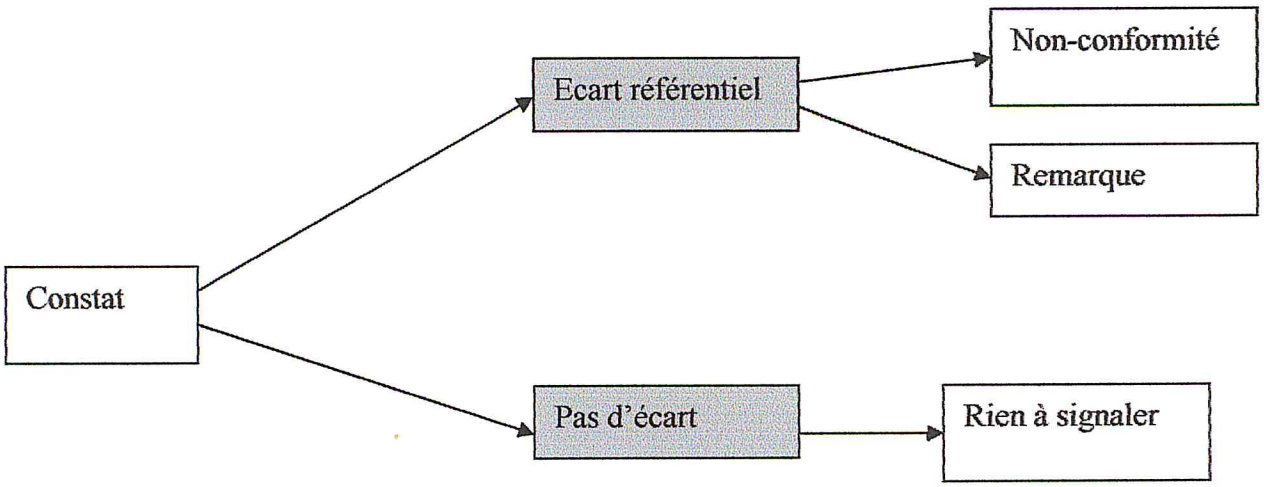


Figure 1-6 : Les constats [PINET 02]

7. conclusion

La qualité est l'affaire de tous. Sa responsabilité est placée à tous les niveaux sur les gens qui la maîtrisent. Ils doivent donc être formés en conséquence.

Après que nous avons étudié la qualité et tous ce qui est lié à ce concept; dans le chapitre suivant, nous allons entamer le concept de la qualité du logiciel, ainsi que les différentes démarches qualité logiciel.

CHAPITRE II

Qualité du logiciel

1. Introduction

Dans ce chapitre, nous présentons les concepts liés a la qualité du produit logiciel : les critères et les démarches qualité.

Avant tout, nous devons définir le logiciel, la qualité logiciel, ensuite nous survolerons les critères du produit logiciel et terminer par une vue sur les démarches qualité logiciel.

Les critères du produit logiciel sont l'ensemble des caractéristiques que le logiciel doit satisfaire.

Une démarche qualité est la mise en place d'une politique qualité appropriée aux besoins d'une entreprise.

2. Logiciel

- **Définition1 :**est une totalité ou un élément constitutif de l'ensemble complet des programmes informatiques, des procédures, ainsi que la documentation et les données offertes devant êtres livrés aux clients ou à l'utilisateur final.
- **Définition2 :** est un ensemble des programmes, des procédures, des règles et toute documentation relatifs au fonctionnement d'un ensemble des données [PINET 97].

▪ Qualité du logiciel

La qualité du logiciel est l'aptitude d'un logiciel à satisfaire les besoins exprimés ou potentiels des utilisateurs .Elle se mesure par rapport à un référentiel défini.

La qualité du logiciel se base sur 3 axes :

- Les caractéristiques du produit.
- Le processus.
- La formation.

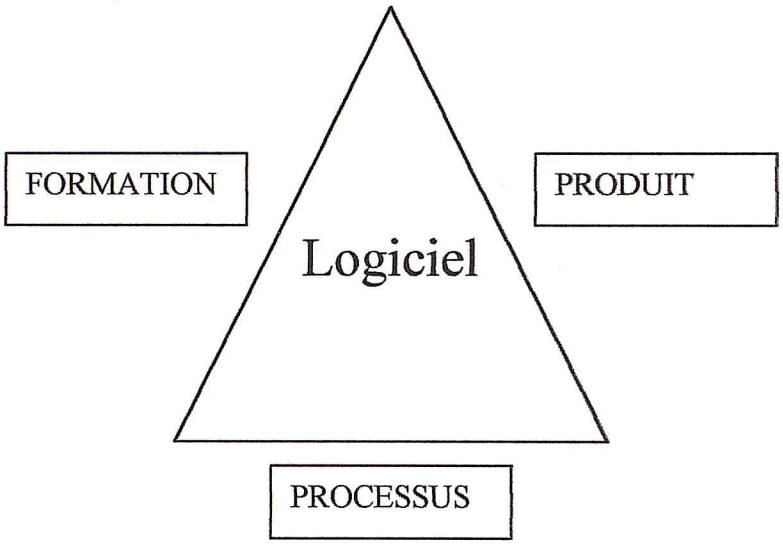


Figure 2-1 : Le triangle de la qualité

3. Caractéristiques de la qualité du logiciel

Une Caractéristique 'est un trait distinctif qui peut être physique, comportemental, temporel, ergonomique ou fonctionnel [PINET 02].

Chaque produit logiciel est caractérisé par des attributs externes et des attributs internes.

❖ **Les attributs internes :** Ils peuvent être mesurés à partir des caractéristiques du logiciel. Ils correspondent à la vision du réalisateur dont l'intérêt se situe au niveau de la fabrication du produit et de son procédé.

Exemple : La productivité, la performance d'une ressource, la complexité d'un programme.

❖ **Les attributs externes :** Ils ne peuvent être mesurés qu'à partir des interactions entre les entités et leur environnement. Ils reflètent le point de vue de l'utilisateur, qui souhaite voir ses besoins satisfaits.

Exemple : La maintenabilité d'un programme ou d'un modèle de conception, et l'ergonomie d'un logiciel.

A chaque attribut externe correspond un facteur se décomposant en un ou plusieurs critères auxquels une valeur numérique ou booléenne calculée par une métrique sera affectée selon la correspondance suivante :

Attributs externes	⇨	Facteurs.
Attributs internes	⇨	Critères.
Mesure quantitative des attributs internes	⇨	Métriques.

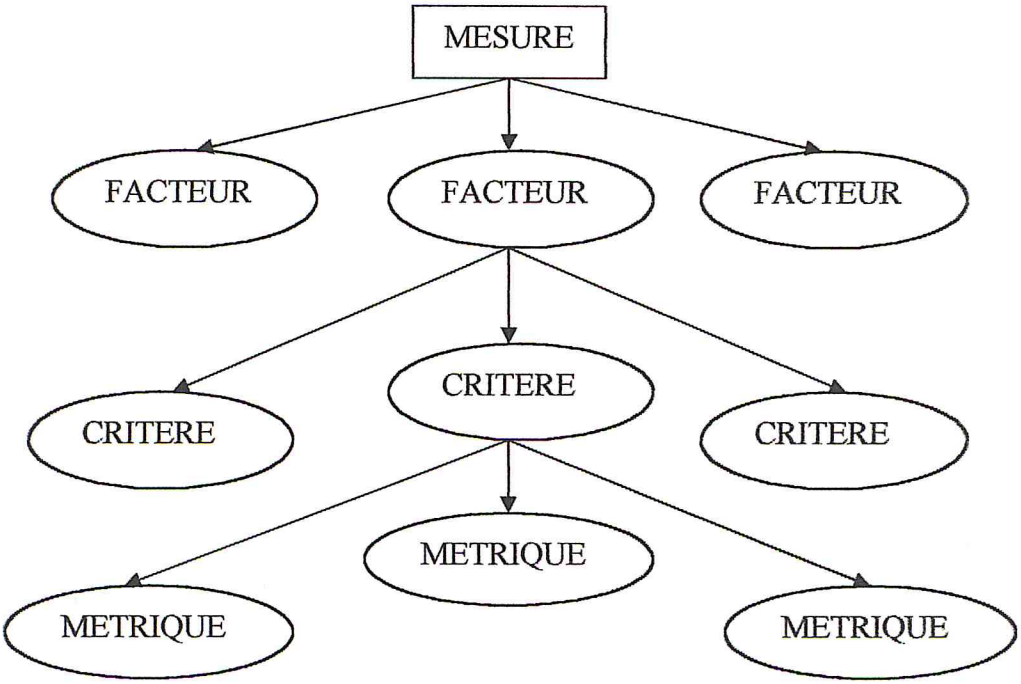


Figure 2-2 : Les facteurs et critères

4. La norme ISO 9126

La qualité du produit est certainement la notion la plus intuitive à comprendre. La norme ISO 9126 définit et décrit une série de caractéristiques qualité d'un produit logiciel (caractéristiques internes et externes, caractéristiques à l'utilisation) qui peuvent être utilisées pour spécifier les exigences fonctionnelles et non fonctionnelles des clients et des utilisateurs.

Chaque caractéristique est détaillée en sous-caractéristique, et pour chacune d'elle, la norme propose une série de mesures à mettre en place pour évaluer la conformité du produit développé par rapport aux exigences formulées [Sunier 95].

4.1. Capacité fonctionnelle

Ensemble d'attributs portant sur l'existence d'un ensemble de fonctions et leurs propriétés données. Les fonctions sont-elles celles qui satisfont aux besoins exprimés ou implicites.

- **Aptitude :** Attributs du logiciel portant sur la présence et l'adéquation d'une série de fonctions pour des tâches données.
- **Exactitude :** Attributs du logiciel portant sur la fourniture de résultats ou d'effets justes ou convenus.
- **Interopérabilité :** Attributs du logiciel portant sur sa capacité à interagir avec des systèmes donnés.
- **Conformité réglementaire :** Attributs du logiciel selon lesquels il respecte l'application de normes, des conventions, des réglementations ou des prescriptions similaires.
- **Sécurité :** Attributs du logiciel portant sur son aptitude à empêcher tout accès non autorisé (accidentel ou délibéré) aux programmes et donnés.

4.2. Fiabilité

Ensemble d'attributs portant sur l'aptitude du logiciel à maintenir son niveau de services dans des conditions précises et pendant une période donnée.

- **Maturité :** Attributs du logiciel portant sur la fréquence des défaillances dues à des défauts du logiciel.
- **Tolérance aux fautes :** Attributs du logiciel portant sur son aptitude à maintenir un niveau de service donné en cas de défaut du logiciel ou de violation de son interface.
- **Possibilité de récupération :** Attributs du logiciel portant sur ses capacités de rétablir son niveau de service et de restaurer les informations directement affectées en cas de défaillances, et sur le temps et l'effort nécessaire pour le faire.

4.3. Facilité d'utilisation

Ensemble d'attributs portant sur l'effort nécessaire pour l'utilisation et sur l'évaluation individuelle de cette utilisation par un ensemble défini ou implicite d'utilisateurs.

- **Facilité de compréhension :** Attributs du logiciel portant sur l'effort que doit faire l'utilisateur pour reconnaître la logique et sa mise en œuvre.
- **Facilité d'apprentissage :** Attributs du logiciel portant sur l'effort que doit faire l'utilisateur pour apprendre son application.

❖ **Facilité d'exploitation :** Attributs du logiciel portant sur l'effort que doit faire l'utilisateur pour l'exploiter et contrôler son exploitation.

4.4. Rendement

Ensemble d'attributs portant sur le rapport existant entre le niveau de service d'un logiciel et la quantité de ressources utilisées, dans des conditions déterminées.

❖ **Comportement vis-à-vis du temps :** Attributs du logiciel portant sur les temps de réponse et de traitement ainsi que sur les débits lors de l'exécution de sa fonction.

❖ **Comportement vis-à-vis des ressources :** Attributs du logiciel portant sur la quantité de ressources utilisées et sur la durée de leur utilisation lorsqu'il exécute sa fonction.

4.5. Maintenabilité

Ensemble d'attributs portant sur l'effort nécessaire pour faire des modifications données.

❖ **Facilité d'analyse :** Attributs du logiciel portant sur l'effort nécessaire pour diagnostiquer les déficiences ou les causes de défaillance, ou pour identifier les parties à modifier.

❖ **Facilité de modification :** Attributs du logiciel portant sur l'effort nécessaire pour modifier, remédier aux défauts ou changer d'environnement.

❖ **Stabilité :** Attributs du logiciel portant sur le risque des effets inattendus des modifications.

❖ **Facilité de test :** Attributs du logiciel portant sur l'effort nécessaire pour valider le logiciel modifié.

4.6. Portabilité

Ensemble d'attributs portant sur l'aptitude du logiciel à être transféré d'un environnement à l'autre.

❖ **facilité d'adaptation :** Attributs du logiciel portant sur la possibilité de son adaptation à différents environnements donnés sans que l'on ait recours à d'autres actions ou moyens que ceux prévus à cet effet pour le logiciel considéré.

❖ **Facilité d'installation :** Attributs du logiciel portant sur l'effort nécessaire pour installer le logiciel dans un environnement donné.

❖ **Conformité relative aux règles de portabilité :** Attributs du logiciel permettant à celui-ci de se conformer aux normes ou conventions ayant trait à la portabilité.

❖ **Interchangeabilité :** Attributs du logiciel portant sur la possibilité et l'effort pour l'utiliser à la place d'un autre logiciel donné dans le même environnement.

5. Processus logiciel

Un Processus logiciel est un ensemble d'opérations ou d'activités réalisées par des acteurs avec des moyens et selon des références en vue d'une finalité [DIRE 01].

La complexité du produit logiciel a permet la décomposition des processus de développement en processus élémentaires :

- **Spécifications** : "Qu'est-ce qu'on veut faire?"

✓ But : définir l'ensemble des caractéristiques externes (côté utilisateur) et internes (côté conception).

✓ Documents : le dossier de spécifications (reprise et reformulation détaillée du cahier des charges), le cahier de recettes (= description des tests de validation), une esquisse du manuel utilisateur.

✓ Contrôles : la revue de spécifications qui doit permettre de chercher l'erreur(s).

- **Conception générale** : "Comment faire?"

✓ But : déterminer l'architecture générale du logiciel et préparation des tests d'intégration, découpe en tâches, fonctionnalités regroupées en modules, interfaces de communication entre les modules, références au dossier de spécification.

✓ Documents : le dossier de conception générale, le cahier d'intégration (description des tests d'intégration), la révision du manuel utilisateur.

✓ Contrôles : la revue de conception générale qui présente les différents choix possibles: détermination des risques, des moyens à mettre en place, avantages et inconvénients.

- **Conception détaillée** : "Comment faire exactement?"

✓ But : s'assurer que le produit peut être testé et en indiquer les moyens, détailler les algorithmes et structures de données utilisés ou développés, décrire les autres solutions possibles mais non retenues, justifier le choix de la solution.

✓ Documents : le dossier de conception détaillée, le dossier de définition des algorithmes, des structures de données, des interfaces..., la procédure de fabrication, le dossier justificatif des choix.

✓ Contrôles : la revue de conception détaillée

• **Réalisation** :

✓ But : coder le programme.

✓ Documents : les sources du programme.

✓ Contrôles : la revue de codage qui vérifie de la présence et de la pertinence des commentaires, du respect des règles de codage.

• **Tests unitaires** : "Est-ce que les modules sont corrects?"

✓ But : s'assurer de la correspondance entre la réalisation et la documentation de définition (au niveau du module), faire des tests aux limites (mémoire, arithmétique, performances…).

✓ Documents : le cahier de tests unitaires complet.

• **Tests d'intégration** : "Est-ce un bon produit?"

✓ But : s'assurer de la conformité des interfaces vis-à-vis de la documentation de conception générale.

✓ Documents : le cahier de tests d'intégration complets.

• **Validation :** "Est-ce le bon produit?"

✓ But : consigner les résultats de tests avec le client

6. Démarches qualité

Une démarche qualité a pour objectif la mise en place d'une politique qualité appropriée aux besoins d'une entreprise.

Le principe consiste à décrire les moyens à mettre en œuvre et les étapes à suivre pour mettre en place un système qualité (si aucun n'existe) ou pour améliorer un système qualité existant.

Nous avons opté pour les démarches basées sur les processus qui permettent de contrôler et d'améliorer des processus du développement logiciel.

Les démarches de type certification ISO 9000(norme) et évaluation du processus (modèle) offrent des référentiels génériques utilisables par la grande majorité des entreprises. Ces démarches sont en effet non spécifiques à des domaines d'application particuliers, ni à certains types d'entreprise, ni à des niveaux technologiques particuliers. Ils englobent les pratiques reconnues comme pertinentes pour le développement de logiciels.

- La norme est une exigence obligatoire utilisée et appliquée pour établir une démarche uniforme et disciplinée de développement logiciel [BOURNICHE 01].

Les normes sont des accords documentés contenant des spécifications techniques ou autres critères précis destinés à être utilisés systématiquement en tant que règles, lignes directrices ou définitions de caractéristiques pour assurer que des matériaux, produits, processus et services sont aptes à leur emploi.

Exemple : le format des cartes de crédit, des cartes à prépaiement téléphonique et des cartes dites « intelligentes « que l'on retrouve partout est dérivé d'une Norme internationale ISO. Le fait d'adhérer à la norme qui définit des caractéristiques telles que l'épaisseur optimale (0,76 mm) signifie que les cartes pourront être utilisées dans le monde entier.

- Le modèle est une abstraction de la réalité qui, pour un domaine, est prise comme une représentation d'une classe de phénomènes plus ou moins habilement dégagés de leur contexte par un observateur, pour servir de support à l'investigation ou à la communication.

6.1. Certification ISO 9000

La certification est une démarche qui consiste à obtenir un certificat, véritable diplôme qualifié. La norme ISO est un document normatif, élaboré selon des procédures consensuelles, approuvé par les membres de l'ISO et les membres du comité responsable [Raisson 99].

6.1.1. Normes fondamentales d'ISO 9000

La famille ISO 9000 comprend toutes les normes internationales générales relatives à la qualité Elle fixe les exigences en matière du système qualité, lorsque :

- Des performances sont demandées et qu'elles doivent être établies ;
- La démonstration de la capacité du fournisseur à réaliser un produit conforme est nécessaire

Le schéma proposé ci-dessous met en évidence la hiérarchie des 3 modèles de la série ISO 9000. La norme ISO 9001 étant la plus contraignante et la 9003 n'étant à appliquer que pour la négociation des produits.

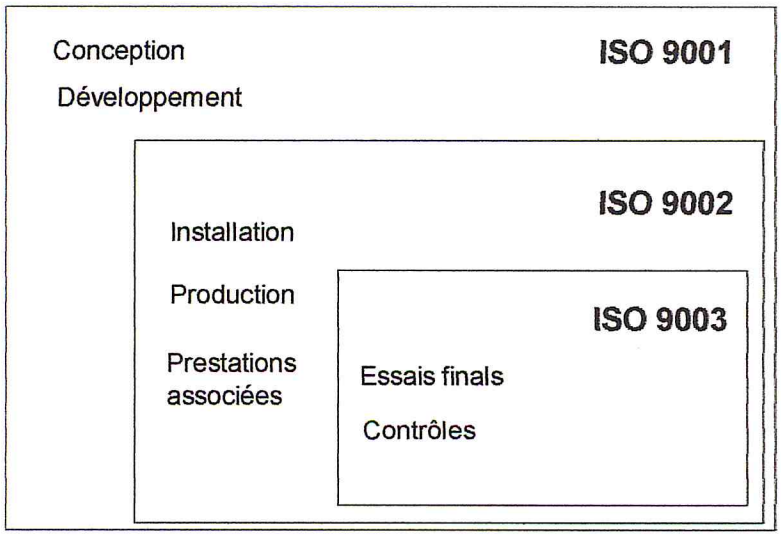


Figure 2-3 : Les normes fondamentales d'ISO 9000

❖ **Norme ISO 9001** : est à utiliser lorsque la conformité à des exigences spécifiées est à assurer par le fournisseur pendant la conception, le développement, la production, l'installation et les prestations associées.

❖ **Norme ISO 9002** : est à utiliser lorsque la conformité à des exigences spécifiées est à assurer par le fournisseur pendant la production, l'installation et les prestations associées.
Il s'agit d'un sous-ensemble de la norme ISO 9001; elle est amputée des aspects ayant trait à la conception et au développement.

❖ **Norme ISO 9003** : est à utiliser lorsque la conformité à des exigences spécifiées est à assurer par le fournisseur uniquement lors des contrôles et essais finals.
ISO 90003 se présente comme un guide permettant de mettre en oeuvre un système de management de la qualité. Ce dernier est prescrit par ISO 9001:2000 dans le cadre de l'acquisition, la fourniture, le développement ou la maintenance de logiciels. Pour ce faire, ISO 90003 propose un ensemble de processus spécifiques à ce type d'organisation et qui sont absents de la 9001v2000.

Les démarches de mise en conformité ISO 9001 dans l'informatique sont récentes. Les entreprises du logiciel n'ont véritablement commencé à entamer des démarches de certification que depuis 6 ou 7 ans. Or, une mise en conformité peut souvent demander plusieurs années.

La norme ISO 9000 a pour but d'assurer :

- La confiance du client,
- Répondre exactement à ses besoins et attentes,
- Améliorer la performance,
- Obtenir une meilleure rentabilité de l'entreprise
- Mais aussi améliorer l'accès au marché.

6.2. Evaluation des processus

Nous allons maintenant étudier les démarches basées sur les modèles d'évaluation de processus. Ces modèles permettent :

- La détermination de la capacité du processus ;
- L'amélioration du processus.

Pour chaque pratique d'un modèle d'évaluation de processus, il peut être associé un niveau de maturité. Ce niveau de maturité peut être calculé ou attribué d'origine, il indique le degré de maîtrise des activités liées au développement logiciel associé à la pratique étudiée.

Nous présenterons quelques modèles d'amélioration de processus les plus connues (CMM, et ISO/SPICE, etc.).

6.2.1. Présentation de quelques modèles

♦ Le modèle CMM

CMM (Capability Maturity Model) "Modèle d'évolution des capacités logiciel" est un modèle conçu dans les années 80, par le SEI (Software Engineering Institute) en collaboration avec le département américain de la défense (DoD), pour évaluer et améliorer la maturité des processus logiciels liés à la défense[JEHANNO 02].

CMM se présente comme un recueil de processus et de bonnes pratiques de gestion et d'ingénierie à mettre en oeuvre dans les organisations qui développent ou assurent la maintenance de logiciels.

Les processus sont répartis en cinq niveaux de maturité (Initial, Reproductible, Défini, Maîtrisé, d'Optimisation) qu'une organisation va gravir en fonction de la qualité des processus qu'elle a mis en œuvre.

Chaque niveau de maturité à l'exception du niveau 1, est caractérisé par des secteurs clés.

Chaque secteur Clé comporte 5 parties appelées caractéristiques communes.

Ces caractéristiques communes indiquent les pratiques clés qui permettent d'atteindre les objectifs du secteur clé.

♦ Le modèle SPICE

En 1993, l'ISO a lancé le projet SPICE (Software Process Improvement and Capability dEtermination) pour développer une norme internationale dans le domaine de l'évaluation des processus logiciel.

Le projet a été mis en place pour garantir un développement rapide du standard, mener des phases d'essais en sollicitant les opinions et les compétences des experts internationaux et disposer d'une dynamique efficace (management) [BOURNICHE 01].

Cette norme, une synthèse des démarches d'évaluation et d'amélioration de processus logiciel : CMM, SPICE, TRILLIUM, est cohérente avec les normes existantes : ISO 9000, ISO 12207, ISO9126. Elle est applicable à un large domaine d'applications, d'affaires, de tailles d'organisation de projets. SPICE est devenu une norme ISO 15504 en 2002.

ISO SPICE fournit un cadre de référence sur l'évaluation des pratiques permettant de :

- Avoir des processus répétables,
- Déterminer leur pertinence par rapport aux objectifs de l'entreprise,
- Les comparer par rapport à un référentiel,

- Favoriser l'obtention des produits ou des services logiciels ayant un niveau de qualité prédéfini,
- Soutenir une amélioration de la productivité.

➢ **Document SPICE :**

ISO 15504 est un ensemble de 9 documents qui constituent globalement le standard :

1) Introduction aux concepts fondamentaux.
2) Modèle de gestion de l'ingénierie des processus.
3) Processus d'évaluation du niveau d'aptitude.
4) Guide de conduite de l'évaluation.
5) Modèle d'évaluation, guide des indicateurs, outils.
6) Guide pour la qualification des évaluateurs.
7) Guide de mise en œuvre de l'amélioration des processus.
8) Guide de détermination des aptitudes des fournisseurs.
9) Dictionnaire, vocabulaire et terminologie.

➢ **Architecture SPICE :**

Le modèle SPICE est un modèle sur 2 dimensions aptitude et processus.

La dimension aptitude est constituée de 6 niveaux :

❖ Au niveau 0, soit le processus n'est pas réalisé, soit il n'atteint que partiellement son objectif.

❖ Le niveau 1 d'un processus se caractérise par l'atteinte des objectifs du processus qui est dit « réalisé ».

❖ Au niveau 2, le processus est « géré ». Ceci concerne deux aspects : d'une part le management du processus lui-même et d'autre part le management des produits issus du processus.

❖ Un processus de niveau 3 est dit « établi » au niveau de l'organisation. A ce niveau, la mise en œuvre du processus se base sur des pratiques documentées standard.

❖ Le niveau 4 caractérise un processus dont la maîtrise se base sur une approche quantitative : le déroulement du processus est mesuré et ses performances sont « prévisibles ».

❖ Au niveau 5, le processus est en « optimisation » : l'organisation est capable d'améliorer ses processus et de les adapter en fonction des objectifs de l'organisation.

La dimension processus est constituée de 5 catégories :

- **client-fournisseur (CUS)**: ce sont les processus qui ont des impacts directs sur les clients,
- **ingénierie (ENG)** : c'est l'ensemble des processus qui spécifient, implémentent ou assurent la maintenance d'un produit (système et logiciel),
- **management de projets (MAN)** : ceux qui établissent le projet, qui coordonnent et assurent la direction de ses ressources,
- **support (SUP)** : ce sont les processus qui permettent la bonne marche des autres processus du projet et qui leur viennent en aide,
- **organisation (ORG)** : ceux qui déterminent les buts commerciaux de l'entreprise, qui évaluent les processus, les produits et les ressources. Ces évaluations aident l'organisation à atteindre ses buts commerciaux.

Chaque catégorie est divisée en processus, eux-mêmes divisés en pratiques.

Les pratiques sont classées en pratiques de base si elles sont essentielles pour un processus particulier ou en pratiques génériques qui peuvent potentiellement être appliquées à n'importe quel processus.

Pour évaluer le niveau d'aptitude de chacun des processus, le modèle définit deux types d'indicateurs :

- **les indicateurs de réalisation** du processus ou dimension processus : pour chacun des processus, des pratiques de bases sont identifiées ainsi que des produits du travail en entrée et en sortie,
- **les indicateurs d'aptitude** du processus ou dimension d'aptitude : pour chacun des niveaux d'aptitude, des pratiques de management sont identifiées ainsi que des caractéristiques de réalisation, de ressources et d'infrastructure.

6.2.2. Analyse comparative

Après l'étude des deux models CMM et SPICE, on a constaté qu'ils ont en commun les cinq niveaux de maturité : niveau 1 (processus informel), niveau 2 (processus planifié et suivi), niveau 3 (processus bien défini), niveau 4 (processus maîtrisé quantitativement), niveau 5 (amélioration permanente des processus).

Ainsi, nous pouvons distinguer que les deux démarches permettent la détermination de capacité et l'amélioration du processus. Cependant, des différences ont pu apparaîtes :

- Selon la structure, surtout le lien établi entre les niveaux de maturité et les pratique clés :
  - par exemple, CMM associe pour chaque niveau de maturité des secteurs clé qui englobent les pratiques clé qui lui sont propres. dans chaque secteur clé, les pratiques clés sont reparties selon des caractéristiques communes.

  - Contrairement, SPICE sépare de deux dimensions la hiérarchie de maturité et les pratiques, car à chaque catégorie de processus contient ses processus, chaqu'un de ses derniers possède son propre profil de maturité décrivant ses pratiques génériques.

- Et bien d'autres différences présentés par le tableau suivant :

SPICE	CMM
• Une structure bidirectionnelle • Stratégie d'amélioration flexible • Un niveau de maturité pour chaque processus de l'organisation • Résultat d'évaluation complexe • Concentré sur l'évaluation du niveau de maturité.	• Une structure unidimensionnelle • Démarche d'amélioration imposée • Un seul niveau pour toute l'organisation • Résultat d'évaluation facile • Concentré sur l'étude de l'organisme et les changements à effectuer pour passer à un niveau supérieur.

Figure 2-4 : Comparaison CMM/SPICE

6.3. Comparaison entre ISO 9001 et les modèles d'évaluation du processus

Il y a une corrélation entre le modèle ISO 9001 et les modèles d'évaluation de capacité car ils ont le même objectif orienté vers la gestion du processus, mais ils diffèrent quant à leur philosophie.

Voici quelques principales différences entre ces modèles :

- Le modèle ISO 9001 est générique, il est conçu pour un usage plus général (services, matériels, logiciels…). Tandis que les modèles d'évaluation de processus sont conçus pour des organismes qui développent des logiciels, elles intègrent les besoins relatifs à l'informatique.
- Le modèle ISO 9001 est centré sur une politique d'assurance qualité, par contre les modèles d'évaluation de processus ont pour finalité une politique de gestion de qualité.
- Le modèle ISO 9001 fonctionne selon un mode réussi ou échoué (certifié ou non), les programmes d'amélioration sont orientés vers la mise en œuvre de la conformité. Les modèles d'évaluation de processus définissent un niveau de maturité globale (SPICE) ou par processus (CMM). A partir de son niveau de maturité, on peut fixer les programmes d'amélioration progressifs.
- Le modèle ISO 9001 a un niveau d'abstraction plus élevé car il décrit les caractéristiques minimales d'un système qualité, par contre les modèles d'évaluation de processus sont plus détaillés (CMM compte 500 pages), ainsi plusieurs pratiques ne sont pas citées dans les normes ISO 9001 et 9000-3.
- Le modèle ISO 9001 pourra servir de départ pour démarrer un projet d'amélioration de processus (une entreprise du niveau 1 peut être certifiée). De même une entreprise de niveau 2 avec quelques exigences du niveau 3 pourra servir de guide pour la certification.

Les deux démarches, certification et évaluation des processus peuvent se compléter, ses dernières peuvent non seulement atteindre un niveau de maturité (ce qui est un objectif organisationnel) requis pour l'obtention d'un certificat ISO 9001 (label d'argument commercial), mais aussi de poursuivre les programmes d'amélioration de la politique de gestion de la qualité.

6.4. Choix de la démarche

Après l'étude des différentes démarches qualité (norme ISO 9001, modèle CMM, modèle SPICE) et leur analyse comparative, on propose l'utilisation du modèle SW-CMM[4] pour les raisons suivantes :

- Le modèle CMM est conçu spécifiquement pour les organismes réalisant des logiciels, alors il prend en considération les besoins spécifiques de l'informatique.
- Le modèle CMM est orienté organisationnel, il gère les processus d'une entreprise pour atteindre la politique qualité requise.
- Dans le modèle CMM, on trouve les différentes politiques qualité, de la politique maîtrisé-structuré (niveau 2) à la politique gestion de la qualité (niveau 5) passant par la politique maîtrise-défini et documenté (niveau 3) et la politique maîtrise-mesuré (niveau 4). Il respecte aussi l'ordre dans la démarche d'amélioration.
- Si on suit le modèle CMM et si on passe au niveau 2, on peut avec un petit effort être certifié ISO 9001(avoir le label commercial).
- Notre premier objectif est de s'améliorer, la certification n'étant que la cerise sur le gâteau.
- Privilégier l'amélioration progressive suivant le rythme de la société et non la course à la certification.
- Le modèle CMM est reconnu mondialement.
- Le modèle CMM a servi de base de plusieurs autres modèles tel que TRILLIUM, BOOTSTRAP, SPICE.
- Le modèle CMM est moins volumineux que le modèle SPICE (CMM contient 316 pratiques clés, par contre SPICE contient 910 décisions d'évaluation).
- Des firmes comme Motorola, Hewlett Packad, Siemens, et bien d'autres, ont implémenté un processus basé sur SW-CMM.
- L e CMM permet d'établir le niveau du maturité de la façon de faire la sécurité dans une organisation (plus le processus de développement est mature, plus le logiciel est sécurisé).

7. Conclusion

Dans ce chapitre, nous avons présenté le concept qualité logiciel regroupant les caractéristiques du produit logiciel et les différentes démarches qualité.

Nous avons choisi le modèle SW-CMM qui nous permet d'améliorer le processus de développement logiciel ainsi d'être prés des normes internationales.

Dans le chapitre prochain, nous expliquerons en détail ce modèle (SoftWare-Capacity Maturity Model).

CHAPITRE III

Modèle CMM

1. Introduction

Après que nous avons choisi le modèle SW-CMM pour des raisons que nous les avons cité précédemment, Dans ce chapitre, nous allons le présenter d'une manière plus détaillée.

Pour cela, nous définissons les différents principes de ce modèle ainsi sa structure de niveaux, secteurs clés et les différentes pratiques clés.

Nous avons basé sur le document TR-25 version1.1 publié par le SEI.

2. Définition et origine du CMM

Le Modèle d'évolution des capacités (CMM) pour le logiciel est un cadre décrivant les éléments clés d'un processus logiciel efficace. Le CMM décrit les améliorations évolutives jalonnant le cheminement d'un processus improvisé et immature vers un processus mature, discipliné.

Le CMM est un système qualité qui vise à améliorer le processus de développement logiciel. Il permet à une organisation de mesurer son niveau de maturité à l'aide d'un questionnaire de maturité fournit par le SEI et de faire évoluer sa capacité de développement logiciel. Ce modèle d'évaluation et d'évolution des capacités se base sur une grille de maturité hiérarchisée [LABORDE 02].

Le gouvernement américain ayant déclaré le logiciel « industrie stratégique », à partir de 1986, le DoD (la défense américaine) finance le SEI pour développer et diffuser un modèle d'évaluation de la maturité des entreprises dans le domaine du développement du logiciel :

le « Capability Maturity Model : CMM »

D'abord aux Etats-Unis puis maintenant en Europe et dans le monde de nombreux donneurs d'ordre imposent à leurs fournisseurs d'atteindre un niveau de maturité élevé, font apparaître ces exigences dans les appels d'offres.

2.1. Destinataires

CMM apport des pratiques efficaces de gestion et d'ingénierie logiciel destinée au développement d'un logiciel de qualité, sécurisé, livré en temps voulu et conformément au budget fixé.

CMM à été conçu pour s'adresser surtout à des lecteurs ayant des connaissances dans le domaine du développement / maintenance logiciel, on les a cerner sous 3 catégories :

- Les managers : des organisations chercheront à comprendre et à améliorer leur capacité à développer des logiciels de façon efficace ; ils voudront comprendre les pratiques clés, qui font partie intégrante de tout processus efficace de développement ou de maintenance des logiciels.

  Ou bien, des managers des organisations clientes ou des maîtres d'oeuvre voudront évaluer les risques éventuels lorsqu'ils confient à une entreprise donnée les travaux couverts par un contrat.
- Les utilisateurs : voudront identifier les pratiques clés nécessaires pour atteindre le niveau de maturité suivant dans le CMM.
- Des auditeurs : le SEI pourra l'utiliser comme outil de référence lors de la rédaction des questions pour le questionnaire sur la maturité, lors d'une évaluation du processus logiciel (SPA) ou lors d'une évaluation de la capacité logiciel (SCE).
- Des formateurs : il pourra être utile aux instructeurs chargés de former les équipes d'évaluation du processus logiciel (SPA) et d'évaluation de la capacité logiciel (SCE).

3. Cadre d'évolution des processus

Les nouvelles technologies et méthodologies n'apportent pas leur promesse quant aux gains de productivité et de qualité car les entreprises et gouvernements ont une incapacité à gérer leurs processus logiciel.

Cette mauvaise gestion est due à un manque d'infrastructure et un manque de soutien de l'organisation.

Un bon produit est généralement obtenu par le déploiement d'efforts héroïque d'une équipe et n'est reproductible qu'avec les mêmes personnes.

La pérennité d'un projet n'est envisageable qu'en mettant en place une infrastructure de processus fondée sur des pratiques efficaces d'ingénierie logiciel et de gestion.

3.1. Comparaison entre organisation immature et organisation mature

- Organisation logiciel immature :

  - Les processus sont généralement improvisés par le personnel et la direction tout au long du projet.
  - Même si un processus logiciel a été défini, il n'est pas appliqué de façon rigoureuse ni mis en vigueur par la suite.
  - Les gestionnaires se contentent de résoudre les crises du moment.
  - Les délais et budgets sont généralement dépassés ce qui compromet la fonctionnalité et la qualité du produit.
  - Impossibilité de juger et donc de prédire le niveau de qualité des produits.

- Organisation logiciel mature :

  - Il y a une gestion générale du processus logiciel et de la maintenance logiciel.
  - Les processus logiciels sont communiqués au personnel en place et aux nouveaux arrivants.
  - Les travaux sont planifiés.
  - Les processus mis en place sont opérationnels et conformes au déroulement réel des travaux
  - Les processus définis sont mis à jour et les améliorations sont développées à l'aide d'essais contrôlés et/ou d'analyse coûts-bénéfices.
  - Les rôles et responsabilités sont répartis de façon non équivoque au niveau du projet et de toute l'organisation.
  - Les gestionnaires effectuent le suivi qualité des produits et de la satisfaction du client.
  - Un fondement objectif quantitatif permet d'évaluer la qualité des produits et d'analyser les difficultés.
  - Les résultats coût, calendrier, fonctionnalité et qualité sont obtenus.

➢ Comment passer d'un processus logiciel improvisé et chaotique à un processus mature bien structuré ? En élaborant un cadre d'évolution du processus logiciel. Ce cadre est le fruit de l'intégration : des concepts de processus logiciel,des concepts de capacités logiciel,des concepts de performance du processus logiciel (mesure des résultats réels) et des concepts de maturité du processus logiciel (définition, gestion, mesure, contrôle, efficacité).

3.2. Concepts d'évolution des processus

- **Processus** : Suite d'étapes réalisées dans un but donné.
- **Processus logiciel** : C'est un ensemble d'activités, de méthodes, de pratiques et de transformations permettant le développement et la maintenance de logiciels et des produits associés. (Plan de projet, document de conception, programmes, jeux de tests et manuels utilisateurs) Plus l'organisation évolue, plus le processus logiciel se précise et plus il est systématiquement mis en œuvre.
- **Performance du processus logiciel** : Résultats réels obtenus par l'application d'un processus logiciel. (Peut être restreinte par l'environnement. Exemple : Formation)
- **Capacité d'un processus logiciel** : Résultats attendus.
- **Maturité** : Désigne dans quelle mesure un processus est explicitement défini, géré, mesuré, contrôlé et efficace. Indicateur de la richesse du processus logiciel de l'organisation [JEHANNO 02].

4. Principe et structure du modèle CMM

L'amélioration continue d'un processus repose sur un grand nombre de petites étapes progressives. Pour pouvoir s'améliorer, l'organisation doit tout d'abord prendre connaissance de son niveau de maturité actuel. Pour ce faire, elle rempli des questionnaires produit par le SEI. Ces questionnaires servent de base pour mettre en évidence ses lacunes et se positionner dans la grille de maturité définie par le SEI.

Puis l'organisation se fixe un niveau de maturité supérieur à atteindre. Grâce au CMM, elle dispose alors d'une liste détaillée des actions à entreprendre. En réalisant l'ensemble des étapes intermédiaires du plan d'action, l'organisation atteint le niveau de maturité recherché.

Le CMM défini cinq niveaux de maturité, qui hiérarchisent la compétence d'une organisation dans le développement logiciel.

A l'exception du niveau 1, chaque niveau de maturité comporte plusieurs secteurs clés. Ces secteurs clés caractérisent les domaines à améliorer, pour répondre aux exigences du chaque niveau de maturité.

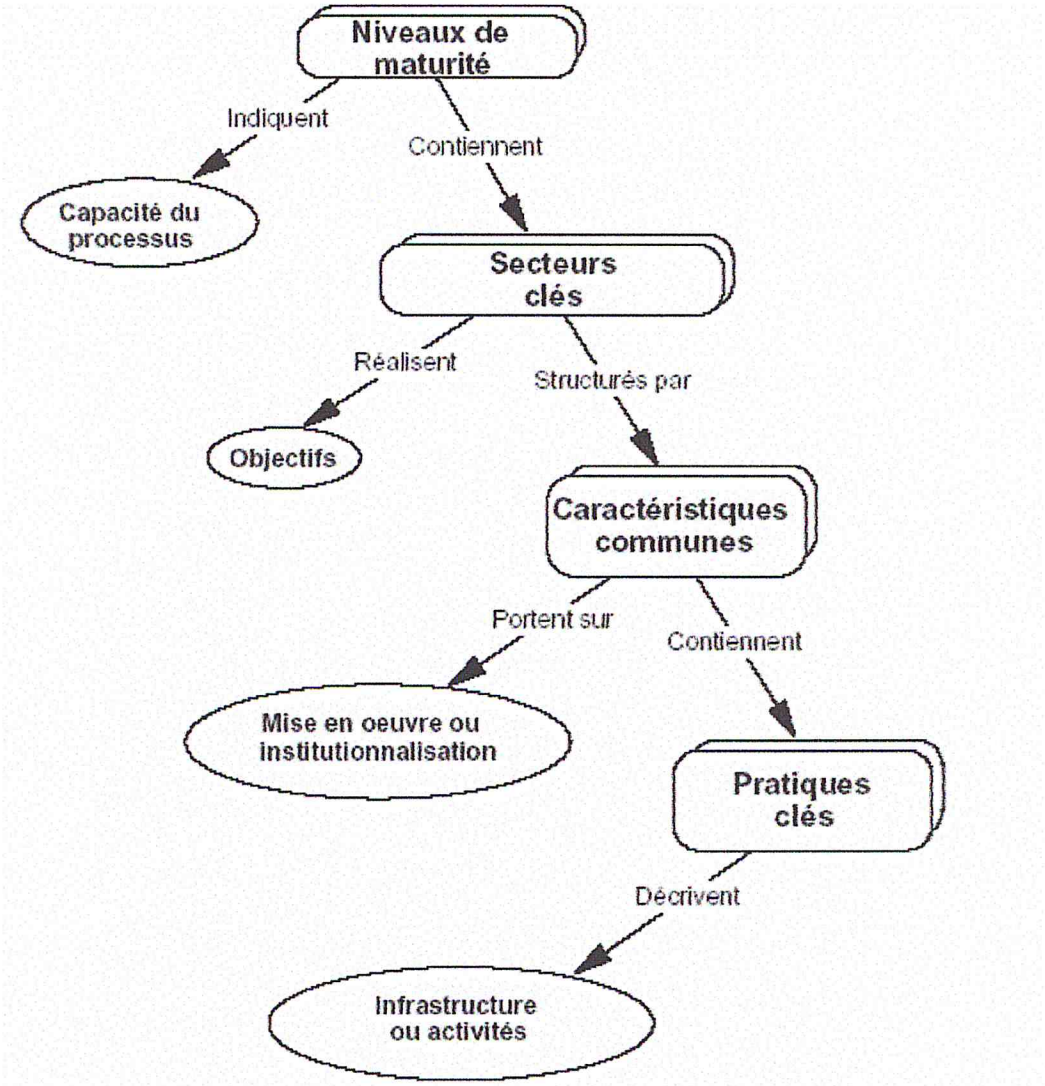


Figure 3-1 : Architecture du CMM

Chaque secteur clé comprend cinq caractéristiques communes. Celles-ci indiquent si la mise en œuvre d'un secteur clé est efficace, reproductible et durable. Enfin chaque secteur clé est décrit par des pratiques clés à respecter. Si ces pratiques clés sont mises en œuvre, on atteint l'amélioration requise pour le secteur clé. Il est à noter que les pratiques clés sont regroupées en terme de caractéristiques communes dans la documentation du CMM.

4.1. Niveaux de maturité

Un niveau de maturité est un palier d'évolution bien défini dans le cheminement vers un processus logiciel mature. Chaque niveau de maturité constitue une phase dans la mise en place des fondements nécessaires à l'amélioration continue d'un processus. Chaque niveau comporte un ensemble d'objectifs qui, une fois atteints, stabilisent un composant important du processus logiciel. L'atteinte de chaque niveau du modèle correspond à l'institutionnalisation d'un composant différent du processus logiciel, avec pour résultat l'enrichissement de la capacité du processus de l'organisation .

CMM comporte 5 niveaux de maturité (figure 3-2), Les 5 niveaux de maturité définissent une échelle pour l'évaluation de la maturité du processus logiciel et de la capacité du processus logiciel. Ils aident l'organisation à établir les priorités de ses efforts d'amélioration.

1) Le niveau initial : Le processus logiciel est caractérisé par la prédominance d'interventions ponctuelles, voire chaotiques. Très peu de processus sont définis et la réussite dépend de l'effort individuel.

2) Le niveau reproductible : Une gestion de projet élémentaire est définie pour assurer le suivi des coûts, des délais et de la fonctionnalité du produit. La discipline nécessaire au processus est en place pour reproduire la réussite de projets d'un même domaine d'application.

3) Le niveau défini : Le processus logiciel des activités de gestion et d'ingénierie est documenté, normalisé et intégré dans le processus logiciel standard de l'organisation. Tout nouveau projet de développement ou de maintenance logiciel fait intervenir une version adaptée et approuvée du processus logiciel standard de l'organisation.

4) Le niveau maîtrisé : Des mesures détaillées sont prises en ce qui concerne le déroulement du processus logiciel et de la qualité des produits de travail logiciel. Le processus logiciel et le niveau de qualité des produits sont connus et contrôlés quantitativement.

5) Niveau d'optimisation : Une amélioration continue du processus logiciel est mise en œuvre par une rétroaction quantitative émanant du processus lui-même et par l'application d'idées et de technologies innovatrices.

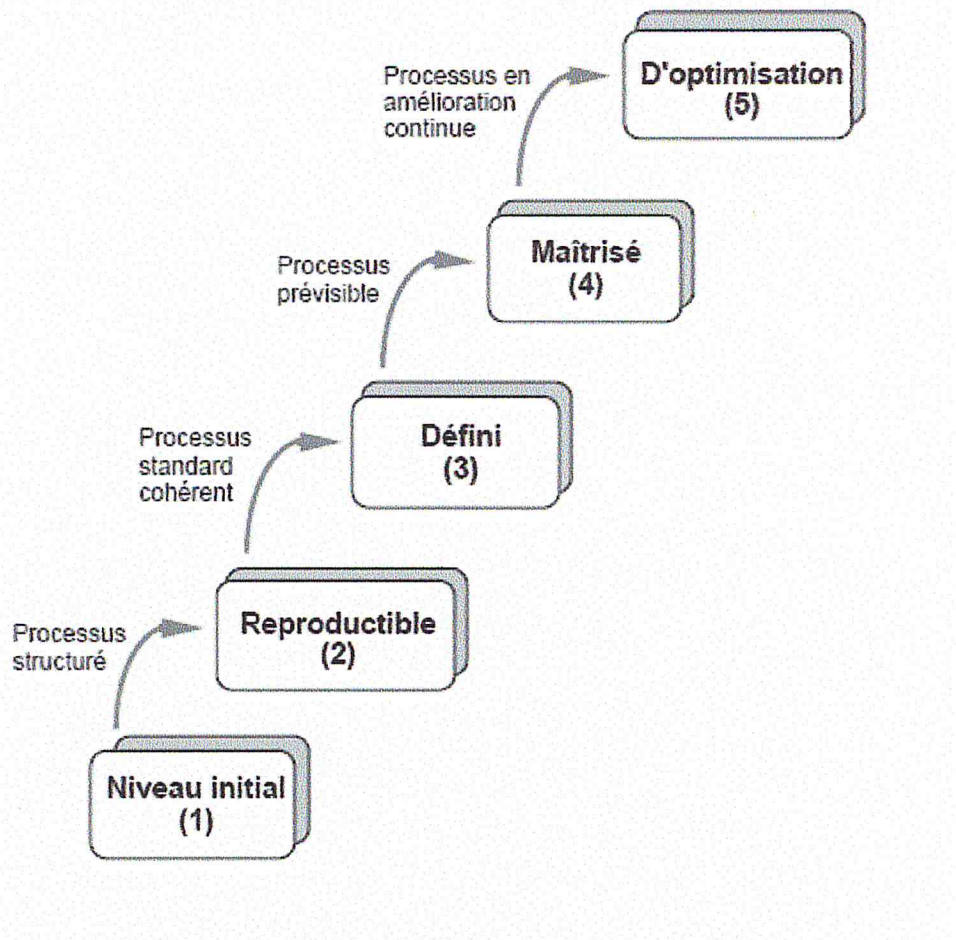


Figure 3-2 : Les cinq niveaux de maturité du processus logiciel

4.2. Définition comportementale des niveaux de maturité

4.2.1. Niveau 1 – Le niveau initial

Au niveau initial, l'organisation ne fournit aucun environnement stable de développement et de maintenance logiciel. Lors d'une crise, les projets abandonnent les procédures planifiées et les efforts régressent alors vers un mode codage et tests. Tout repose sur un chef de projet aux qualités exceptionnelles qui arrive à résister aux pressions. En cas de départ, son action stabilisatrice disparaît avec lui.

Les processus logiciels sont constamment remplacés ou modifiés au fur et à mesure de l'avancement des travaux (processus improvisés). Les délais, les budgets, la fonctionnalité et la qualité des produits sont généralement imprévisibles. La performance ne peut être évaluée que sur une base individuelle et non en fonction des capacités organisationnelles.

4.2.2. Niveau 2 – Le niveau reproductible

Au niveau reproductible, les directives de gestion de projets logiciel et les procédures permettant l'application de ces directives sont établies. La planification et la gestion des nouveaux projets dont fondées sur l'expérience antérieure acquise à l'occasion de projets semblables.

Un des objectifs menant au niveau 2 est l'institutionnalisation de processus efficaces de gestion des projets logiciels qui permettent à l'organisation de reproduire des pratiques éprouvées lors de projets antérieurs, même si les processus utilisés pour le projet en cours sont différents.

Un processus opérationnel se distingue par l'existence de pratiques et de documentation, par sa mise en œuvre obligatoire, par l'existence d'une formation appropriée et de mécanismes de mesure ainsi que par son potentiel d'amélioration.

Les projets des organisations de niveau 2 font intervenir des mécanismes élémentaires de contrôle de gestion logiciel. Les engagements sont réalistes et sont pris en fonction des résultats antérieurs observés et des exigences du projet en cours.

Les responsables logiciel effectuent le suivi des coûts, des délais et de la fonctionnalité. Les difficultés en matière de respect des engagements sont identifiées au fur et à mesure qu'elles surgissent. On établit un référentiel des exigences logiciel et des produits devant les satisfaire, et leur intégrité est contrôlée. Le projet logiciel fait intervenir des normes définies et l'organisation veille à ce que celles-ci soient scrupuleusement respectées.

En résumé :

La capacité du processus logiciel des organisations de niveau 2 peut être définie comme étant structurée. La planification et le suivi des projets sont stables et les réussites antérieures peuvent être reproduites. Les contrôles sont efficaces, le système de gestion est fondé sur des plans réalistes dressés en fonction de la performance réalisée au cours des projets antérieurs.

4.2.3. Niveau 3 – Le niveau défini

Au niveau défini, Le processus standard de développement et de maintenance à travers toute l'organisation est documenté, intégrant en un tout cohérent les processus logiciel et de gestion.

Ce processus est mentionné dans le CMM comme étant "Processus logiciel standard de l'organisation".

Le niveau 3 permet une intervention plus efficace du responsable logiciel et du personnel technique qui dispose de pratiques efficaces d'ingénierie logiciel. Un groupe est chargé des activités reliées au processus logiciel de l'organisation. (SEPG : Groupe d'ingénierie du processus logiciel).

Des programmes de formation sont mis en place dans toute l'organisation afin que chacun acquière les compétences pour assumer le rôle qui lui a été confié.

Au cours d'un projet, le processus logiciel standard de l'organisation est adapté de façon à développer un processus propre à ce projet en prenant en compte les caractéristiques spécifiques de ce dernier.

Dans le CMM, ce processus adapté est appelé : "Processus logiciel défini du projet".Ce processus contient un ensemble intégré et cohérent de processus d'ingénierie logiciel et de gestion bien définis. Un processus bien défini se reconnaît à la présence de critères de déclenchement, de normes et procédures quant à l'exécution des travaux, de mécanismes de vérification, de sortie et de critères d'achèvement. La direction a un bon aperçu des progrès techniques pour tous les projets.

En résumé :

La capacité du processus des organisations de niveau 3 est définie comme étant normalisée et cohérente car les activités d'ingénierie logiciel et de gestion sont stables et reproductibles. Les coûts, les délais et la fonctionnalité sont contrôlés pour chaque gamme de produit et la qualité fait l'objet d'un suivi. C'est le fruit d'une compréhension commune et généralisée au sein de l'organisation quant aux activités, aux rôles et aux responsabilités intrinsèques à tout le "processus logiciel défini du projet".

4.2.4. Niveau 4 - Le niveau maîtrisé

Au niveau maîtrisé, l'organisation se fixe des objectifs quantitatifs de qualité à la fois pour les produits et les processus logiciels. La productivité et la qualité sont mesurées pour les tâches importantes de tous les projets dans le cadre d'un programme organisationnel de mesures.Une base de données du processus logiciel de l'organisation est utilisée pour recueillir et analyser les données disponibles des processus logiciel définis du projet mis en œuvre dans le cadre de chaque projet.

Au niveau 4, les processus logiciels sont dotés de moyens qui permettent des mesures cohérentes et bien définies. Ces mesures constituent une base d'évaluation quantitative des processus et des produits logiciels.

Au cours d'un projet, les produits et processus sont contrôlés par diminution des variations de performance de processus de façon à ce que ces variations ne dépassent pas des limites quantitatives acceptables.

En résumé :

La capacité du processus logiciel des organisations de niveau 4 peut être défini comme étant prévisible parce que le processus est mesuré et opère dans des limites mesurables. L'organisation peut prévoir les tendances en termes de qualité des processus et des produits à partir des limites quantitatives établies. Lorsque ces limites sont dépassées, une action est entreprise afin de corriger la situation. Les produits logiciels sont de la haute qualité attendue.

4.2.5. Niveau 5 – Le niveau d'optimisation

Au niveau d'optimisation, l'amélioration continue des processus est la principale préoccupation de toute l'organisation. L'organisation a les moyens d'identifier les faiblesses de ses processus et peut renforcer ceux-ci de façon proactive afin de prévenir tout défaut.

Les données d'efficacité du processus logiciel sont utilisées dans le cadre d'analyses coûts-bénéfices des nouvelles technologies et des propositions de changement du processus logiciel. Les innovations fondées sur les meilleures pratiques d'ingénierie logiciel sont identifiées et mises en œuvre dans toute l'organisation.

Les équipes de projet logiciel des organisations de niveau 5 analysent les défauts en vue de déterminer leurs causes. Les processus logiciel sont évalués de façon à empêcher la réapparition des défauts connus et les leçons retenues sont communiquées aux équipes des autres projets.

En résumé :

La capacité du processus logiciel des organisations de niveau 5 peut être définie comme étant continuellement en cours d'amélioration puisque ces organisations ne cessent d'élargir l'éventail des capacités de leur processus, améliorant ainsi la performance du processus mis en œuvre pour chaque projet. Cette amélioration est à la fois le fruit de l'évolution progressive du processus actuel et des innovations mettant en œuvre des méthodes et des technologies nouvelles.

4.3. Tentation de sauter d'un niveau dans l'évolution

Les niveaux de maturité du CMM décrivent les caractéristiques d'une organisation à un niveau donné. Chaque niveau établit les bases sur lesquelles s'appuient les niveaux suivants pour mettre en œuvre des processus d'une manière complète et efficace.

Une organisation peut toutefois utiliser à profit des processus décrits à un niveau plus élevé que le sien. Bien que les processus d'ingénierie notamment l'analyse des exigences, la conception, la programmation et les tests ne soient abordés dans le CMM qu'à partir du niveau 3, une organisation de niveau 1 doit également réaliser ces activités. Il est possible qu'une organisation de niveau 1 ou 2 soit en mesure d'effectuer efficacement des revues par les pairs (Niveau 3), des analyses de Pareto (Niveau 4) ou des essais pilotes de nouvelles technologies (Niveau 5).

Sauter un ou plusieurs niveaux de maturité est contre productif puisque chaque niveau permet de jeter les bases nécessaires au niveau suivant. Le CMM identifie des niveaux de maturité que l'organisation doit atteindre successivement pour implanter une culture d'excellence en matière d'ingénierie logicielle.

- Une organisation de niveau 1 tentant d'appliquer un processus défini (Niveau 3) avant d'avoir établi la reproductibilité des processus (Niveau 2) échoue habituellement parce que les chefs de projets ne peuvent faire face aux pressions des coûts et des délais.
- Une organisation désirant mettre en œuvre un processus maîtrisé (Niveau 4) sans avoir auparavant mis au point un processus défini (Niveau 3) verra ses efforts voués à l'échec puisqu'il n'existe aucune base commune permettant d'interpréter les mesures.

Les efforts d'amélioration d'un processus doivent être axés sur les besoins de l'organisation dans son contexte d'entreprise. La possibilité de mettre en œuvre des processus d'un niveau de maturité plus élevé n'implique nullement qu'il soit possible de sauter un ou plusieurs niveaux d'évolution.

5. Définition opérationnelle du modèle CMM

Le CMM est un cadre illustrant le cheminement d'amélioration recommandé pour les organisations logiciel désirant augmenter la capacité de leur processus. L'élaboration opérationnelle du CMM permet d'utiliser ce dernier de diverses façons, dont les quatre suivantes :

♦ Par les équipes d'évaluation du processus logiciel (**SPA**) pour identifier les forces et faiblesses de l'organisation. c'est une équipe de spécialistes en logiciels en vue de faire le point sur le processus logiciel actuel de l'organisation et de déterminer les questions prioritaires, en termes de processus, auxquelles l'organisation doit faire face et d'obtenir le soutien organisationnel voulu en vue de l'amélioration du processus logiciel.

♦ Par les équipes d'évaluation de la capacité logiciel (**SCE**) pour identifier les risques que présente chaque maître d'œuvre lors de l'octroi des contrats et du suivi de leur exécution.

Maître d'œuvre : Particulier, société, compagnie ou association gérant un contrat de sous-traitance en vue de la conception, du développement et/ou de la fabrication d'un ou plusieurs produits.

♦ Par les gestionnaires et les techniciens en vue de comprendre les activités nécessaires à la planification et à la mise en œuvre d'un programme d'amélioration du processus logiciel de l'organisation.

♦ Par les groupes chargés de l'amélioration des processus (Dont le **SEPG**, groupe d'ingénierie du processus logiciel) comme référence dans la définition et l'amélioration du processus logiciel dans leur organisation.

Compte tenu de sa polyvalence, le CMM doit être décomposé de façon suffisamment détaillée pour que les recommandations pratiques puissent être dégagées de la structure des niveaux de maturité. Cette décomposition identifie également les processus et leurs structures qui caractérisent la maturité d'un processus logiciel et la capacité d'un processus logiciel.

6. Structure interne des niveaux de maturité

6.1. Les secteurs clés

Le SEI définit le secteur clé étant un ensemble d'activités associées dont l'accomplissement collectif mène à un ensemble d'objectifs considérés comme importants pour la mise en place d la capacité du processus à un niveau de maturité donné. Par le terme ' clé' le SEI juge que certains secteurs sont déterminants pour la capacité des processus.

A part le niveau 1, chaque niveau de maturité comporte plusieurs secteurs clés (entre 2 et 7) représentant une voie par la quel l'organisation identifie les questions et décrit les aspects pour améliorer un processus logiciel afin d'atteindre un niveau de maturité donné.

Nous allons citer les secteurs clés de chaque niveau de maturité :

- **Secteurs-clés au niveau 2**

Objectif: établir les contrôles de base de gestion de projets

- La gestion des exigences vise à établir une compréhension commune, entre le client et le projet logiciel, des exigences du client que le projet logiciel se propose de satisfaire. Cet accord avec le client est le fondement de la planification (secteur clé planification de projet logiciel) et de la gestion (secteur clé suivi et supervision de projet logiciel) du projet logiciel. Le contrôle des relations avec le client repose sur un processus efficace de contrôle des changements (secteur clé Gestion de configuration logiciel).
- La planification de projet logiciel vise à établir des prévisions raisonnables pour la mise en oeuvre des travaux d'ingénierie logiciel et la gestion du projet logiciel. Ces prévisions sont indispensables à une bonne gestion du projet (secteur clé suivi et supervision de projet logiciel). L'absence de prévisions réalistes empêche toute gestion efficace du projet.
- Le suivi et la supervision de projet logiciel visent à donner une bonne perspective de l'avancement réel des travaux de façon que les gestionnaires puissent intervenir efficacement lorsque la performance du projet logiciel s'écarte de façon significative des prévisions logicielles.
- La gestion de la sous-traitance logicielle vise à sélectionner des sous-traitants qualifiés et à les gérer efficacement. La gestion de base mise en oeuvre dans le cadre de ce secteur clé repose sur des considérations de Gestion des exigences, de Planification de projet logiciel, de Suivi et supervision de projet logiciel de même que sur la coordination nécessaire des activités d'assurance qualité logiciel et de Gestion de configuration logiciel. Ces mécanismes de contrôle sont appliqués de façon appropriée aux activités des sous-traitants.

- L'assurance qualité logicielle vise à fournir aux gestionnaires la vision appropriée sur le processus utilisé par le projet logiciel et sur les produits en élaboration. L'assurance qualité logicielle fait partie intégrante de la plupart des processus d'ingénierie logiciel et de gestion.
- La gestion de configuration logicielle vise à établir et à maintenir l'intégrité des produits du projet logiciel tout au long du cycle de vie logiciel du projet. La gestion de configuration logicielle fait partie intégrante de la plupart des processus d'ingénierie logiciel et de gestion.

• **Secteurs-clés au niveau 3**

Objectif: établir l'infrastructure pour l'organisation qui permet d'assurer la gestion efficace des processus de gestion et d'ingénierie de tous les projets.

- La focalisation organisationnelle sur les processus vise à établir une responsabilité organisationnelle quant aux activités reliées au processus logiciel en vue d'améliorer l'ensemble de la capacité du processus logiciel de l'organisation. Les activités de Focalisation organisationnelle sur les processus ont pour premier résultat de produire un ensemble d'acquis processus logiciel, acquis qui sont décrits dans la définition du processus de l'organisation. Ces acquis sont utilisés dans le cadre des projets logiciel (secteur clé gestion logiciel intégrée).
- La définition du processus de l'organisation vise à développer et à maintenir un ensemble utilisable d'acquis processus logiciel améliorant la performance du processus d'un projet à l'autre et constituant le fondement pour des bénéfices cumulatifs et sur le long terme pour l'organisation. Ces acquis constituent un fondement stable permettant l'institutionnalisation par le biais d'interventions telles que la formation (secteur clé programme de formation).
- Le programme de formation vise à développer les compétences et les connaissances des personnes pour qu'elles puissent jouer efficacement le rôle qui leur a été attribué. Bien que la formation soit une responsabilité organisationnelle, chaque projet logiciel devrait donner lieu à l'identification des compétences essentielles à sa réalisation et inclure la formation requise pour répondre aux exigences particulières correspondantes, s'il y a lieu.

- La gestion logiciel intégrée vise à intégrer les activités d'ingénierie et de gestion logiciel sous forme d'un processus logiciel défini et cohérent adapté à partir du processus logiciel standard de l'organisation et des acquis processus associés (secteur clé définition du processus de l'organisation). Cette adaptation est fonction du contexte d'entreprise et des exigences techniques du projet, tels que décrits dans le secteur clé Ingénierie de produits logiciel. La gestion logiciel intégrée est le fruit des activités de Planification de projet logiciel et de Suivi et supervision de projet logiciel du niveau 2.
- L'ingénierie de produits logiciels vise à exécuter systématiquement un processus d'ingénierie bien défini intégrant toutes les activités d'ingénierie logiciel en vue de produire efficacement des produits logiciels cohérents et corrects. Ce secteur clé décrit les activités techniques du projet, dont l'analyse des exigences, la conception, la programmation et les tests.
- La coordination intergroupe vise à établir un moyen permettant au groupe d'ingénierie logiciel de collaborer activement avec les autres groupes d'ingénierie de façon que le projet puisse satisfaire plus efficacement les besoins du client. La coordination intergroupe est l'aspect interdisciplinaire de la gestion logiciel intégrée qui dépasse le cadre de l'ingénierie logiciel. En effet, non seulement un processus logiciel doit-il être intégré, mais les interactions du groupe d'ingénierie logiciel avec les autres groupes doivent être coordonnées et contrôlées.
- Les revues par les pairs visent à éliminer tôt et efficacement les défauts des produits. Un important corollaire de cette activité est le développement d'une meilleure compréhension des produits de travail logiciel et des défauts pouvant être empêchés. La revue par les pairs est une méthode importante et efficace utilisée en ingénierie de produits logiciel et pouvant être appliquée par le biais d'inspections telles que préconisées par des «lectures croisées» structurées, ou par d'autres méthodes d'évaluation collégiale.

  • **Secteurs-clés au niveau 4**

  Objectif: établir la compréhension quantitative du processus et des produits logiciels développés.

- La gestion quantitative de processus vise à contrôler quantitativement la performance du processus appliqué dans le cadre du projet logiciel. La performance du processus logiciel correspond aux résultats réels obtenus en respectant un processus logiciel donné. L'accent est mis sur l'identification des causes spéciales de variation observées dans un projet mesurable et stable, et sur la correction, au besoin, des circonstances à l'origine de ces variations transitoires. Le secteur clé Gestion quantitative de processus permet d'ajouter un programme complet de mesures aux pratiques des secteurs clés définition du processus de l'organisation, gestion logiciel intégrée, Coordination intergroupes et Revues par les pairs.
- La gestion de la qualité logicielle vise à développer une compréhension quantitative de la qualité des produits logiciels issus du projet et à favoriser la réalisation d'objectifs de qualité spécifiques. La gestion de la qualité logiciel consiste à appliquer un programme complet de mesures aux produits de travail logiciel décrits dans le secteur clé ingénierie de produits.

• **Secteurs-clés au niveau 5**

Objectif: amélioration continue et mesurable du processus

- La prévention des défauts vise à identifier les causes de défauts et à empêcher qu'ils ne se reproduisent. Le groupe de projet logiciel analyse les défauts, identifie leurs causes et modifie en conséquence le processus logiciel défini, tel que décrit dans le secteur clé Gestion logiciel intégrée. Les changements de processus de portée générale sont également effectués dans les autres projets logiciels, selon la procédure décrite dans le secteur clé gestion des changements du processus.
- La gestion des changements technologiques vise à identifier les nouvelles technologies avantageuses (outils, méthodes et processus) et à en effectuer le suivi de façon ordonnée au sein de l'organisation, conformément à la marche à suivre décrite dans le secteur clé gestion des changements du processus. L'accent est mis sur la mise en œuvre efficace des innovations dans un environnement en constante évolution.
- La gestion des changements du processus vise l'amélioration continue du processus logiciel utilisé par l'organisation en vue d'améliorer la qualité logiciel, d'augmenter la productivité et de diminuer la durée du cycle de développement des produits. La gestion des changements du processus étend à l'organisation tout entière les améliorations progressives découlant de la prévention des défauts et les améliorations innovatrices produites par la gestion des changements technologiques.

Il existe toutefois des liens entre les secteurs clés puisqu'ils peuvent être classés en trois catégories générales (figure 3-2).

Catégorie de processus **Niveaux**	**Processus de gestion** Planification, gestion, etc. du projet logiciel	**Processus organisationnels** Revues par la Direction, etc.	**Processus d'ingénierie** Analyse des exigences, conception, programmation, tests, etc.
5 D'optimisation		▪Gestion des changements technologiques ▪ Gestion des changements du processus	▪ Prévention des défauts
4 Maîtrisé	▪Gestion quantitative de processus		▪Gestion de la qualité logiciel
3 Défini	▪Gestion logiciel intégrée ▪Coordination intergroupes	▪ Focalisation organisationnelle sur les processus ▪ Définition du processus de l'organisation ▪ Programme de formation	▪ Ingénierie de produits logiciel ▪ Revues par les pairs
2 Reproductible	▪Gestion des exigences ▪Planification de projet logiciel ▪Suivi et supervision de projet logiciel ▪Gestion de la sous-traitance logiciel ▪Assurance-qualité logiciel ▪Gestion de configuration logiciel		
1 Niveau initial	Processus improvisé		

Figure3-3 : Secteurs clés classés par catégories [WEBER 93]

6.2. Les caractéristiques communes

Les pratiques clés de chaque secteur clé sont regroupées et organisées parmi 5 caractéristiques communes.

Les caractéristiques communes sont des attributs indiquant si la mise en œuvre et l'institutionnalisation d'un secteur clé sont ou non efficaces, reproductibles et durables.

Les 5 caractéristiques communes du modèle CMM sont :

- Engagement de réalisation

  Les activités que l'organisation doit entreprendre pour établir un processus durable. Etablir une politique au niveau de l'organisation et obtenir le parrainage de la haute direction.

- Capacité de réalisation

  Les conditions préalables d'un projet ou d'une organisation pour implémenter le processus de manière compétente. Implique ressources, structures organisationnelles et formation.

- Activités réalisés

  Les rôles et procédures nécessaires pour implémenter un secteur-clé. Etablir plans et procédures, faire le travail, le traquer, entreprendre des actions correctrices.

- Mesures et analyse

  Mesurer le processus et analyser les mesures obtenues, contrôle statistique.

- Vérification de mise en œuvre

  Description des étapes pour assurer que les activités sont exécutées conformément au processus standard, inclut revues et audits de gestion et assurance de qualité.

6.3. Les pratiques clés

Chaque pratique clés est décrite en termes de pratiques clés contribuant à en satisfaire des objectifs. Une pratique clé est énoncée sous forme d'une seule phrase souvent suivie d'une description plus détaillée appelée pratique secondaire, comprenant aussi des exemples et des renseignements complémentaires.

Les pratiques clés décrivent «ce qui est à faire» mais n'indiquent pas de manière absolue «comment» ces objectifs devraient être atteints. Des pratiques équivalentes peuvent aussi permettre d'atteindre les objectifs d'un secteur clé.

Les pratiques clés devraient être interprétées rationnellement afin d'apprécier si les objectifs d'un domaine clé ont été effectivement atteints.

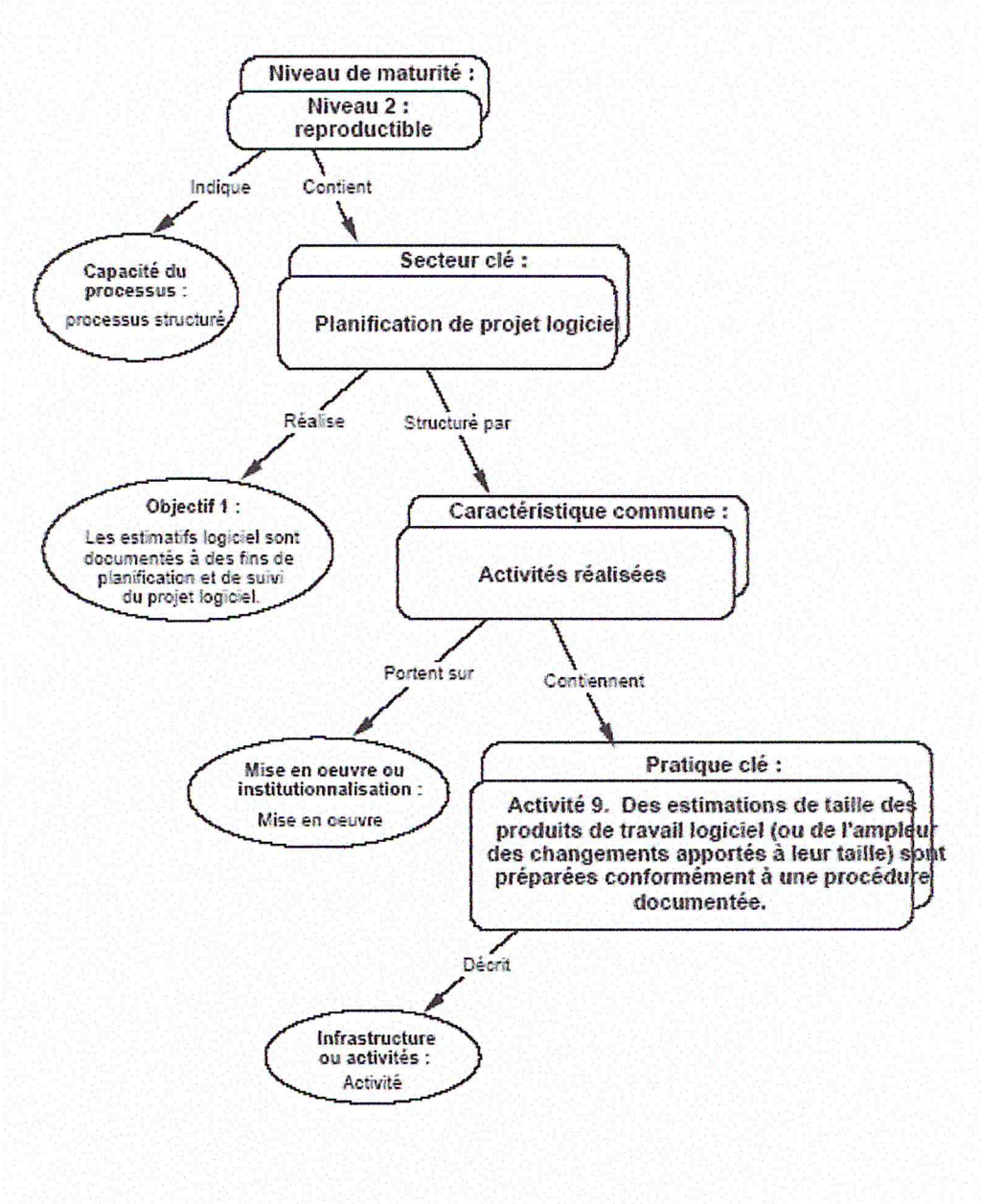


Figure 3-4 : Exemple d'énoncé d'une pratique clé [BOURNICHE 01]

7. Conclusion

Le modèle CMM est un cadre structurel qui permet un bon cheminement des processus de développement logiciel.il ne décrit pas la façon de faire mais fait état des pratiques généralement utilisé. Il prend en compte toutes les étapes du cycle de vie de logiciel (conception, test, etc.), tous les éléments de l'entreprise (gestionnaire, ingénierie,...), les interrelations interne (logiciels, matériels,..) et externe (fournisseurs, certification,...).

Le chapitre qui suit décrira le système de gestion de la qualité associé au modèle CMM.

CHAPITRE IV

Système de gestion de la qualité du logiciel et UML

1. Introduction

Dans le chapitre précédant, nous avons présenté la démarche qualité basée sur l'évaluation de processus SW - CMM. Dans ce qui suit, nous allons exposé la mise en œuvre de cette dernière, c'est-à-dire le système de gestion de la qualité.

Aussi, nous allons UML (Unified Methode Language), un formalisme de modélisation orienté objet. UML sera utilisé pour le développement du système de gestion de la qualité.

2. Système de gestion de la qualité

Notre système de gestion de la qualité est basé sur le cycle IDEAL (Initiating Diagnosting Establishing Acting Leveraging), celui-ci est très proche du cycle de Deming.

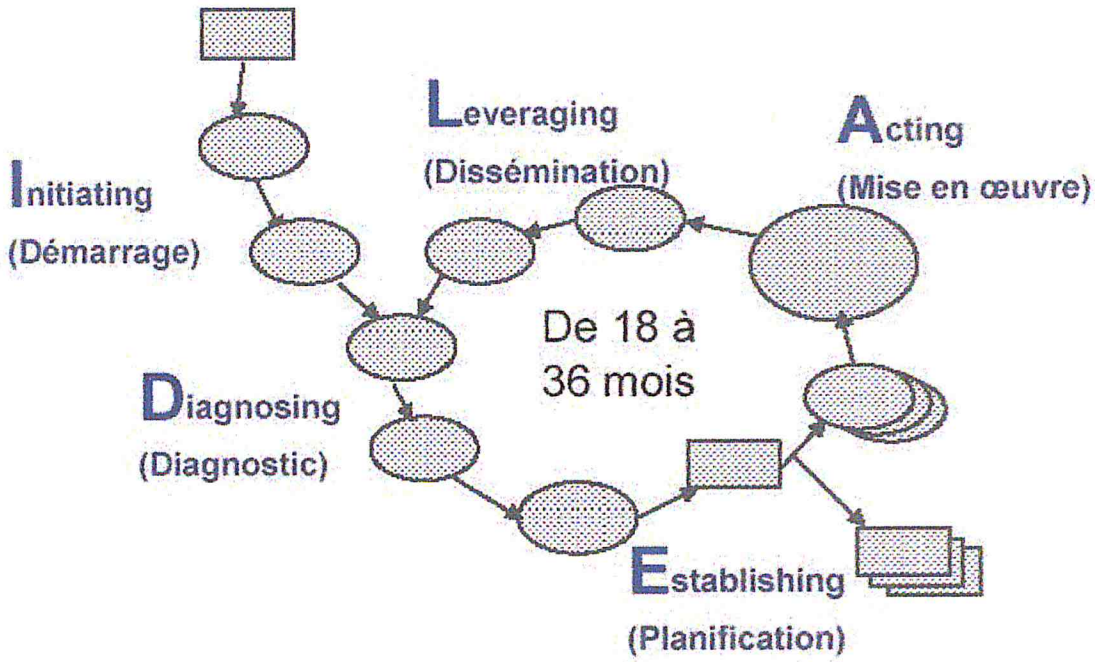


Figure 4-1 : Le cycle IDEAL [Basque 02].

Notre système consiste à améliorer les processus logiciel d'une manière cyclique :

- Déterminer l'état courant des processus ainsi que ses forces et faiblesses.
- Identifier et planifier les améliorations souhaitées.
- Mise en application les actions d'amélioration.
- Mesurer et contrôler les effets de l'amélioration avant de recommencer le cycle.

Ainsi le système de gestion de la qualité est structuré en modules, comme représenté par la figure4-2.

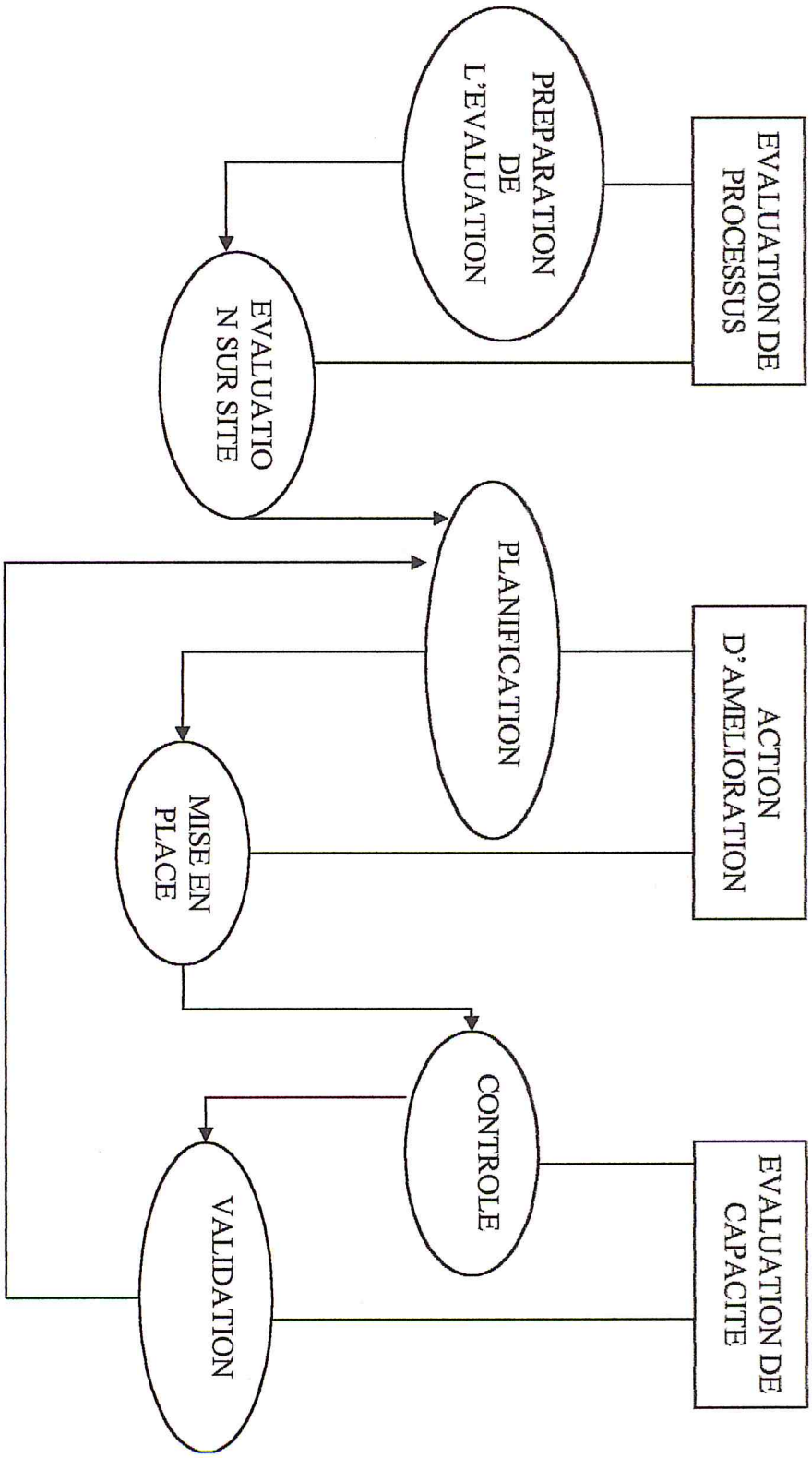


Figure 4-2 : Le système qualité

Avant toute action, l'organisme doit spécifier sa politique qualité, ses engagements et ses besoins.

2.1. Evaluation des processus

Dans ce module, on établit l'état des lieux, décrit la maturité des processus à travers les besoins propres de l'entreprise.

Un constat décrivant le niveau de maturité de l'entreprise, une liste des forces et faiblesses et des contraintes organisationnelles, ainsi des recommandations sur les éventuelles amélioration à apporter (classés par priorité).

Ce module contient 2 phases :

➢ **Préparation de l'évaluation** : cette phase comporte ;

- identification des objectifs ;
- sélection et formation de l'équipe d'évaluation (SPA) ;
- collection des documents (questionnaires, documents techniques.).

➢ **Evaluation sur site** : Dans cette phase, une interview sera faite avec les chefs de projet et les responsables des services représentatifs. Le rapport final contient en général des informations sur : le niveau de maturité du processus et de ces différentes parties, des commentaires sur certaines pratiques et des recommandations sur les éventuelles améliorations à apporter (classées parfois par priorité).

2.2. Action d'amélioration

Ce module consiste à mettre en place l'amélioration des processus tel décrit par CMM.

Le CMM s'avère particulièrement utile au niveau de la planification des actions, de l'exécution des plans d'action et de la définition des processus.

Ce module contient 2 phases :

➢ **Planification** : Dans cette phase, on doit :

- sélectionner le groupe d'ingénierie du processus logiciel (SEPG) ;
- comparer les pratiques actuelles aux objectifs des secteurs clés de CMM, tenant compte des objectifs de l'entreprise, des priorités de la direction et la capacité de l'organisation à mettre en œuvre une pratique ;
- Déterminer quelles pratiques doivent êtres appropriées à chaque processus, suivant un plan de priorité.

➢ **Mise en place** : Elle doit être réalisée de façon ordonnée et en respectant le plan, la perspective finale étant de maintenir et d'institutionnaliser les améliorations. Les différentes étapes peuvent être :

- la définition du processus logiciel ;
- la définition du groupe du processus logiciel ;
- la documentation approprié à chaque processus logiciel ;
- la définition des instruments de mesure pour quantifier son évolution ;
- les tests des nouveaux processus et instruments dans un projet pilote ;
- la vérification de l'amélioration par l'utilisation des mesures collectées ;
- l'institutionnalisation des nouveaux processus et résultats à travers l'organisation.

2.3. Evaluation des capacités

L'évaluation de la capacité du logiciel se déroule dans un contexte plus axé sur l'audit. L'accent est mis sur la création d'une piste de vérification documentée décrivant le processus logiciel réellement mis en oeuvre par l'organisation.

Ce module comporte 2 phases :

➢ **le contrôle des effets de l'amélioration** : On procède à une sélection d'un groupe d'évaluation de capacité logiciel (SCE), celui-ci fait une réévaluation limitée aux processus affectés par les actions engagées pour en constater les effets. Ensuite, ses résultats sont comparés avec ceux de la première évaluation. Cela permet de vérifier si les améliorations sont bien réelles.

➢ **la validation de l'amélioration** : Les profits et avantages tirés de ces efforts doivent être permanents et stables. Les efforts d'amélioration doivent être analysés et documentés pour qu'ils soient poursuivis.

3. Le langage UML

UML (Unified Methode Language) est un langage unifié de modélisation objet.

UML est issu de divers méthodes orienté objet : JACOBSON (OOSE), BOOCH (OOD), RUMBAUGH (OMT), ainsi il prend le meilleur de chacune ses méthodes.

Le modèle conceptuel d'UML [BERNADI 02] comprend les notions de base génériques du langage .Il définit trois sorte de bases :

- Des éléments, qui sont les abstractions élémentaires à un modèle.
- Des relations, qui constituent les liens entre les éléments.
- Des diagrammes, qui regroupent les éléments et les liens en un ensemble dignes d'intérêts.

3.1. Les éléments

Il existe quatre types d'éléments dans UML :

- Structurels : représentent les parties les plus statiques de modèle (classes, interfaces, collaborations…).
- Comportementaux : représentent les parties dynamiques (interactions, automates à états).
- Regroupements : représentent des boites dont le système peut être décomposé (paquetage).
  - Paquetage : cet élément regroupe des éléments structurels, comportementaux et autres [Gabay 98].

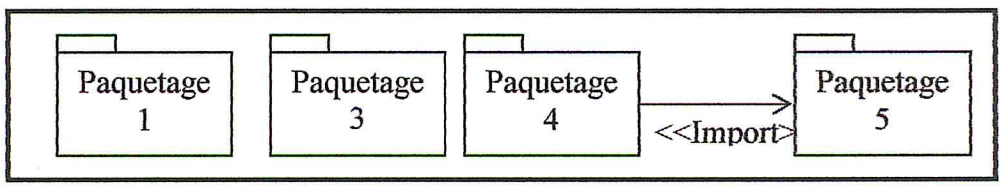


Figure 4-3 : Représentation du paquetage

- Annotations : représentent des parties explicatives d'UML (note).
  - Note : commentaire accompagnant tout élément à des fins de descriptions, d'explications et de remarques.

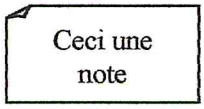


Figure 4-4 : Représentation d'une note

3.2. Les relations

Il existe quatre types de relations dans UML :

➢ Dépendance : relation sémantique entre deux éléments selon laquelle un changement apporté à l'un peut affecter la sémantique de l'autre.

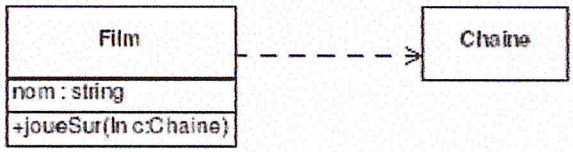


Figure 4-5 : Relation de dépendance

➢ Association : relation structurelle qui décrit un ensemble de liens entre les objets des éléments différents.

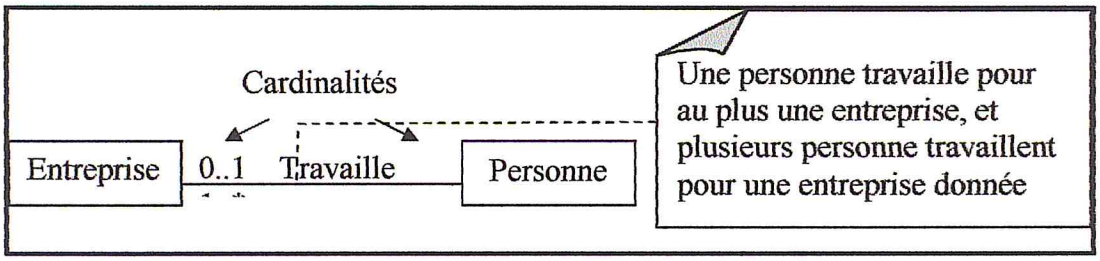


Figure 4-6 : Relation d'association

• Agrégation : est une association dont laquelle une relation ''partie-de'' est définit entre ses extrémités.

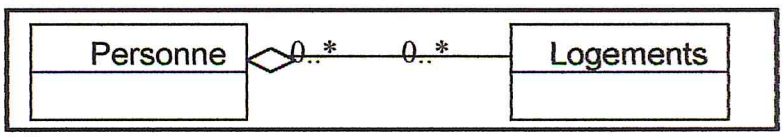


Figure 4-7 : Relation d'agrégation

• Composition : est une forme de forte agrégation mettant en avant une notion de priorité forte et de coïncidence du cycle de vie. Les composites sont créés et détruits au même temps que le composant.

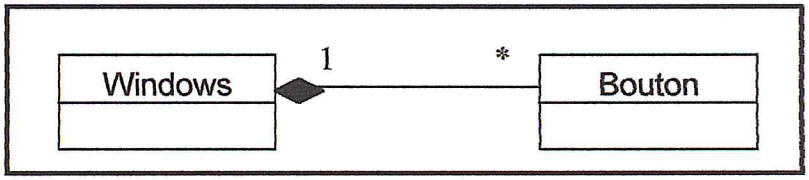


Figure 4-8 : Relation de composition

➢ Généralisation : relation entre deux éléments dont les attributs de l'élément spécialisé peuvent êtres substituer aux attributs de l'élément généralisé.

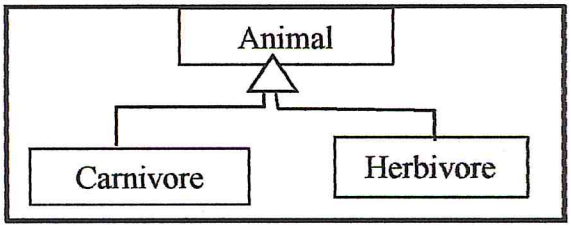


Figure 4-9 : Relation de généralisation

➢ Réalisation : relation entre une classe et une interface spécifiant que la classe implémente les opérations définies par l'interface.

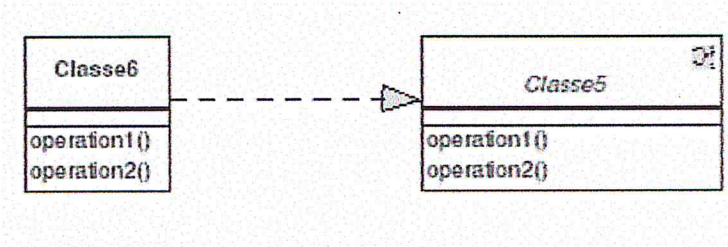


Figure 4-10 : Relation de réalisation

3.3. Les diagrammes

➢ Un diagramme est une représentation graphique d'un ensemble d'éléments qui constituent le système. C'est un graphe où les sommets correspondent aux éléments et les arcs aux relations.

UML définit neuf types de diagrammes repartis en deux catégories :

- Diagrammes structurels : diagrammes de classes, d'objets, de composants, de déploiements et cas d'utilisations.
- Diagrammes comportementaux : diagramme d'activités, de séquences, d'états de transitions et de collaboration.

3.3.1. Diagramme de cas d'utilisations

➢ Un diagramme de cas d'utilisation représente des acteurs, des cas d'utilisations englobés par la limite du système et des relations entre eux.

Un diagramme de cas d'utilisation décrive sous forme d'action et de réaction, le comportement d'un système d'un point de vue utilisateur.

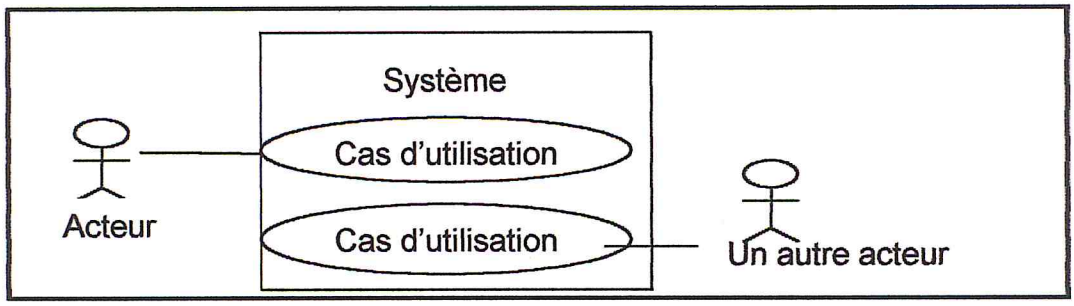


Figure 4-11 : Représentation d'un diagramme de cas d'utilisation

- Un acteur représente un rôle joué par une personne ou une chose qui interagit avec un système [Muller 97].
- Un cas d'utilisation spécifie une séquence d'actions q'une entité réalise, en interagissant avec les acteurs de l'entité.

Il existe trois types de relations entre les cas d'utilisations :

- La relation de utilisation :
- La relation d'inclusion : le cas d'utilisation source comprend également le cas d'utilisation destination.
- La relation d'extension : un cas d'utilisation source ajoute son comportement à un cas d'utilisation destination.

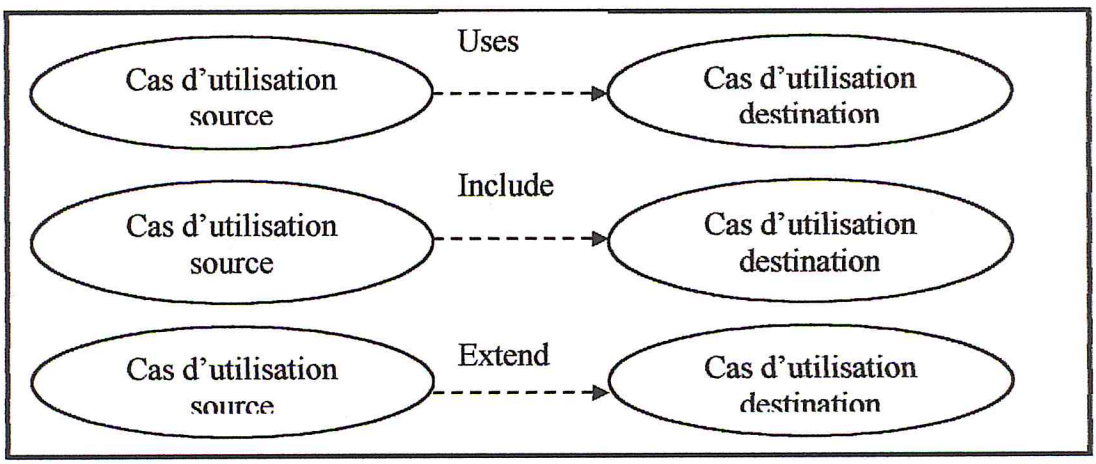


Figure 4-12 : Représentation des relations entre cas d'utilisation

3.3.2. Diagramme de classe

➢ un diagramme de classe est une collection d'éléments de modèle (statique), tel des classes, des interfaces et leurs relations.

- Une classe est une représentation d'un ensemble d'élément partageant les mêmes attributs, les mêmes opérations, les même relations.

• Une interface fournit une vue totale ou partielle d'un ensemble de services offerts par une classe, un paquetage, ou un composant. Les éléments qui utilisent l'interface peuvent exploiter tout ou partie de l'interface.

3.3.3. Diagramme d'objet

➢ Un diagramme d'objet permet de représenter les relations existantes entre les différentes instances des classes à un instant donné de la vie du système.

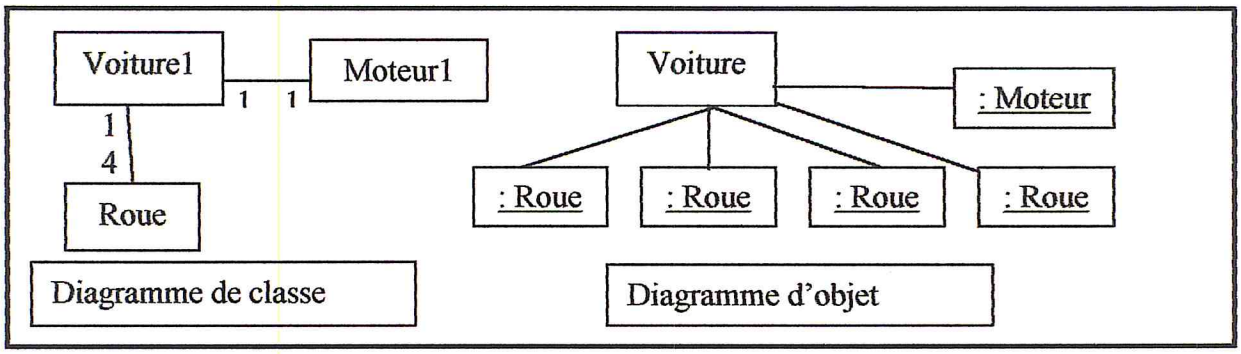


Figure 4-13 : Représentation d'un diagramme d'objet et de classe

3.3.4. Diagramme de composant

➢ Un diagramme de composant permet de décrire les composants et leurs dépendances dans leur environnement d'implémentation.

• Un composant est un élément physique qui représente une partie implémentée du système. Il peut être un code, un script. Il représente un ensemble d'interfaces.

• Une interface de composant est un élément définissant les comportements des composants.

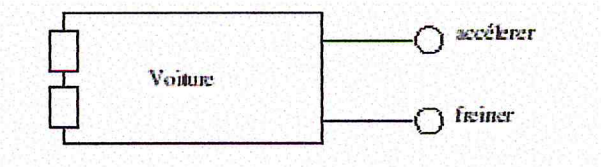


Figure 4-14 : Représentation de composant et d'interface

La majore partie des relations entre composant est constituée de contraintes de compilation et d'édition de liens.

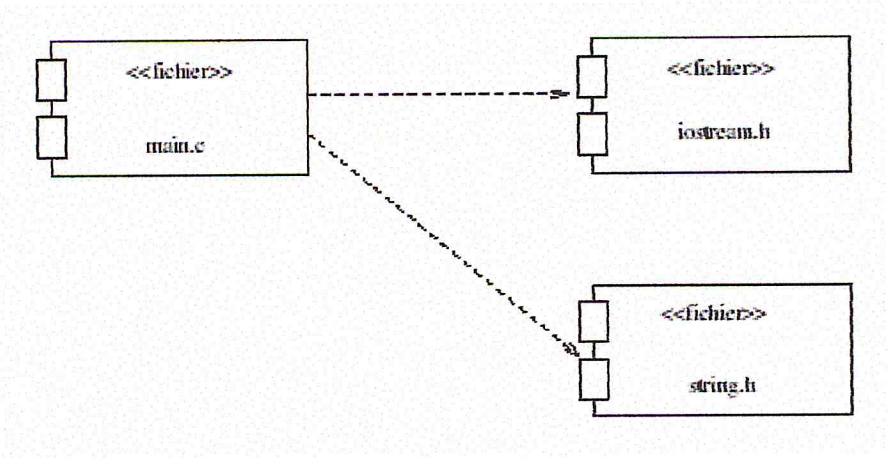


Figure 4-15 : Représentation des relations entre composants

3.3.5. Diagramme de déploiement

➢ Un diagramme de déploiement permet de monter la disposition physique des matériels qui composent le système, ainsi que la répartition des composants sur ses matériels représentés par des noeuds. -

- Un nœud est une ressource matérielle du système étudier.

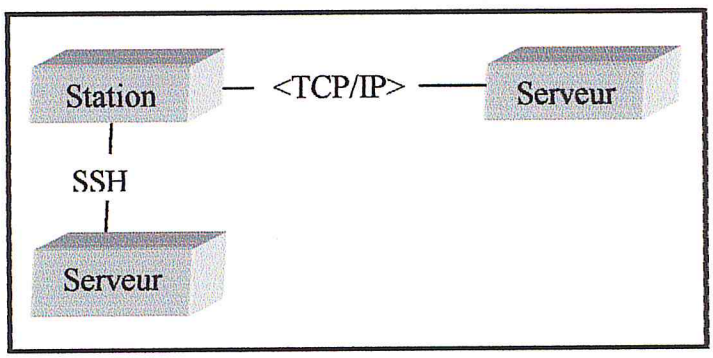


Figure 4-16 : Représentation des relations entre noeuds

3.3.6. Diagramme de séquence

➢ Un diagramme de séquence montre les interactions entre les objets selon la chronologie des envois de messages.

- Une interaction modélise un comportement dynamique entre objets. Elle se traduise par envoi de message entre objets.
- Un message est représentation de communication dans laquelle des informations sont échangées.

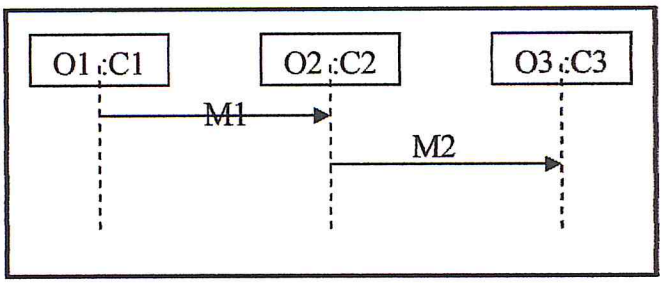


Figure 4-17 : Représentation de séquence

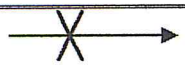
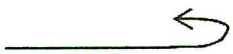
Type de message	La signification
	Message simple, exemple : à une passation de contrôle en mono tâche.
	Message asynchrone, pas de réponse attendue par l'émetteur.
	Message synchrone, réponse nécessaire du destinataire
	Message dérobant, le destinataire doit être à l'écoute
	Message minuté, l'émetteur est bloque pendant un laps de temps.

Figure 4-18 : Représentation des types de message

3.3.7. Diagramme de collaboration

➢ un diagramme de collaboration met en évidence l'organisation structurelle des objets qui envoient et reçoivent des messages.

- Une collaboration définit une interaction, elle constitue une société de rôles et de divers éléments qu travaillent ensemble pour fournir un comportement coopératif.

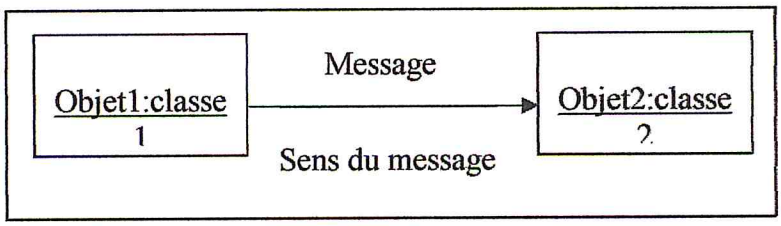


Figure 4-19 : Représentation de collaboration entre deux objets

3.3.8. Diagramme d'état de transition

➢ un diagramme d'état de transition est un graphe orienté états (noeuds) connectés par des transitions (arc orientés). Il décrit l'évolution au cours du temps d'une instance d'une classe en réponse aux interactions avec d'autres objets.

- Etat est la situation dans laquelle un objet se trouve à un instant donné. Un objet peut être en état initial, intermédiaire et final.
- Transition est une relation entre 2 états indiquant qu'un objet dans le premier état va exécuter une action et entrer dans le deuxième état quand un événement apparaîtra.

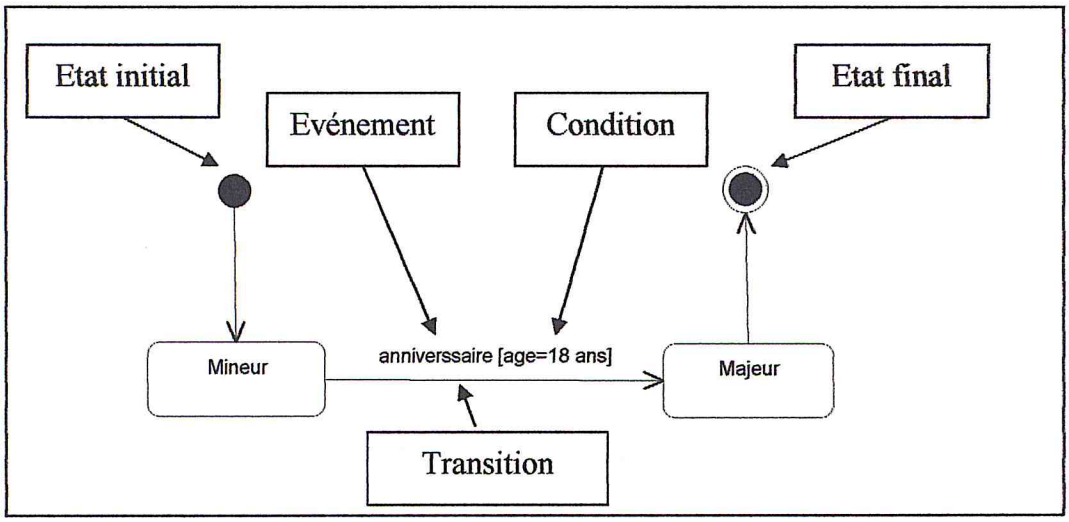


Figure 4-20 : Exemple d'état de transition

3.3.9. Diagramme d'activité

➢ un diagramme d'activité représente le déroulement d'une opération ou d'un cas d'utilisation, ce diagramme met l'accent sur les activités, leurs relations et leurs impacts sur les objets.

- Activité est une suite d'opération qui s'effectue dans un état.

Ce diagramme permet de décrire le déroulement d'un cas d'utilisation particulier. Il est possible de décrire les acteurs responsables de chaque activités par l'utilisation des «couloirs d'activités» qui permettent de répartir graphiquement les différentes activités entre les acteurs opérationnels. Chaque activité est placée dans le «couloir» correspondant à l'acteur qui assume cette activité. [Muller 01].

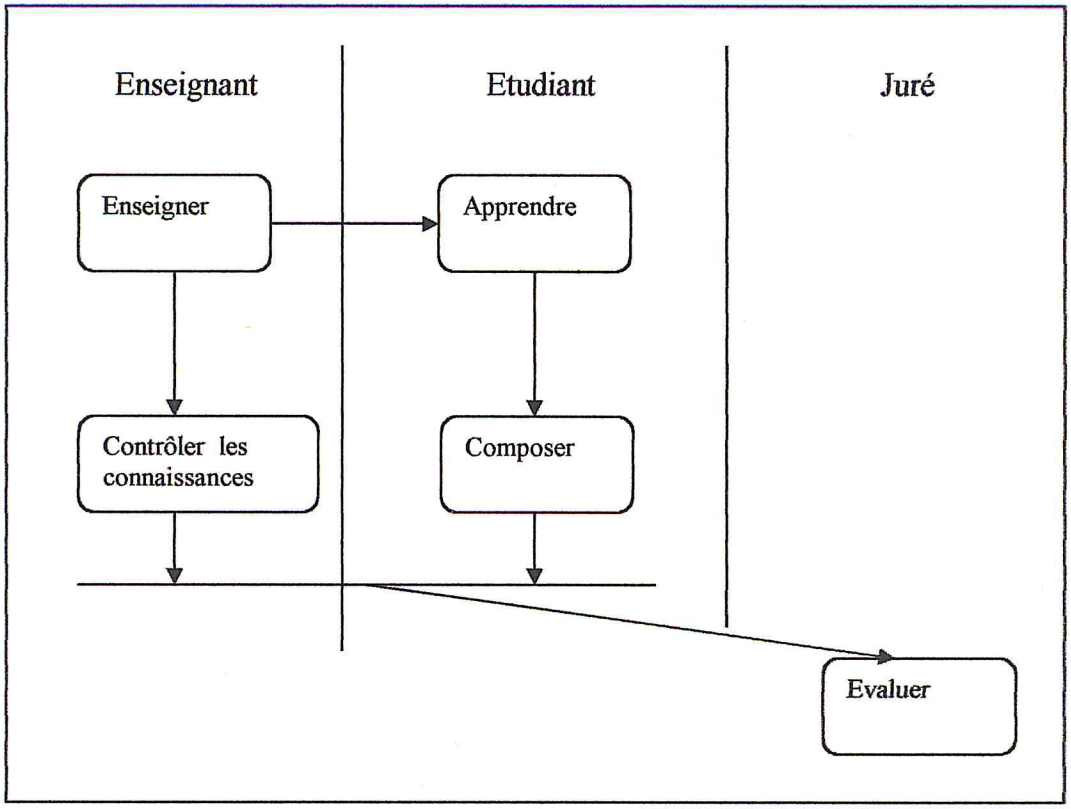


Figure 4-21 : Exemple de diagramme d'activité

4. Conclusion

Dans ce chapitre, nous avons présenté l'architecture de notre système de gestion de la qualité, encore on a exposé les notions de base du formalisme de modélisation UML.

Notre système de gestion de la qualité est basé sur trois modules : Evaluation des processus logiciels, Amélioration des processus enfin Evaluation de capacité des processus améliorés.

UML est un langage de modélisation orienté objet, il permet de représenter les systèmes par des éléments de base et spécifier leurs comportements sous formes de diagrammes.

Dans le chapitre suivant, nous utiliserons UML pour le développement du système de gestion de la qualité du logiciel.

CHAPITRE V

Développement du système de gestion de la qualité

3. Spécification des besoins

L'étape de la spécification est fondée sur la question *Quoi ?* En effet avant tout, il faut spécifier se qu'on doit faire : quelles sont les fonctionnalité à réaliser ? Quelles sont les contraintes à respecter ? La réponse à ces deux questions permet de définir ce que le système doit faire.

Pour ce faire, UML propose les diagrammes de cas d'utilisation, les diagrammes de séquences.

3.1. Diagramme de cas d'utilisation

Les diagrammes de cas d'utilisation représentent les fonctions du système du point de vue de l'utilisateur. L'étude des cas d'utilisation débute par la détermination des acteurs (utilisateurs) du système puis les cas d'utilisation (fonctions).

3.1.1. Les acteurs

Les acteurs sont les utilisateurs directs du système. Pour le notre on a :

- Le manager : est un utilisateur chargé de définir les modules et les processus selon CMM.

3.1.2. Les cas d'utilisation

Un cas d'utilisation est un ensemble de fonctions réalisés par le système pour un acteur particulier. Les différents cas d'utilisation recensés pour le système sont :

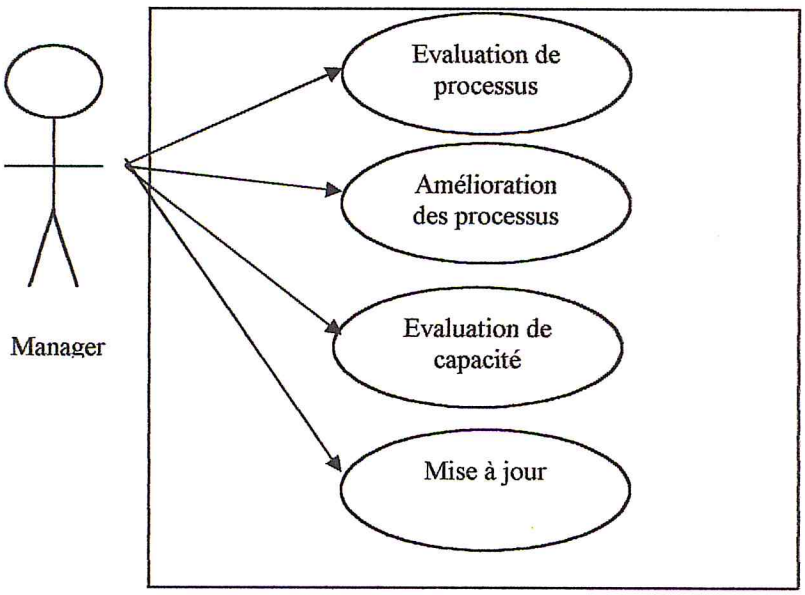


Figure 5-2 : Cas d'utilisation global

3.2.1. Evaluation de processus

Ce cas d'utilisation utilise deux cas d'utilisation : " définition du questionnaire" et "définition du processus", exprimé par la relation « uses ».

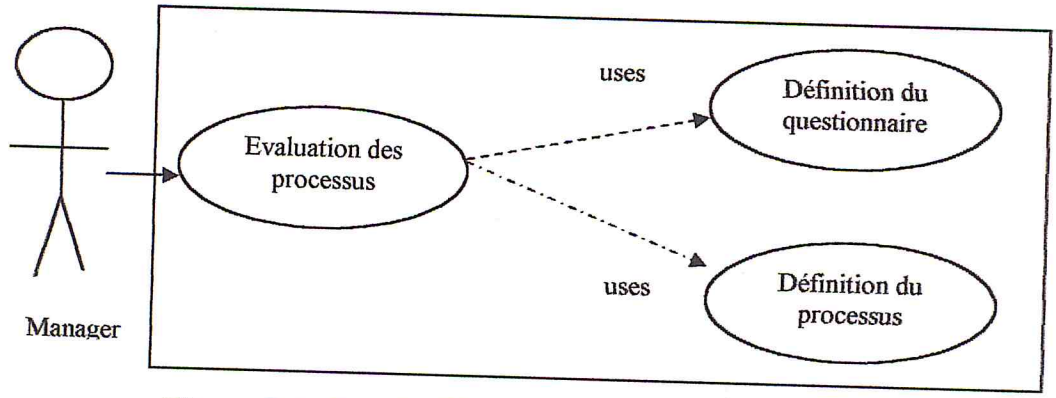


Figure 5-3 : Cas d'utilisation "évaluation de processus"

3.2.2. Définition du processus

Pour définir un processus, il faut définir : le groupe qui le réalise, les documents et les objectives associés ainsi les activité réalisés. Le cas d'utilisation "définition du processus" inclut les trois cas : définition de processus, de l'organisation et du SI, cette inclusion est exprimé par la relation <<include>>.

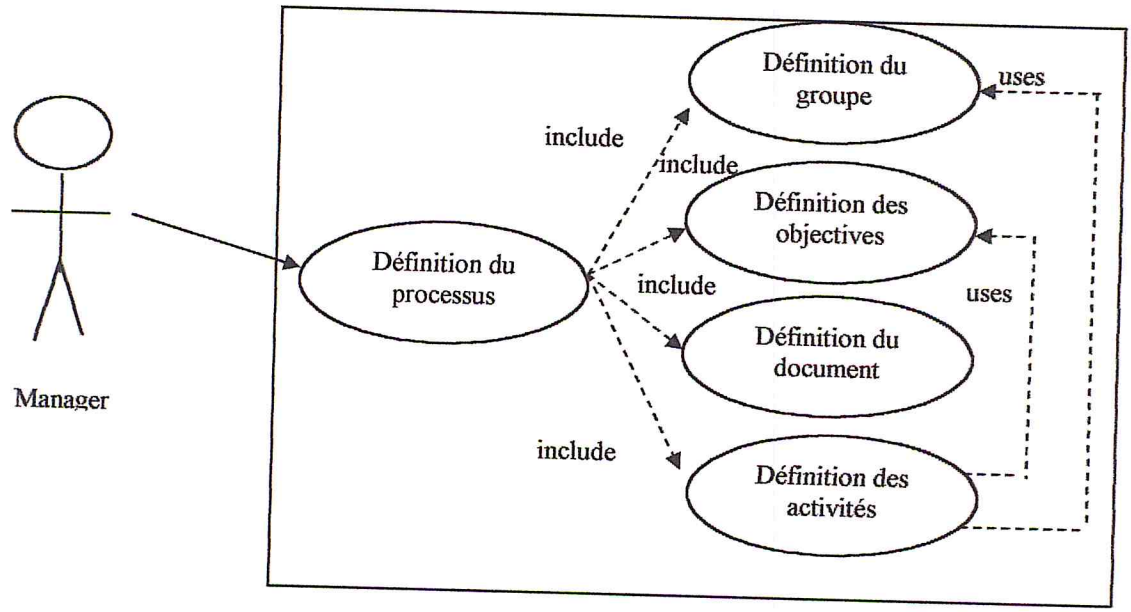


Figure 5-4 : Cas d'utilisation "définition du processus"

3.2.3. Amélioration de processus

Ce cas d'utilisation utilise trois cas d'utilisation : ''définition du processus'', ''planification des processus'' et ''mise en place des processus'', exprimé par la relation « uses ».

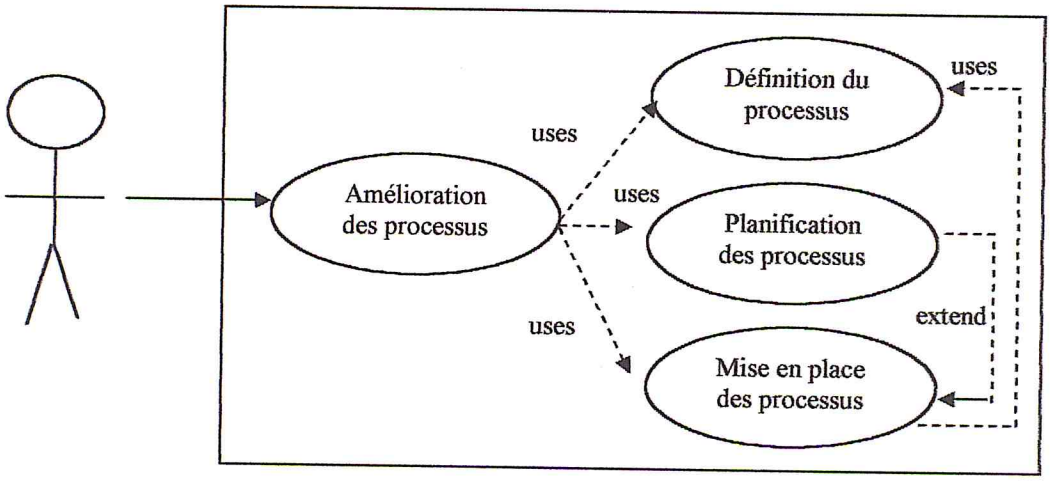


Figure 5-5 : Cas d'utilisation ''amélioration du processus''

3.2.4. Mise à jour

La mise à jour inclus la mise à jour des processus, du questionnaire, de la planification et des instruments de mesures.

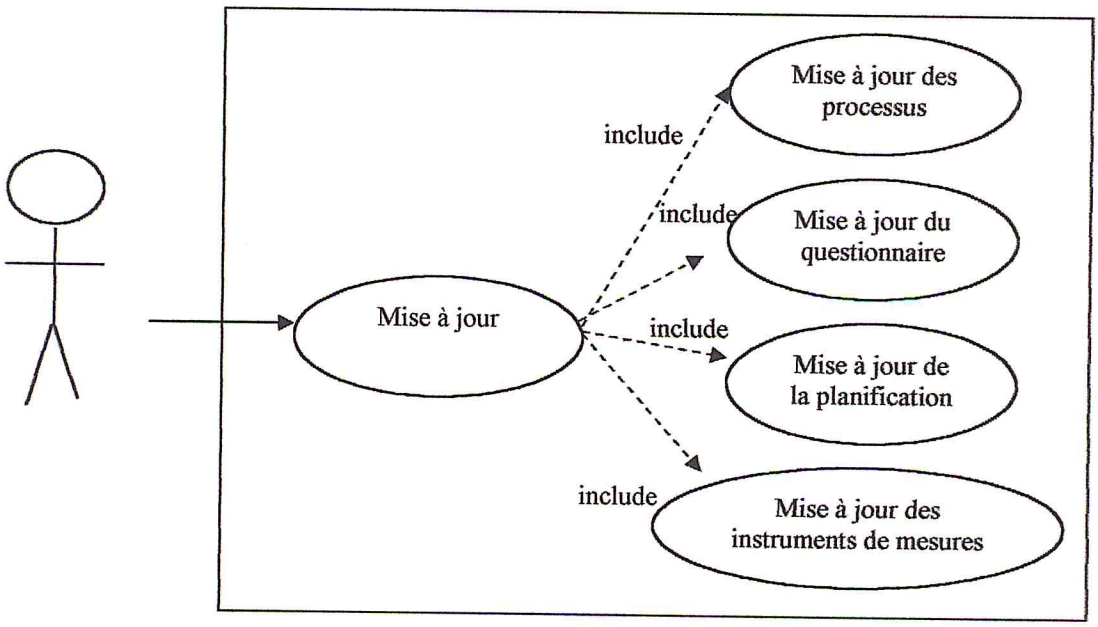
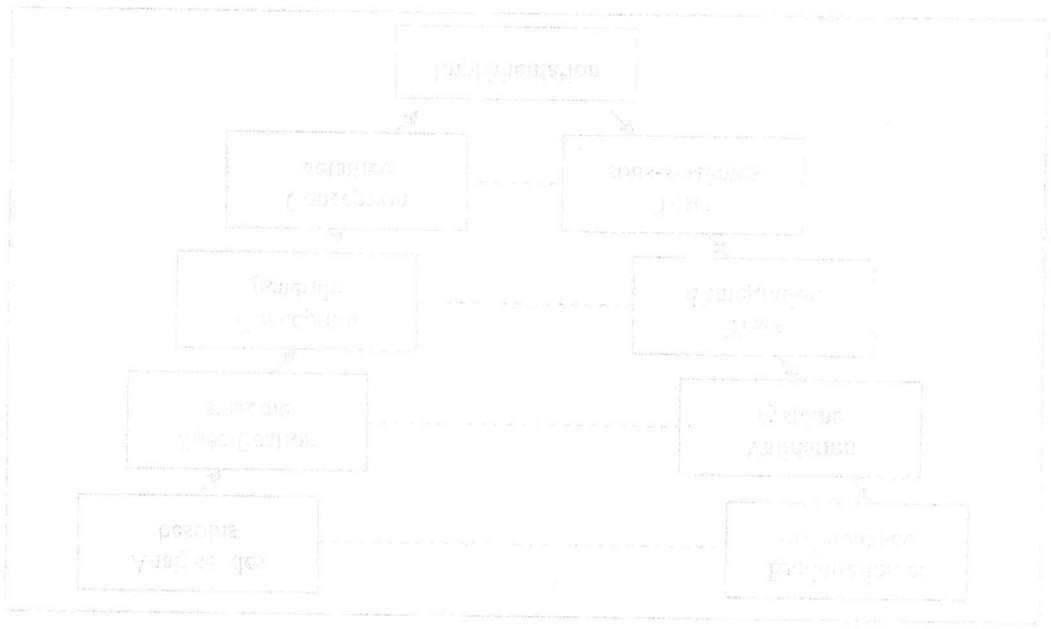


Figure 5-6 : Cas d'utilisation ''mise à jour''



La figure 5-7 montre l'ensemble des fonctionnalités offertes par notre système :

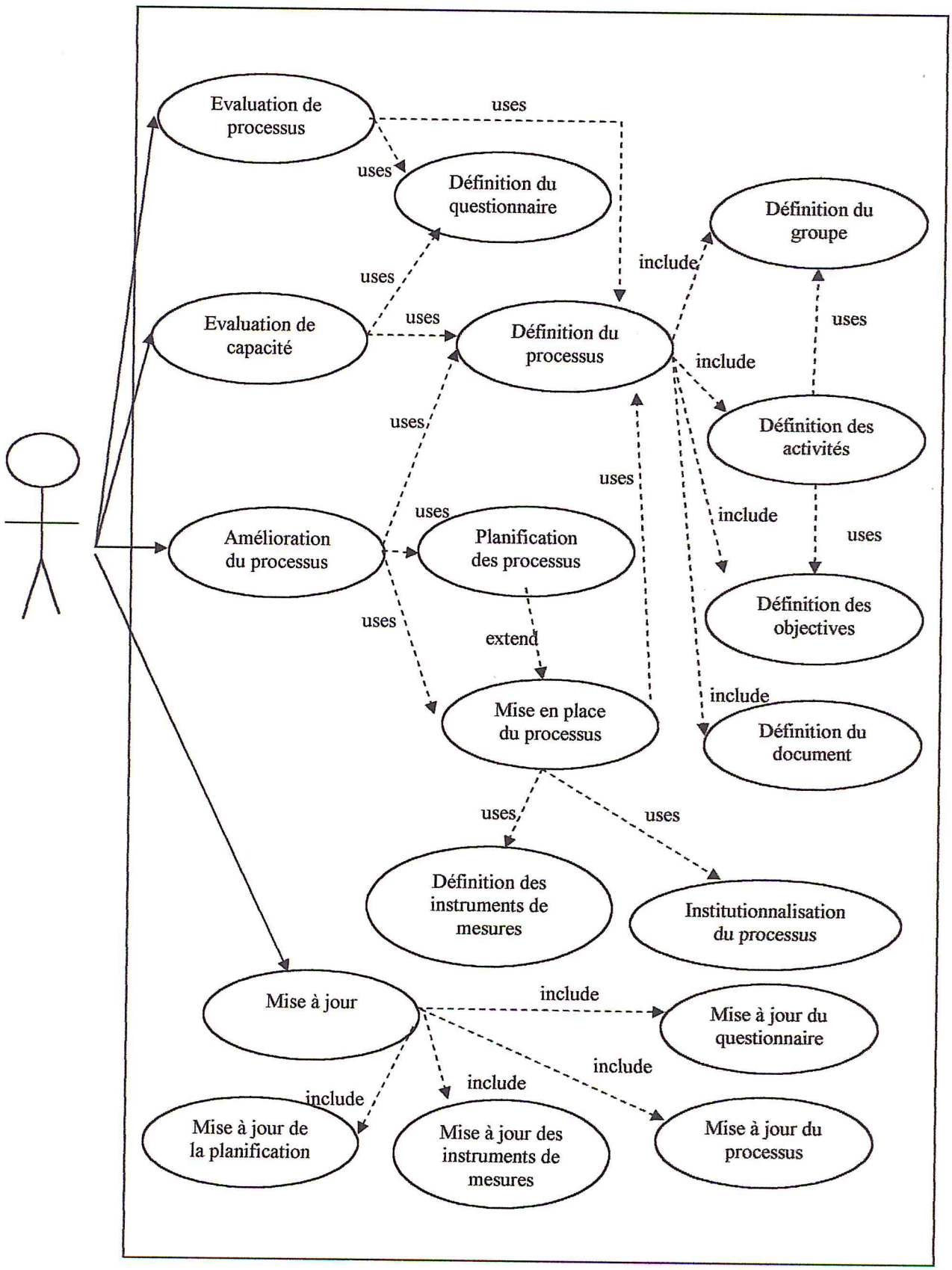


Figure 5-7 : Cas d'utilisation du système de gestion de la qualité du logiciel

3.2 Diagrammes de séquences

Le diagramme de cas d'utilisation est une représentation spatiale du fonctionnement du système. Le diagramme de séquence représente un scénario (instance de cas d'utilisation) d'une manière temporelle et montre les interactions entre les objets.

3.2.1 Evaluation de capacité

- **Evaluation du processus**

Scénario : « Evaluation du processus »

- Le manager exécute le scénario « définition du processus »
- Le manager exécute le scénario « définition du questionnaire »
- Le manager demande l'affichage du questionnaire
- Le système lui affiche le questionnaire
- Le manager répond au questionnaire
- Le système affiche le résultat de l'évaluation

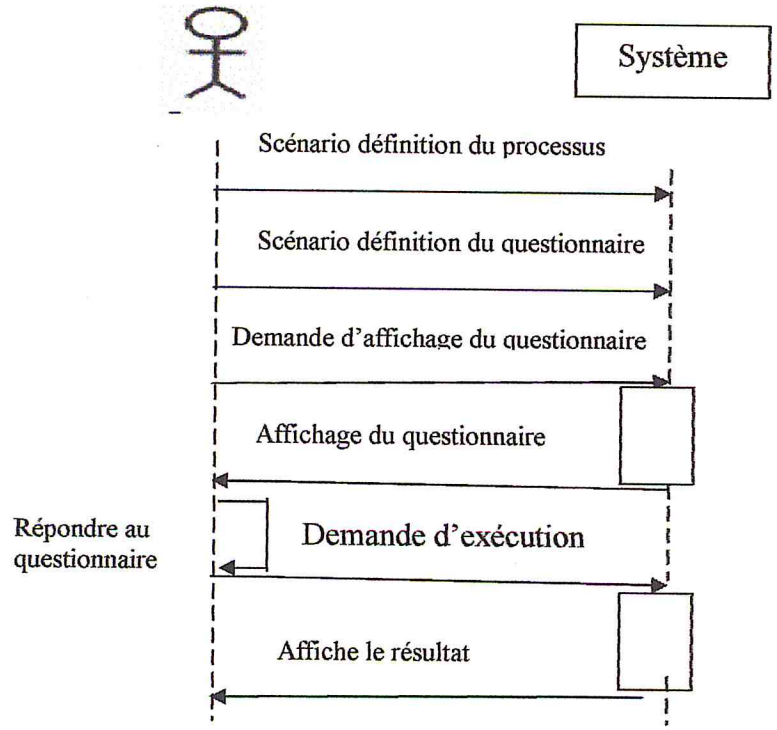


Figure 5-8 : Diagramme de séquence ''évaluation du processus''

- **Définition du processus**

La définition du processus consiste à définir le groupe qui le réalise, les objectifs du processus, les activités réalisées selon les objectifs ainsi que les documents appropriés.

Scénario : « définition du groupe »

- Le manager ajoute un groupe
- Le système lui affiche les propriétés du groupe
- Le manager remplit les propriétés du groupe

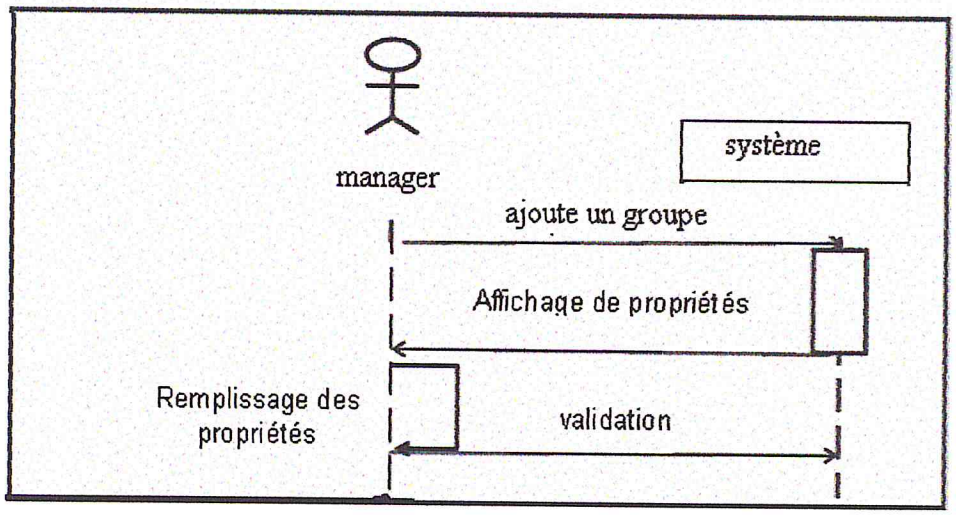


Figure 5-9 : Diagramme de séquence ''définition du groupe''

Scénario : « définition des objectifs »

- Le manager ajoute un objectif
- Le système lui affiche les propriétés de l'objectif
- Le manager remplit les propriétés de l'objectif

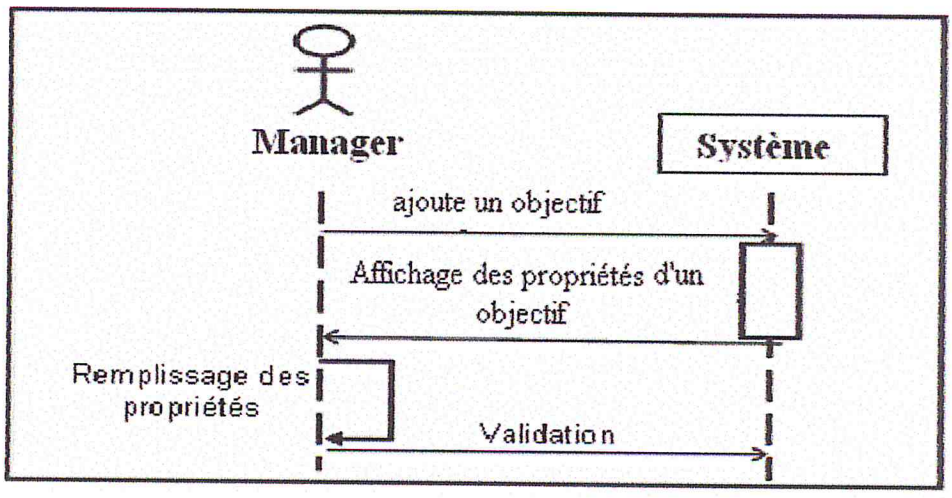


Figure 5-10 : Diagramme de séquence ''définition d'un objectif''

Scénario : « définition d'une activité »

- Le manager ajoute une activité
- Le système lui affiche les propriétés de l'activité
- Le manager remplit les propriétés de l'activité

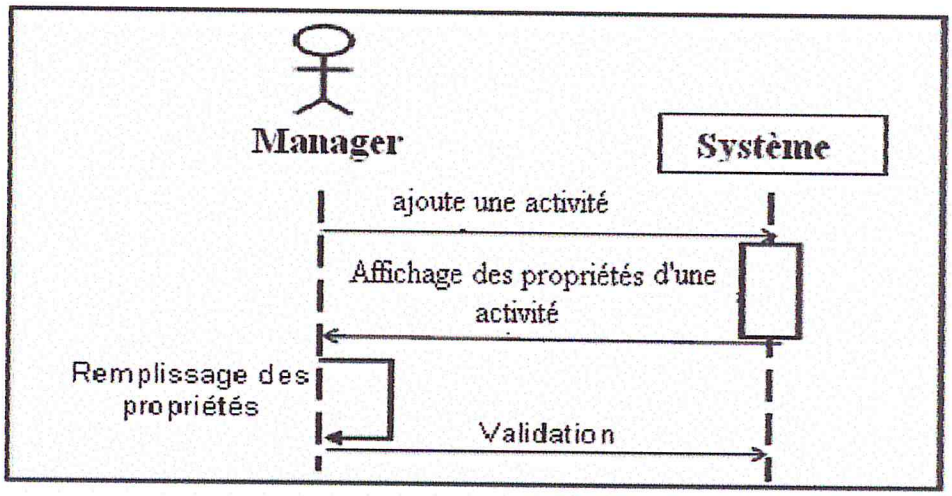


Figure 5-11 : Diagramme de séquence ''définition d'une activité''

Scénario : « définition d'un document »

- Le manager ajoute d'un document
- Le système lui affiche les propriétés du document
- Le manager remplit les propriétés du document

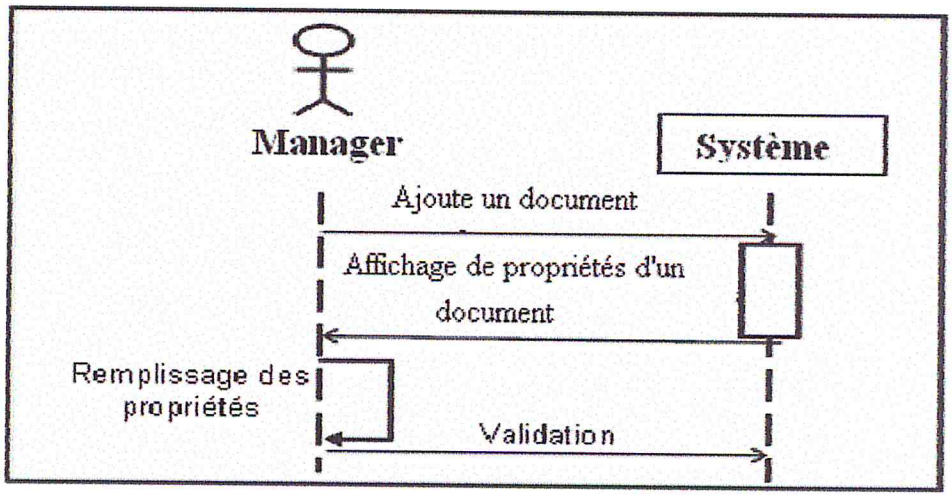


Figure 5-12 : Diagramme de séquence ''définition d'un document''

- **Définition du questionnaire**

Scénario : « définition du questionnaire »

- Le manager ajoute une nouvelle question
- Le système lui affiche les propriétés de la question
- Le manager remplit les propriétés

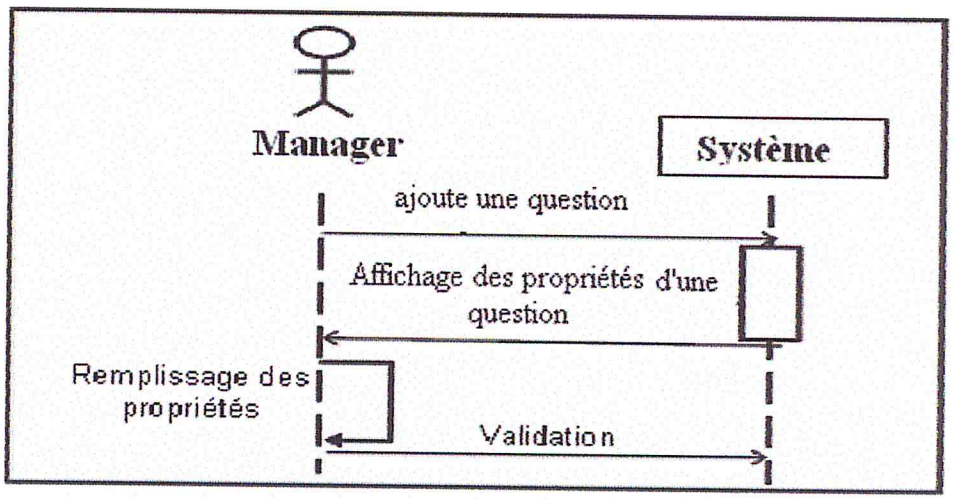


Figure 5-13 : Diagramme de séquence ''définition d'une question''

3.2.2 Amélioration des processus

- **Définition du processus**

Scénario : « définition du processus »

Ce scénario est montré précédemment.

- **Planification des processus**

Scénario : « Planification des processus »

- Le manager ajoute une planification
- Le système lui affiche la fiche de planification
- Le manager remplit la fiche de planification

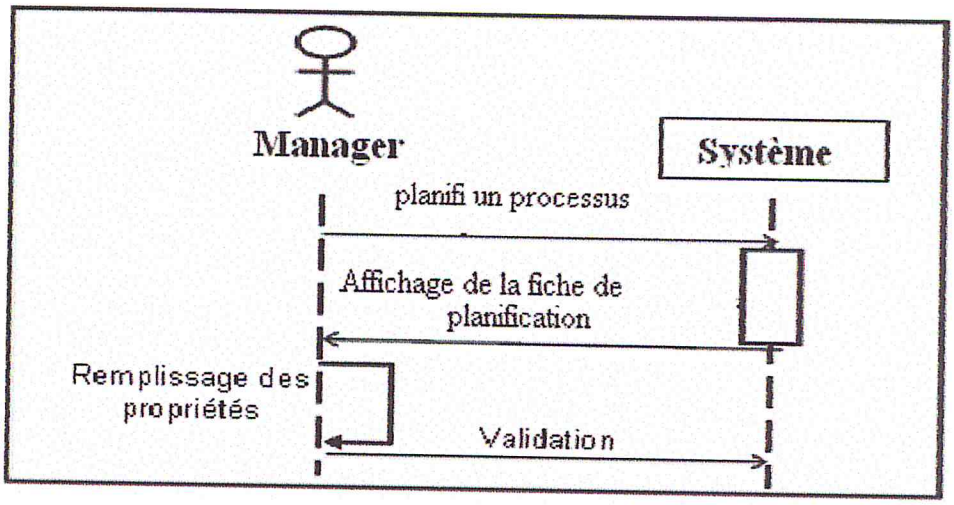


Figure 5-14 : Diagramme de séquence ''Planification de processus''

- **Mise en place du processus**

Scénario : « définition du processus »

Ce scénario est montré précédemment.

Scénario : « définition des instruments de mesures »

- Le manager ajoute un instrument de mesure
- Le système lui affiche les propriétés de l'instrument de mesure
- Le manager remplit les propriétés

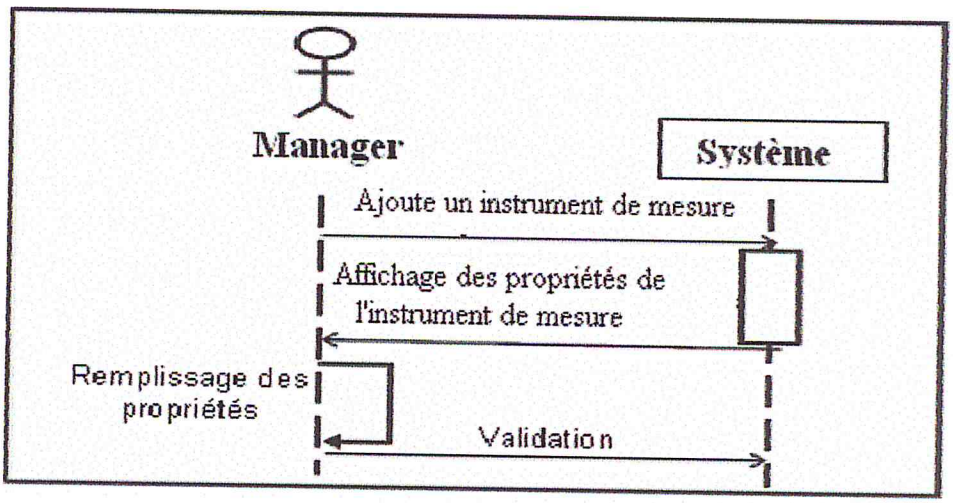


Figure 5-15 : Diagramme de séquence ''définition d'une mesure''

3.2.3 Evaluation de capacité

- **Evaluation de capacité de processus**

Scénario : « Evaluation de capacité de processus »

- Le manager exécute le scénario « définition du processus »
- Le manager exécute le scénario « définition du questionnaire »
- Le manager demande l'affichage du questionnaire
- Le système lui affiche le questionnaire
- Le manager répond au questionnaire
- Le système affiche le résultat de l'évaluation de capacité
- Le manager valide les améliorations

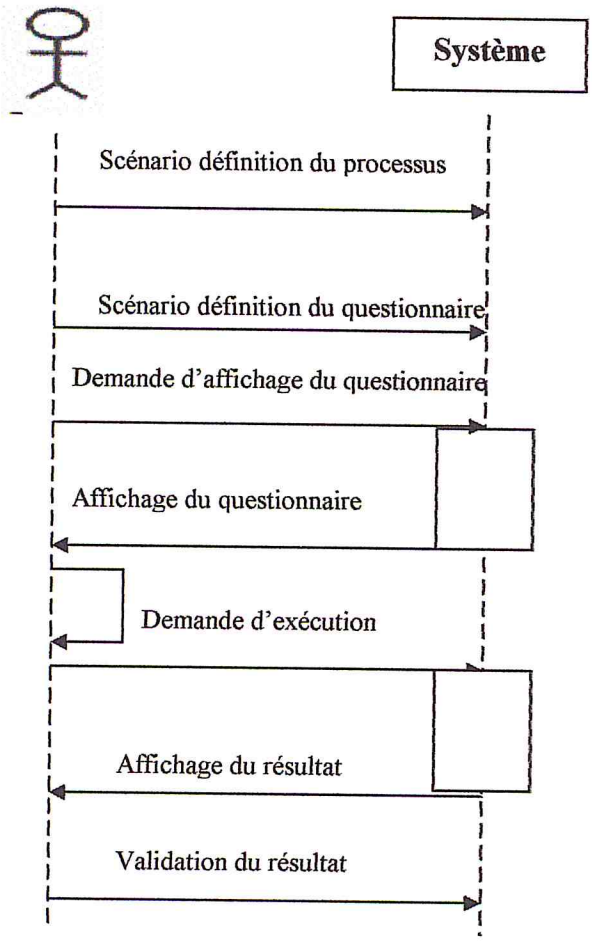


Figure 5-16 : Diagramme de séquence "évaluation de capacité des processus"

4. L'analyse

L'étape de l'analyse doit fournir une approche conceptuelle du problème. En effet, il faut définir une structure adéquate utilisée comme base pour la construction du système.

L'objectif de cette base est de déterminer les éléments intervenant dans le système à construire, ainsi que les structures et les leurs relations.

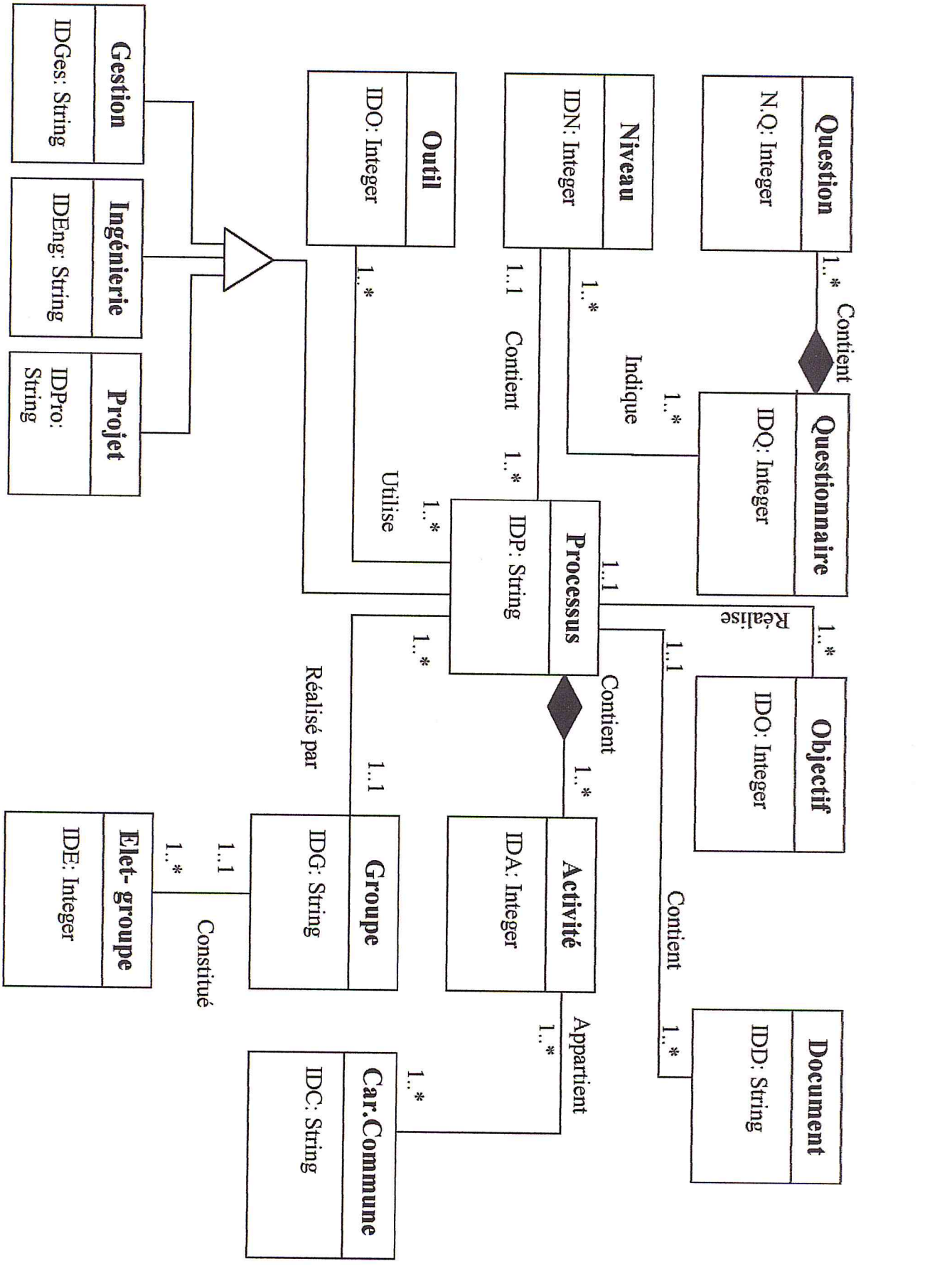
Pour cela, UML propose les diagrammes de classes, diagrammes de collaboration, d'activités et d'états de transitions.

4.1. Diagrammes de classes

Le diagramme de classe représente la structure du système sous forme de classes et de relations entre classes.

Après l'étude des différents concepts du modèle SW-CMM aux Chapitre 4, un processus est décrit comme un ensemble d'activité, appartenant à un niveau de maturité, réalisant des objectives, contenant un document et réalisé par un groupe. Le niveau indique la capacité de l'organisme. Un questionnaire cible un processus pour déterminer le niveau de l'organisme, il permis aussi la déterminé la conformité des objectives d'un processus.

Ces concepts nous on permis de présenter le diagramme de classe suivant :


Question
N.Q: Integer
Questionnaire
IDQ: Integer
Contient
1..*
Niveau
IDN: Integer
Indique
Processus
IDP: String
Objectif
IDO: Integer
Réalise
Document
IDD: String
Contient
Outil
IDO: Integer
Utilise
Activité
IDA: Integer
Appartient
Car.Commune
IDC: String
Groupe
IDG: String
Réalisé par
Elet- groupe
IDE: Integer
Constitué
Gestion
IDGes: String
Ingénierie
IDEng: String
Projet
IDPro: String

4.2. Diagramme d'état de transition

Un diagramme d'état de transition est une présentation du comportement d'une classe en terme d'états et de transitions d'états. Il montre l'enchaînement d'une calasse suite à des traitements particuliers.

Nous utiliserons ce diagramme pour représenter les différents états que peuvent prendre un objectif (**Figure5-18**) et une question (**Figure5-19**).

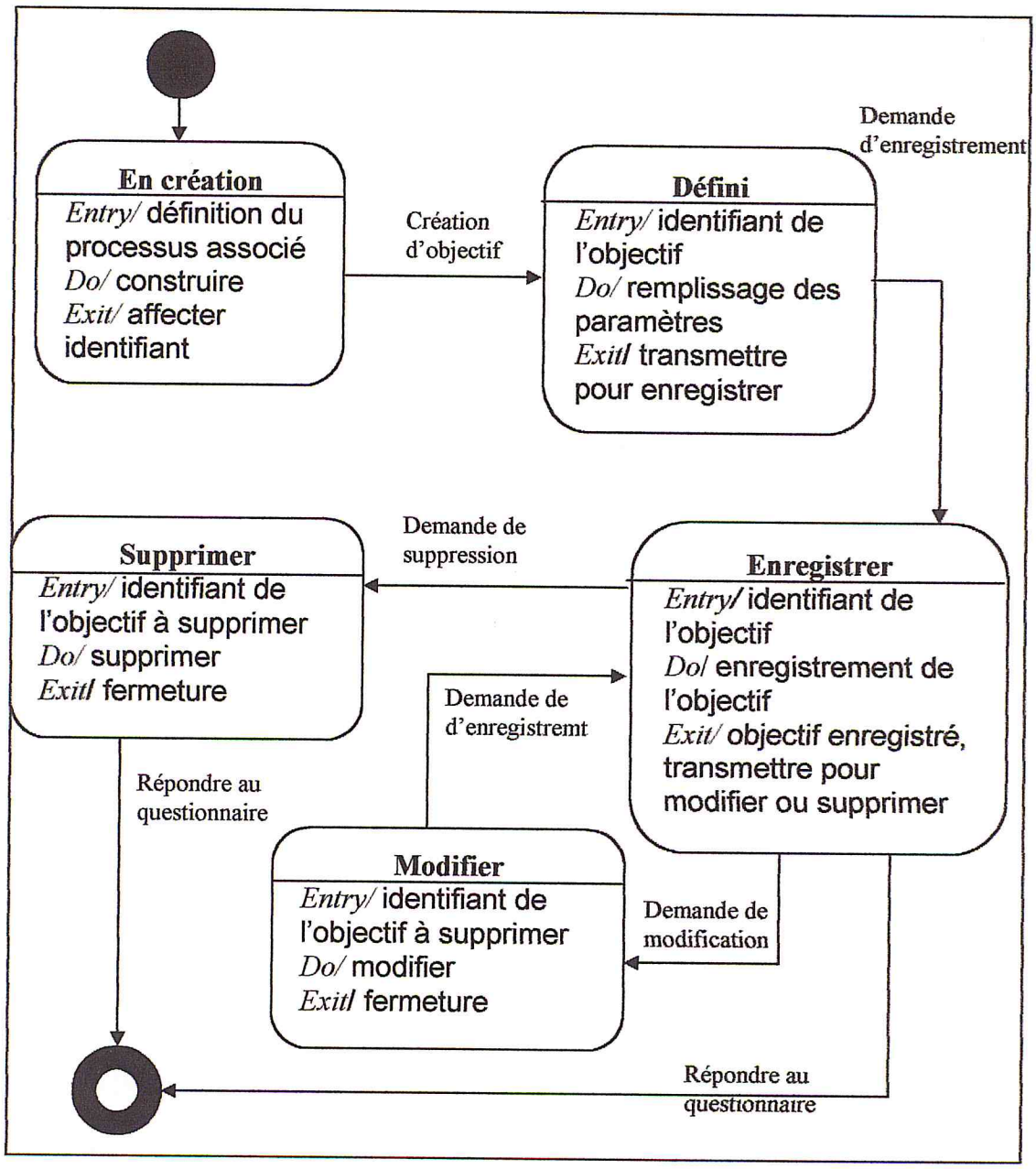


Figure5-18 : Diagramme d'état de transition de la classe objectif

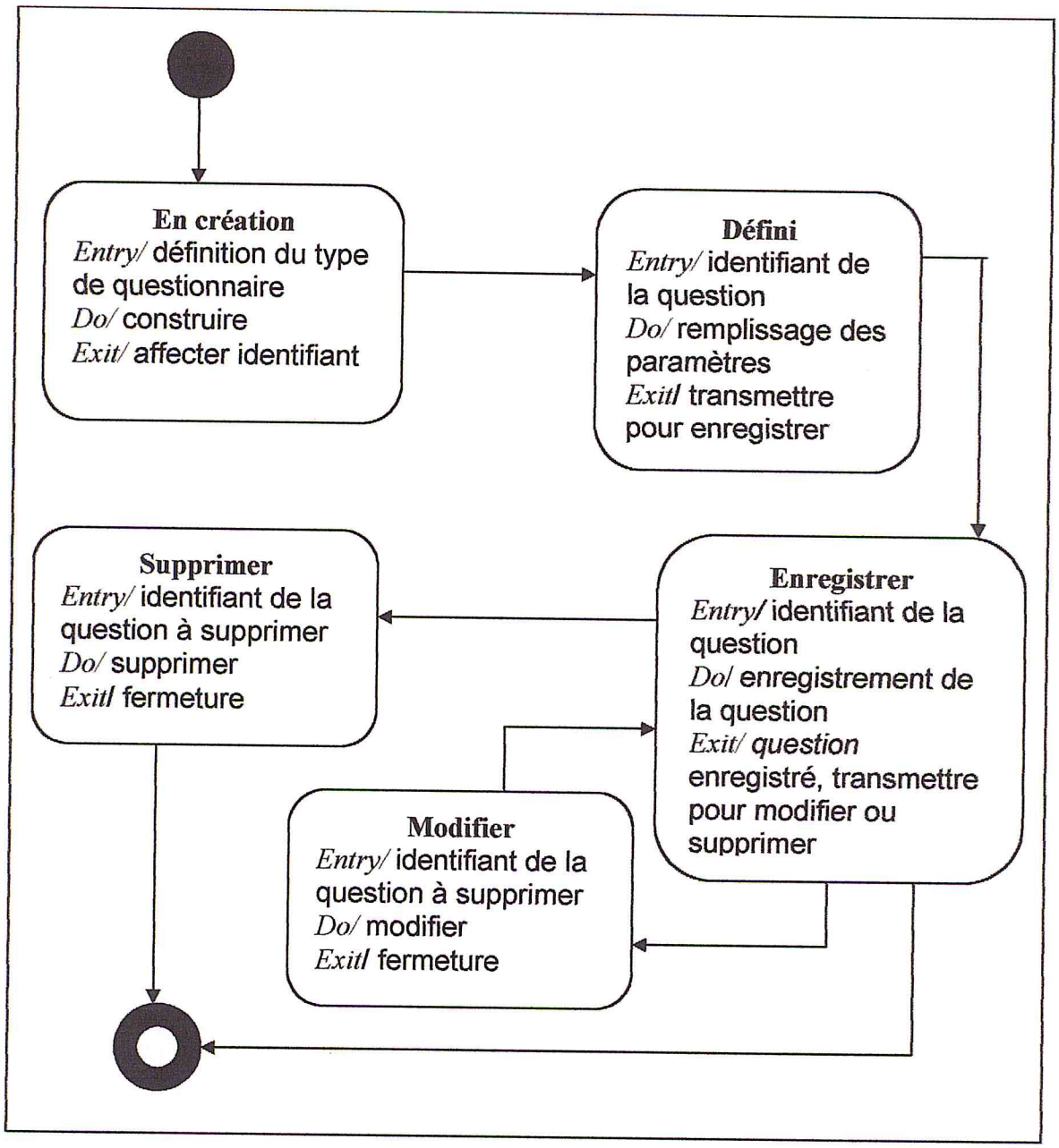


Figure5-19 : Diagramme d'état de transition de la classe question

4.3. Diagramme d'activités

Le diagramme d'activité est une variante des diagrammes d'états de transitions, ce diagramme met l'accent sur les activités, leurs relations et leurs impacts sur les objets.

Le diagramme d'activité suivant décrit le cas d'utilisation "mise à jour d'instrument de mesure".

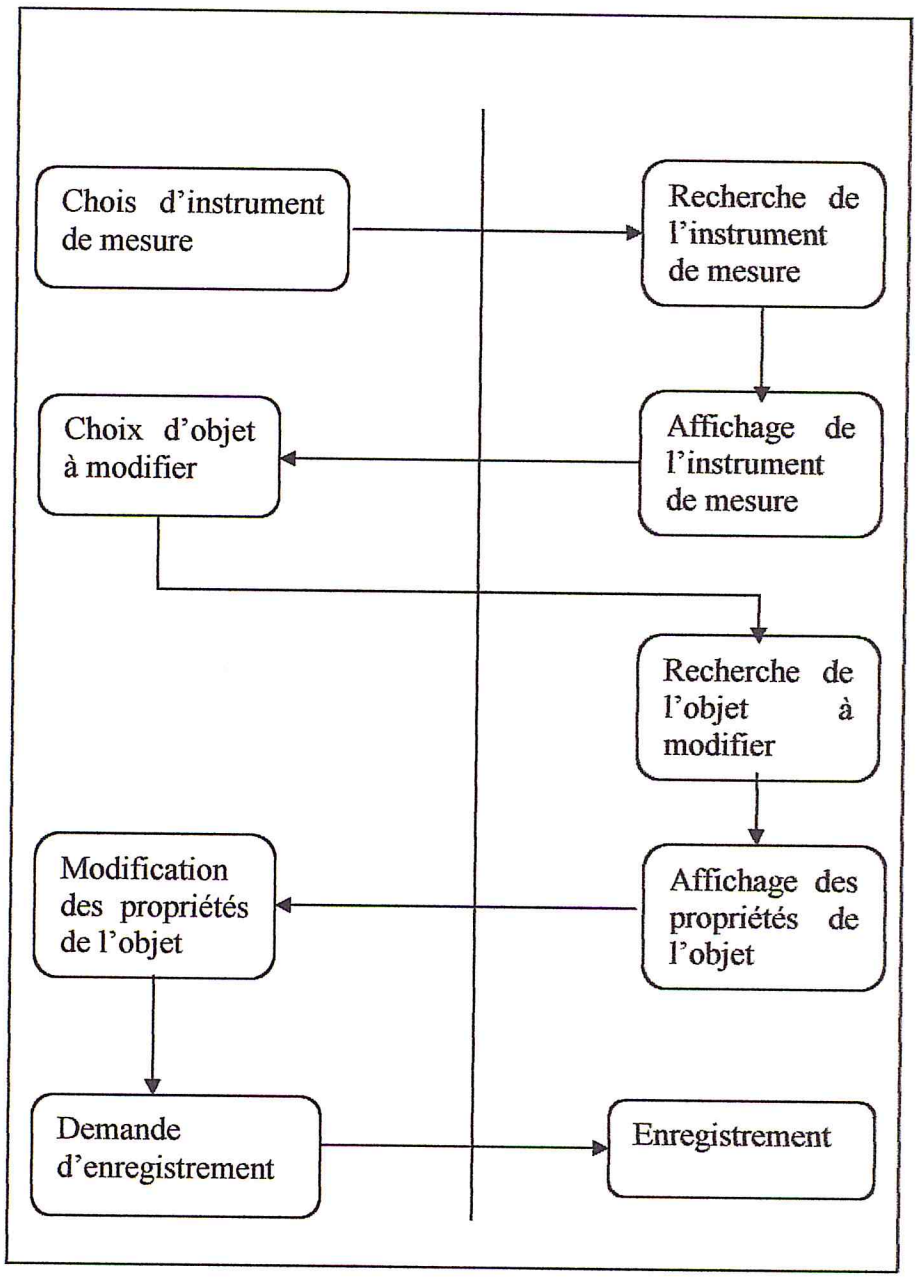


Figure5-20 : Diagramme d'activité mise à jour d'instrument de mesure

4.4. Diagrammes de collaboration

Le diagramme de collaboration décrit le comportement collectif d'un ensemble d'objets, en vue de réaliser une activité en décrivant leurs interactions modélisées par des envois (éventuellement numérotés) de messages.

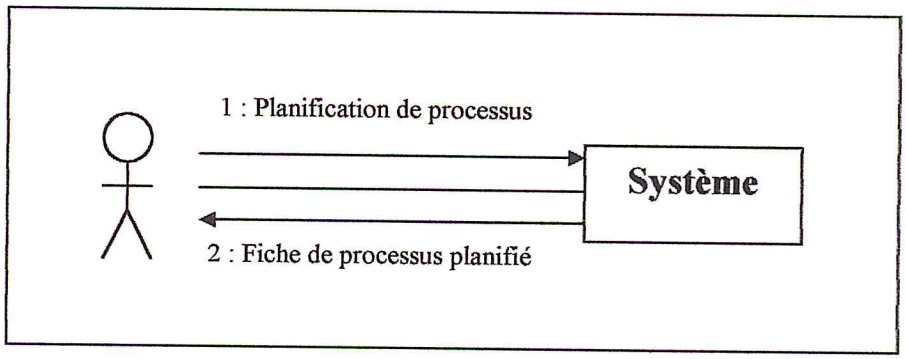


Figure5-21 : Diagramme de collaboration planification de processus

5. Conception

L'objectif de cette étape est de répondre à la question : Comment ? Ceci amène à l'architecture du système et à le diviser en sous ensembles, en prenant en compte la dimension technique des outils informatiques.

Pour réaliser la conception, UML propose les diagrammes de classes, de composants et les paquetages.

La conception se déroule sur deux étapes : la conception générale et la conception détaillée.

La conception générale a pour but de décomposer le logiciel en modules et de préciser les interfaces et les fonctions de chaque module. A l'issue de cette étape, on obtient une description de l'architecture du logiciel et un ensemble de spécifications de ses divers composants. La conception détaillée fournit pour chaque module une description détaillée de la manière dont les fonctions du composant sont réalisées : algorithmes, représentation des données.

5.1. Conception globale

Elle a pour objectif d'étudier l'architecture générale du système par ensembles fonctionnels homogènes a partir des besoins des utilisateurs.

L'architecture du système que nous proposons a été élaborée en vue de suivre les processus, depuis l'évaluation de son existence jusqu'à l'évaluation de sa capacité passant par sa mise en œuvre. La figure suivante donne une vision globale de l'architecture de notre système.

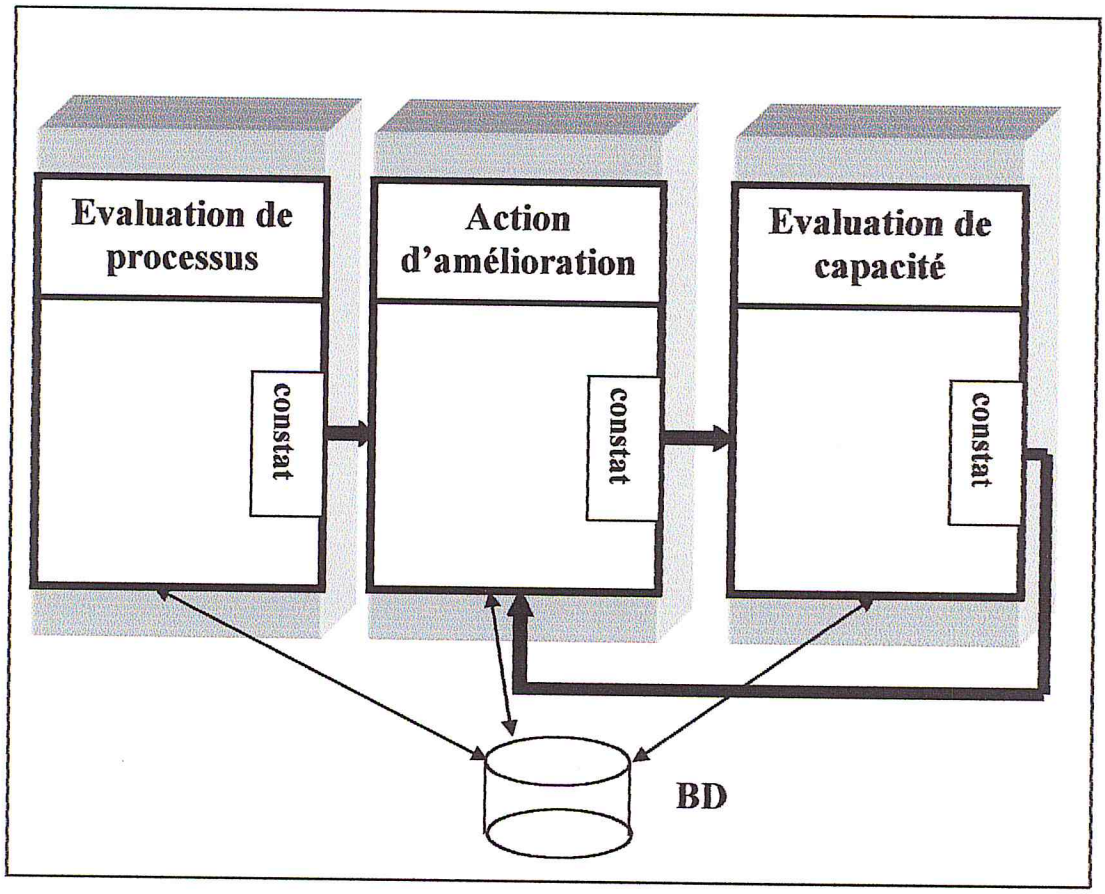


Figure5-22 : Architecture du système de gestion de la qualité

Afin de mettre en œuvre notre outil, nous avons défini une architecture logicielle composée de trois (3) modules:

- Module 1 : Evaluation des processus.
- Module 2 : Action d'amélioration.
- Module 3 : Evaluation de la capacité des processus.

Le module d'évaluation de capacité consiste à établir l'état des lieux, décrire la maturité des processus à travers les besoins propres de l'entreprise.

Un constat décrivant le niveau de maturité de l'entreprise, une liste des forces et faiblesses et des contraintes organisationnelles, ainsi des recommandations sur les éventuelles amélioration à apporter (classés par priorité).

Le module Action d'amélioration consiste à mettre en place l'amélioration des processus tel décrit par CMM.

Le CMM s'avère particulièrement utile au niveau de la planification des actions, de l'exécution des plans d'action et de la définition des processus.

Le module d'évaluation de la capacité logiciel se déroule dans un contexte plus axé sur l'audit.

L'accent est mis sur la création d'une piste de vérification documentée décrivant le processus logiciel réellement mis en oeuvre par l'organisation.

5.2. Conception détaillé

Elle a pour objectif d'étudier l'architecture technique du système par modules conformément au découpage réalisé lors de la conception générale.

5.2.1. Module évaluation des processus

Ce module décrit le niveau de maturité des processus de l'organisme. C'est un processus, alors il contient les notions de groupe, de document et d'activités.

Ce module utilise un questionnaire de maturité pour effectuer l'évaluation.

Un constat d'évaluation comportant le niveau de maturité et les recommandations d'amélioration est établi.

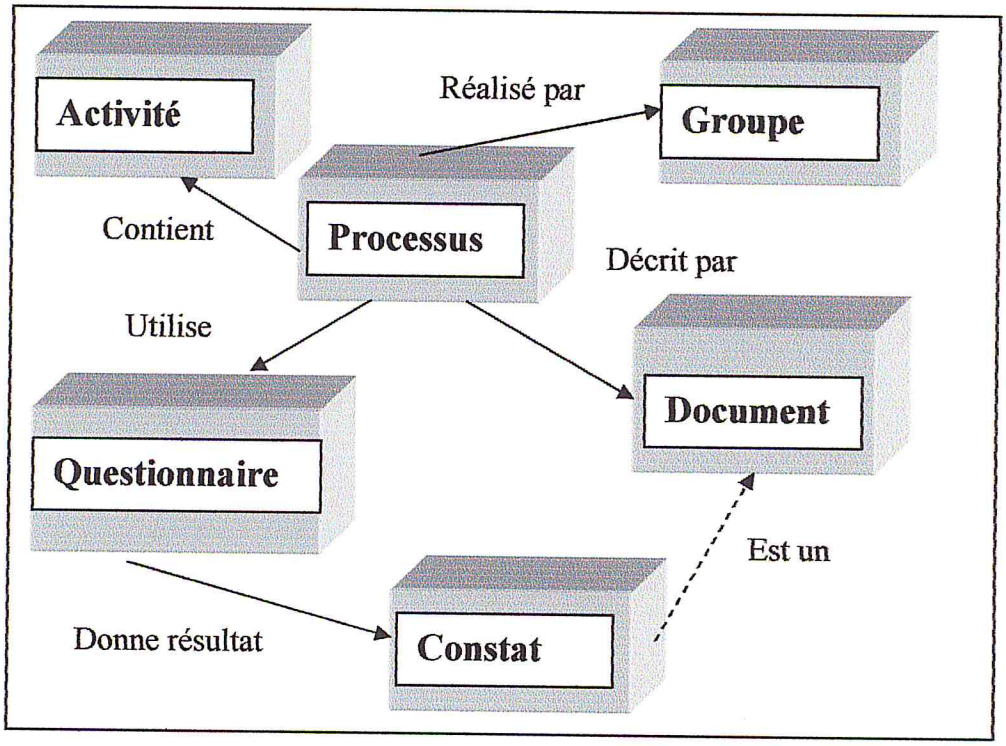


Figure5-23 : Module évaluation des processus

5.2.1. Module action d'amélioration

Ce module met en place l'amélioration des processus tel décrit par CMM. C'est un processus, alors il contient les notions de groupe, de document et d'activités.

Ce module est répartie en deux phases : planification des processus et la mise en œuvre de ses derniers.

Le CMM s'avère particulièrement utile au niveau de la planification des actions, de l'exécution des plans d'action et de la définition des processus

Un constat comportant les processus réalisés est établi.

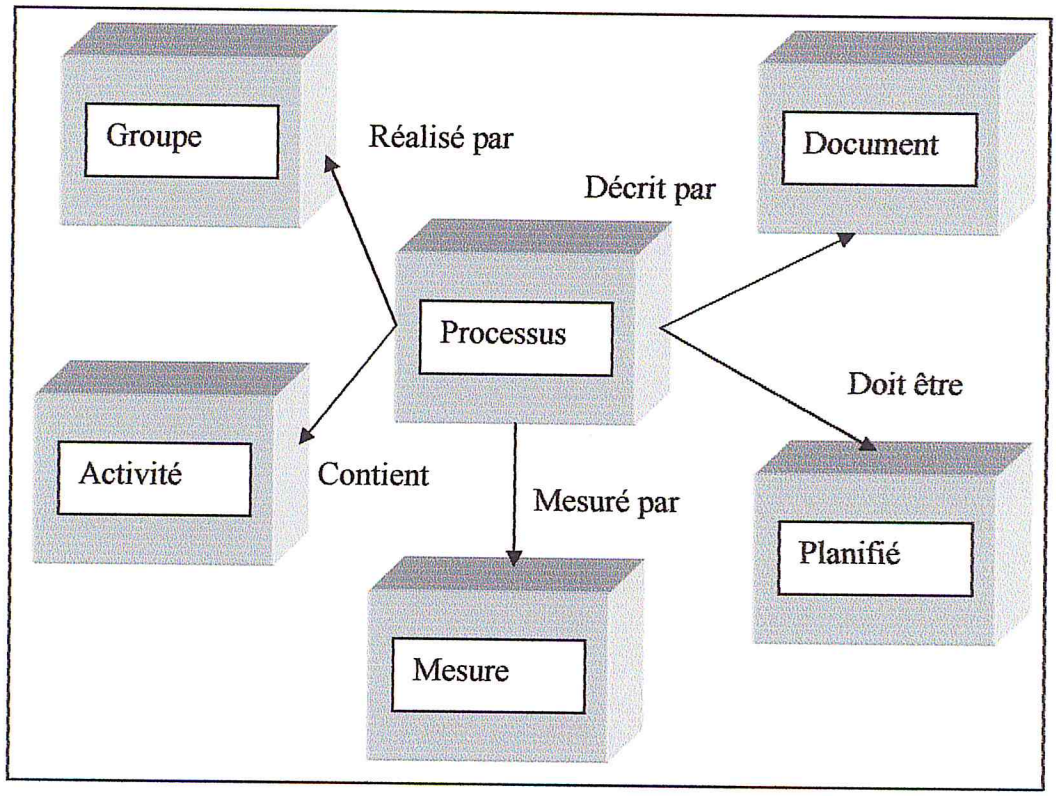


Figure5-24 : Module action d'amélioration

5.2.1. Module évaluation de capacité

Le module d'évaluation de la capacité logiciel se déroule dans un contexte plus axé sur l'audit.

C'est un processus, alors il contient les notions de groupe, de document et d'activités.

Ce module utilise un questionnaire de capacité pour effectuer l'évaluation.

Un constat d'évaluation comprend les processus logiciel réellement mis en oeuvre par l'organisation.

6. Implémentation

Cette étape consiste à traduire en code, compréhensible par un ordinateur, l'architecture technique conçue lors de l'étape de conception.

➢ Langage de programmation

Dans notre cas le langage de programmation devait être choisi parmi les langages suivants : Delphi, C++, java et visual Basic. Notre choix s'est porté sur le langage Delphi parce que c'est un langage orienté objet, et la programmation dans un langage objet est la manière la plus commode de traduire une conception objet en un langage exécutable par machine.

Traditionnellement, Delphi et les bases de données font bon 'ménage'. Delphi est connu pour être l'outil idéal pour développer des frontaux de bases de données.

Delphi 5 entreprise, est un logiciel de développement rapide (RDA :Rapide Application Développement)conçu par Borland pour développer plus rapidement et plus facilement des applications sous Windows.

➢ Système de gestion de base de données

Vu l'importance du nombre des contrats et sociétaires qui sont inscrit dans la plus part des entreprises, nous avons choisi un SGBD qui supporte le plus grand nombre d'enregistrements et qui offre la possibilité d'augmenter la taille de la base de données au besoin, chose qui est assurée avec INTERBASE.

INTERBASE est un langage de définition et d'administration de données en même temps interactif et bien adapté à la programmation avec des données. C'est un SGBD relationnelles complet.

7. Test

L'objectif du test d'un logiciel est de détecter les erreurs. La mise au point a pour but de localiser ces erreurs et de les corriger.

Le test permet de réaliser des contrôles pour la qualité du système. Il s'agit de relever les éventuels défauts de conception et de programmation (revue de code, tests des composants,...) [PINET 02].

8. Validation

Dans cette section des diagrammes de cas d'utilisation sont présentés par des copies d'écrans, pour illustrer l'activité de validation et pour être sur que les fonctionnalités du système sont réellement réalisées par notre outil.

Nous allons présenter des copies d'écran de quelques diagrammes de cas d'utilisation.

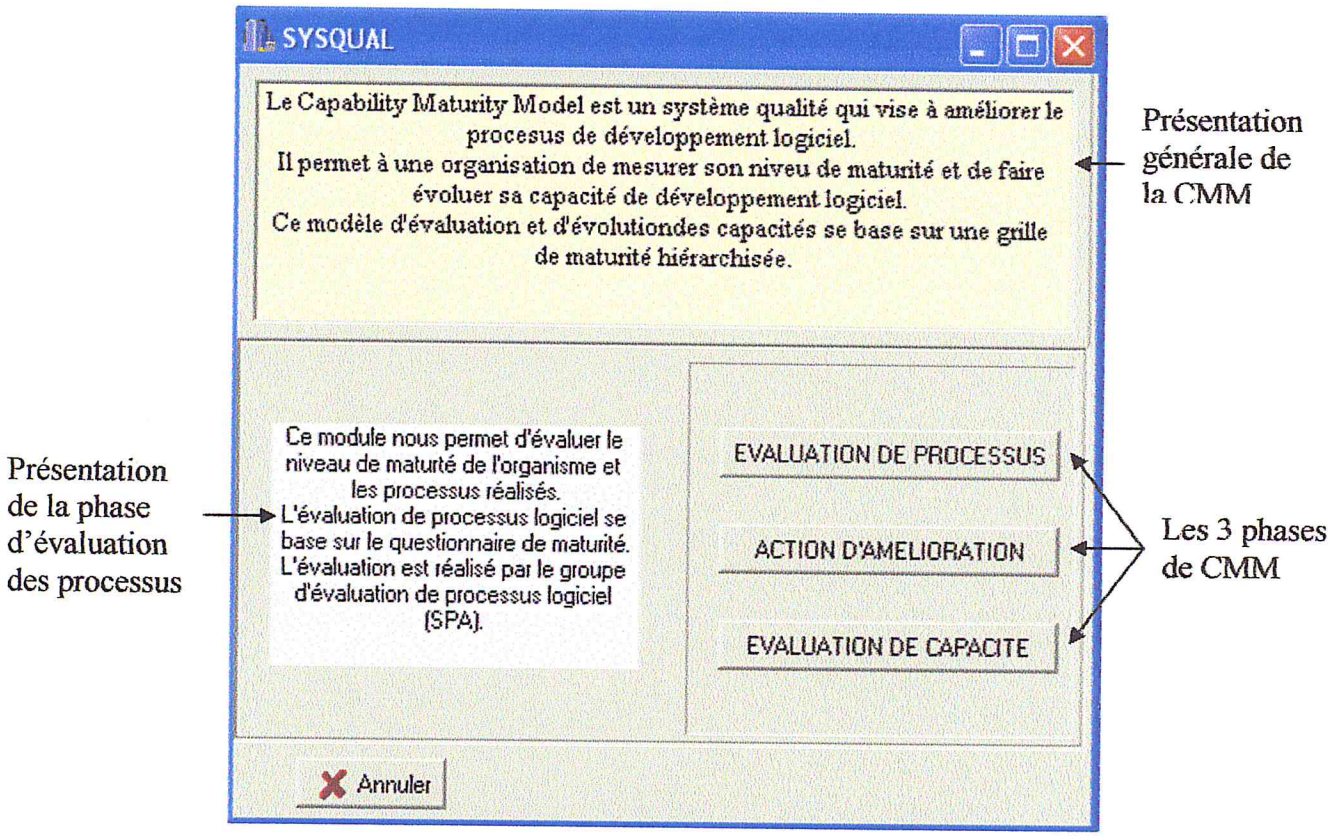


Figure 5-25 : cas d'utilisation global

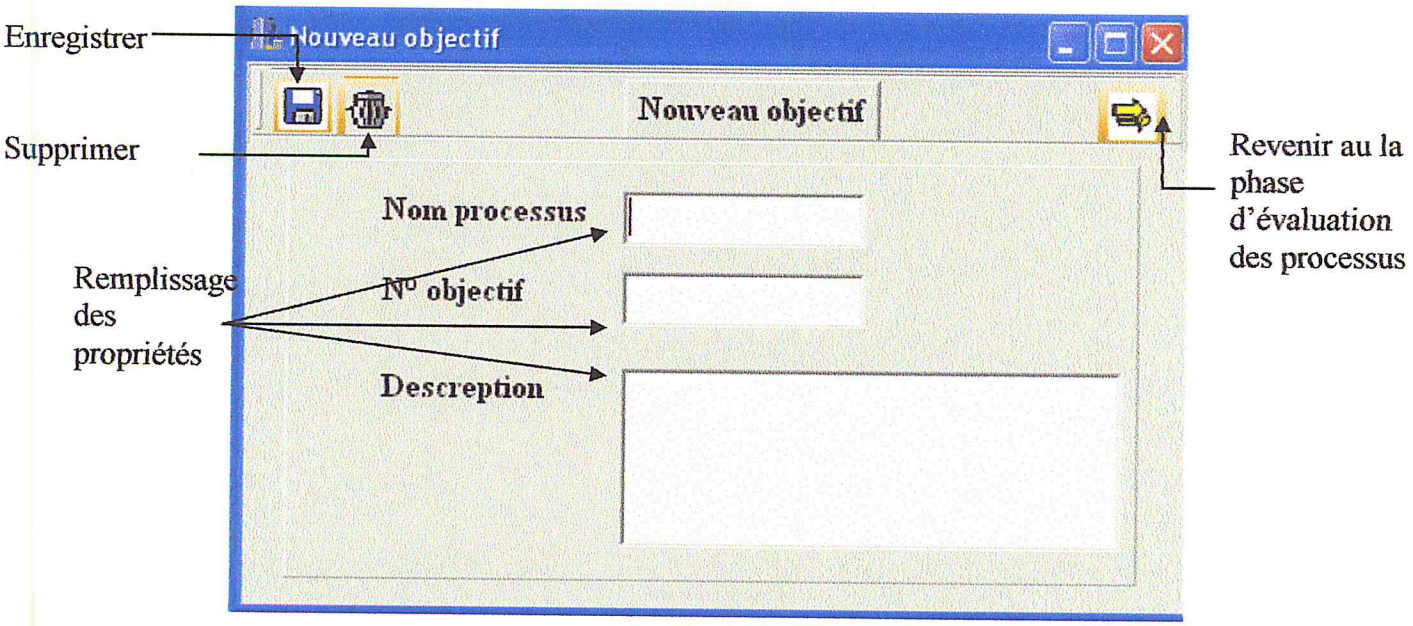


Figure 5-26 : cas d'utilisation " Définition d'un objectif "

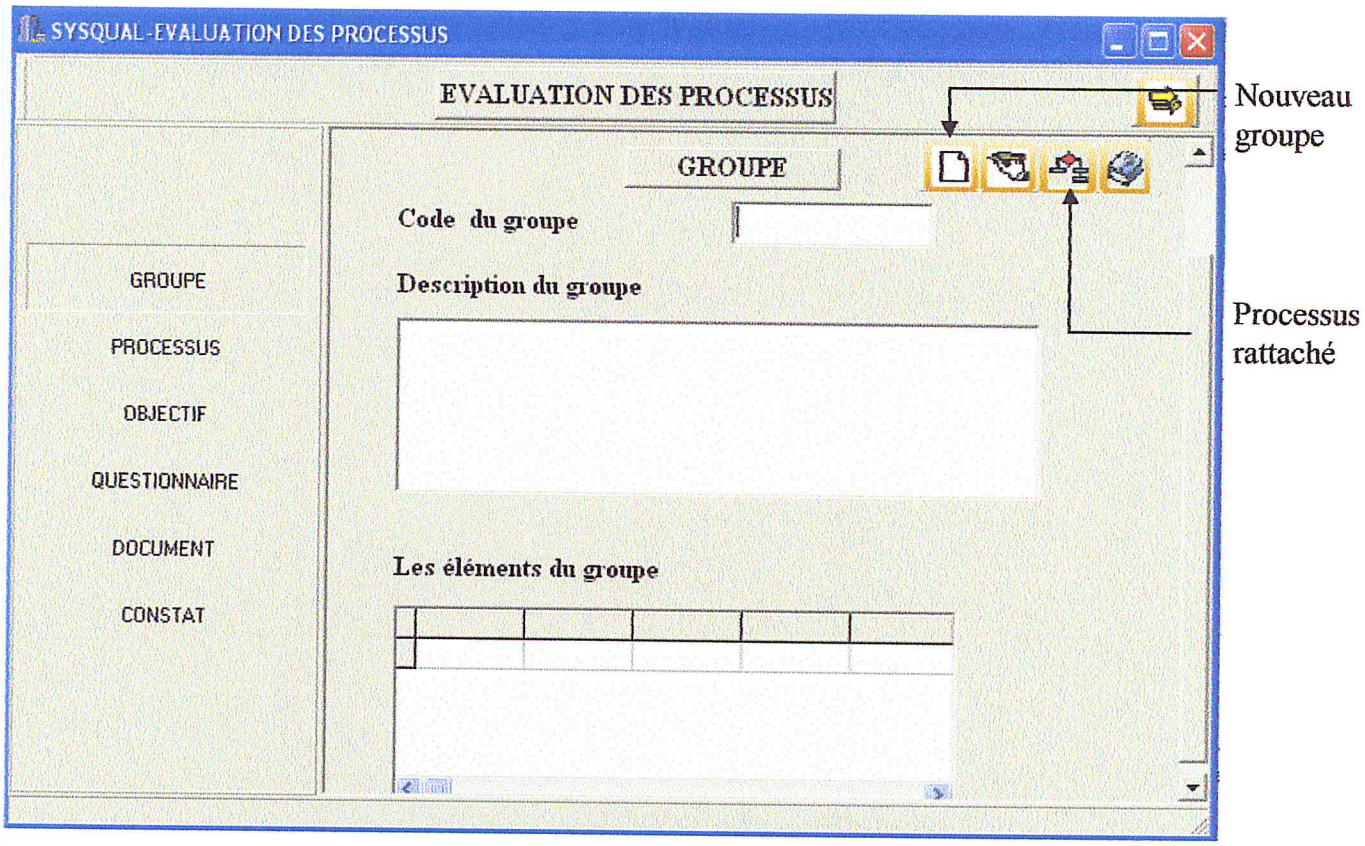


Figure 5-26 : cas d'utilisation " Définition d'un groupe "

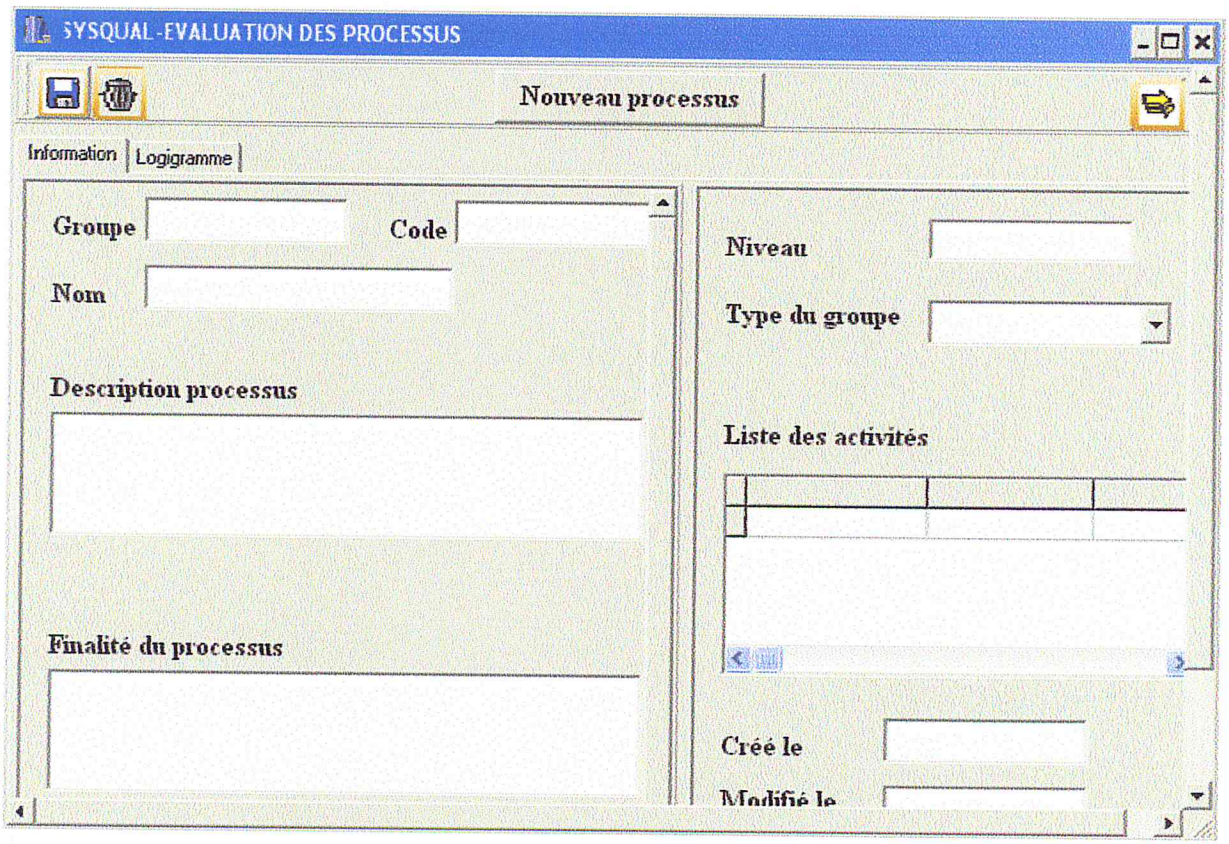


Figure 5-26 : cas d'utilisation " Définition d'un processus "

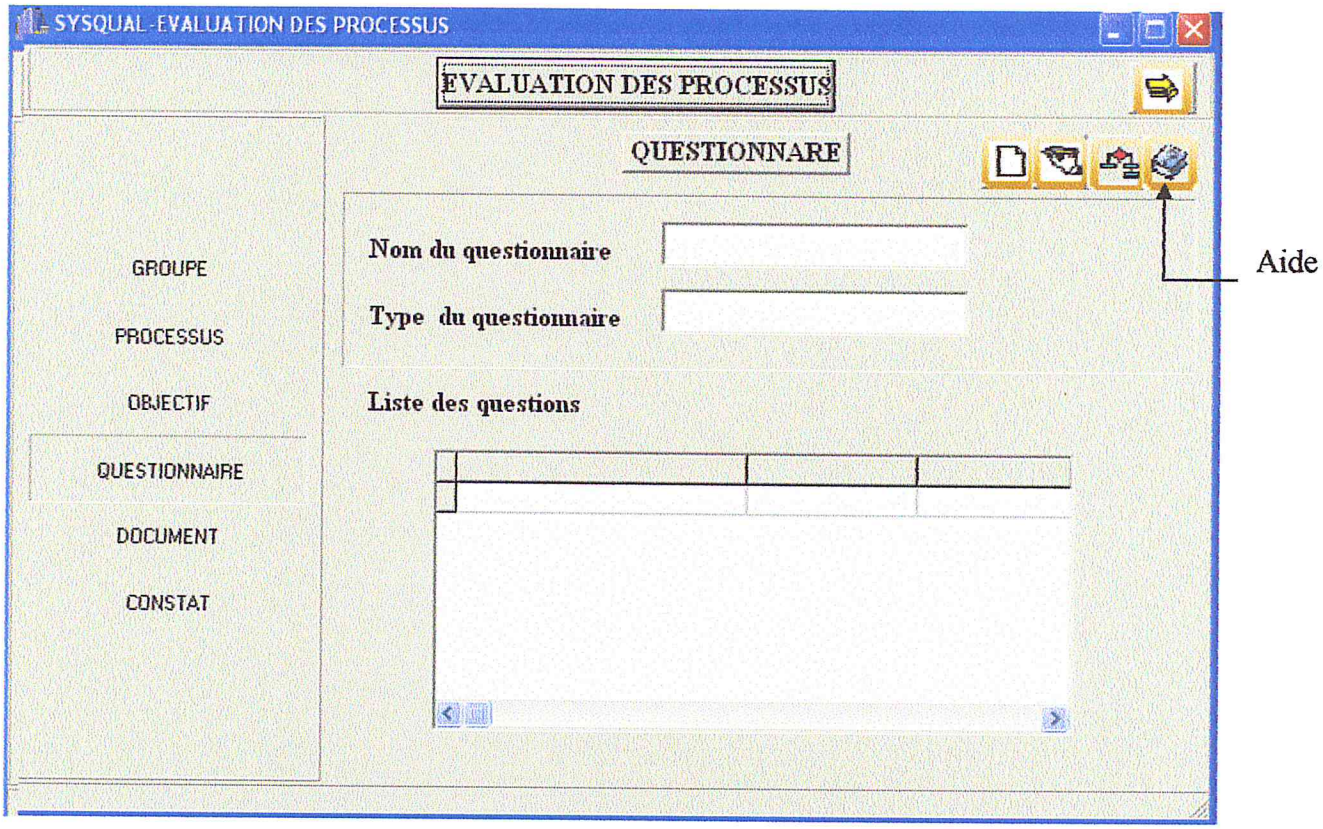


Figure 5-26 : cas d'utilisation " Définition d'un questionnaire"

9. conclusion

Dans ce présent chapitre, nous avons présenté les étapes de développement de notre outil en utilisant une démarche dans un environnement orienté objet. Un cycle de vie en V représente la démarche à suivre, et l'utilisation des notions d'UML montre l'environnement.

Le système de gestion de la qualité logiciel que nous avons développé est un outil, il aide le manager à poursuivre l'amélioration des processus de son organisme afin d'atteindre un niveau de maturité et de faire évoluer sa capacité de développement logiciel.

Conclusion générale

Conclusion générale

L'objectif principal de ce mémoire consiste à concevoir un système de gestion de la qualité du logiciel. Ce système permet aux managers de mieux organiser et maîtriser les processus logiciels.

Au cours de notre projet, nous avons développé un outil dédié entièrement au processus de développement logiciel. Notre outil ne permet pas de tester un produit fini mais c'est un moyen de concevoir un logiciel de bonne qualité.

Notre système permet aux développeurs de faire un bon suivi de projet de développement du logiciel. Il offre ainsi une bonne gestion de documentation du processus logiciel.

Il est basé sur le modèle d'évaluation de processus, ce qu'il permet aux managers de planifier les processus, les mettre en œuvre et évaluer leurs capacités.

Le système ainsi développé, offre donc de nombreux avantages aux managers et aux organisations, notamment la possibilité de structurer leurs processus logiciels sans imposer les outils et méthodes pour le réaliser.

Au cours de notre projet, nous avons acquis de bonnes connaissances sur la qualité ainsi la nécessité de mettre en œuvre un système de management de qualité.

Ce travail a révélé un certain nombre de perspectives qui sont :

- La première concerne la mise en œuvre de notre système dans un organisme qui développe des logiciels.
- La deuxième concerne le développement de notre outil sous un environnement Intranet, ce développement facilite la communication entre le manager et les chefs de groupes.
- Une autre perspective consiste à une réelle mise en œuvre des processus clés.

bibliographie

Bibliographie

[BASQUE 00] Richard Basque, « Amélioration de processus avec le modèle CMM », 2000 disponible au site :

www.alcyonix.com

[BERNADI 02] F.Bernadi, « méthodes d'analyse orienté objet UML », 2002, disponible sur :
//spe.univ-corse.fr bernardiweb/cours.html.

[BOURNICHE 01] François Bourniche, « le modèle CMM du SEI et les actions de normalisation », 2001

[DIRE 01] DIRE , »Fascicule optimiser les processus », disponible sur :
www.fonction_publique.gouv.fr

[FLEURQUIN 96] Régis Fleurquin, « Proposition d'une démarche qualité logicielle pour les PME. Un modèle d'évaluation de la qualité et des critères et conseils permettant sa mise en oeuvre à travers les outils et les méthodes organisation», Institut I.N.S.A, Toulouse,
soutenue le 11-12-1996 disponible au site :
http://www.lesia.insa-tlse.fr/zz-fichiers-associes/documents/these-regis-fleurquin.pdf

[JEHANNO 02] Nicolas JEHANNO, « Les modèles de maturité »CNAM – B5, 2002 disponible au site :
http://perso.club-internet.fr/jehanno/modelematurite.zip

[LARBORDE 02] Carinne Laborde, « capacity maturity model »,2002 ,disponible au site :

www.essi.fr

[LAROUSSE 01] Larousse, «Dictionnaire de français», Manchecourt ,2001

[LEMOINE 02] M.Lemoine « Sensibilisation à l'Assurance qualité », Toulouse ,2001

[MULLER 01] Pierre-Alain Muller, « Modélisation objet avec UML », Eyrolles, 2001

[PINET 02] Claude Pinet, «Processus d'ingénierie du logiciel», Pearson Education France, 2002

[Raisson 99] Céline Raisson , Normes qualité et génie logicie Norme ISO 9000 ,1999-2000.

[SUNIER 95] P.A. Sunier «Analyse des systèmes, Qualité»,1995

[WEBER 93] Charles v.weber « Pratique du modèle d'évolution des capacités logiciels »,1993.

Annexes A

ANNEXE A

Historique de la qualité

La première approche organisée de la gestion de la qualité est due à l'Américain Walter Shewhart, qui mit au point dès 1924, à la Western Electric Company, une méthode pour assurer la qualité des produits

A cette époque, la production de postes téléphoniques d'abonnés était dans une phase de croissance rapide, et la direction de l'entreprise cherchait à prévenir des difficultés qui apparaissent souvent en pareil cas.

Le Dr. Reginald Jones, qui était le directeur du département d'assurance-qualité reçut la mission d'optimiser la qualité finale de la production, c'est-à-dire d'atteindre la satisfaction des clients au moindre coût. Il constitua une équipe avec divers spécialistes : statisticiens, mécaniciens, chimistes, etc. La description des fonctions du service, qu'il rédigea lui-même, ressemble beaucoup à celle que l'on put trouver aujourd'hui dans les entreprises les mieux organisées.

La plupart de ceux qui ont fait partie de l'équipe initiale de Jones ont joué par la suite un rôle important dans le développement de la gestion de la qualité. Ce sont notamment H.F.Dodge, H.G. Romig, G.D.Edwards, J.M. Juran, W.A. Shewhart. Il est très intéressant de remarquer que des notions qui paraissent aujourd'hui très modernes, au point qu'elles ne sont pas encore assimilées par toutes les entreprises industrielles, ont été clairement perçues par Jones et ses collaborateurs. Par exemple, ils ont recherché la fiabilité au stade de la conception, et entrepris l'exploitation complète des résultats d'inspection. Quelques mois après sa création, le service de Jones prit d'ailleurs le nom de << Quality Assurance Department >>.

Shewhart fur parmi les premiers à comprendre l'intérêt des méthodes statistiques dans les systèmes industriels. Une note de Shewhart datée du 16 mai 1924, fait aujourd'hui figure de document historique : elle donne le principe de la carte de contrôle qui est utilisée actuellement. Un peu plus tard, Dodge et Romig définirent des méthodes statistiques d'échantillonnage sur lots, et publièrent des tables qui sont toujours d'usage courant dans l'industrie.

En 1931, Shewhart publia un livre : *Economic Control of Quality manufactured product,* qui resta longtemps la principale référence dans sa spécialité. Ce livre donne les objectifs de la gestion de la qualité, ainsi que l'outil statistique qui permet de les atteindre.
Formé à la même école, J.M. Juran est devenu le plus célèbre, propagandiste de la gestion de la qualité. Son << Quality control handbook >> a été tiré a plus de 40 000 exemplaires, et vendu en plusieurs langues dans le monde entier. Juran a animé des stages de gestion de la qualité dans de nombreux pays, notamment au Japon où son influence a été considérable.
Malheureusement pour l'industrie, les principes énoncés par Jones et Shewhart, qui nous paraissent aujourd'hui essentiels, ont connu une diffusion très lente jusqu'à la Seconde Guerre mondiale.

C'est la guerre elle-même qui a favorisé le développement de la gestion de la qualité. En effet, les usines d'armement aux Etats-Unis se trouvèrent dans l'obligation d'augmenter considérablement leur production en peu de temps, en gardant l'objectif de livrer avec certitude un matériel en bon état de marche. Dans ces conditions exceptionnelles, le gouvernement des Etats-Unis sut mettre à profit l'expérience des quelques spécialistes qui se trouvaient dans l'industrie, dont ceux de Western Electric. De 1941 à 1945, des milliers d'ingénieurs et de techniciens reçurent une formation accélérée en contrôle statistique de la qualité. Des cours bien adaptés aux besoins industriels furent organisés dans les universités, avec ces spécialistes comme professeurs. Après la guerre, un esprit était créé et les << quality engineers >> fondèrent une association très vivante, dans la meilleure tradition des Etats-Unis. Elle rassemblait, en 1979, 30 000 membres.

C'est vers 1960 que le << quality control >> a pris sa véritable dimension aux Etats-Unis, en s'intégrant au système de gestion industrielle. Des programmes d'amélioration de la qualité ont alors puissamment contribué à augmenter le rentabilité des entreprises en réduisant les coûts inutiles par des actions préventives. Le << concept zéro défaut >> de Crosby mettait en évidence le rôle primordial de la prévention des défauts dans la gestion de la qualité.

Le Japon qui fait aujourd'hui figure de leader dans la gestion de la qualité, a dû vaincre la très mauvaise image de marque qui s'attachait à sa production d'avant la guerre. Au sortir de celle-ci les Japonais comprirent vite que leur survie dépendait de l'exportation et que la compétition internationale la qualité constituerait pour eux un atout majeur.

Un groupe de travail fut constitué par la J.U.S.E (Association des ingénieurs et scientifiques japonais) afin d'étudier les méthodes de contrôle statistique de la qualité. Elle organisa des conférences et invita des experts américains : en 1950 le docteur Deming, et en 1954 le docteur Juran. D'après le témoignage de ces deux experts, les directeurs japonais qui étaient venus assister à ces conférences furent très impressionnés et suivirent leurs conseils.
Jusqu'en 1960, les efforts de promotion de la gestion de la qualité dirigés par la J.U.S.E., furent orientés exclusivement vers les ingénieurs et les techniciens.

Un peu plus tard, une brochure à l'intention des agents de maîtrise et des ouvriers fut diffusée largement; puis un magazine mensuel fut édité dans le même esprit. Il est diffusé à 100 000 exemplaires. Présenté sous une forme attrayante, il comporte des exposés pédagogiques, l'étude de cas réels, et des informations professionnelles.
Considérant que son rôle était d'inciter les agents de maîtrise et les ouvriers à étudier les concepts et les techniques de la gestion de la qualité, le comité de rédaction du journal suggéra à ses lecteurs de constituer des groupes pour une lecture en commun. Ces groupes, nommés cercles de contrôle de la qualité, se multiplièrent et s'organisèrent au niveau national.

Toutes les spécialités et tous les niveaux hiérarchiques y sont représentés. Il y a maintenant au Japon 300 000 cercles enregistrés. Leur activité est grande, car chaque cercle se réunit en moyenne 2 fois par mois.
Ce mouvement, qui bénéficie du support permanent des chefs d'entreprise, permet d'expliquer en grande partie le succès mondial que l'industrie japonaise connaît maintenant dans de nombreux domaines.

Annexes B

ANNEXE B

Présentation du secteur clé « Gestion des exigences »

Un secteur clé au Niveau 2 : reproductible

La Gestion des exigences vise à établir une compréhension commune,entre le client et le projet logiciel, des exigences du client que le projet logiciel se propose de satisfaire.

La Gestion des exigences comprend la mise en place et la maintenance d'un accord avec le client quant aux exigences pour le projet logiciel. Cet accord est appelé «exigences système allouées au logiciel». Le terme «client» peut désigner le groupe d'ingénierie système, le groupe de mercatique, une autre organisation interne ou un client externe. L'accord couvre à la fois les exigences techniques et non techniques (dates de livraison, par exemple). L'accord constitue la base de l'estimation, de la planification, de la mise en oeuvre et du suivi des activités du projet logiciel tout au long du cycle de vie logiciel.

L'allocation des exigences système aux composants logiciel, matériels et autres (personnes, par exemple) peut être réalisée par un groupe extérieur au groupe d'ingénierie logiciel (groupe d'ingénierie système, par exemple), et le groupe d'ingénierie logiciel n'a parfois aucun contrôle direct sur cet aspect. Le groupe d'ingénierie logiciel doit, compte tenu des contraintes inhérentes au projet, prendre les mesures appropriées pour s'assurer que les exigences système allouées au logiciel, qu'il a la responsabilité de traiter, soient documentées et contrôlées.

Pour réaliser ce contrôle, le groupe d'ingénierie logiciel passe en revue les exigences système initiales et révisées allouées au logiciel afin de clarifier toutes les questions avant leur prise en charge par le projet logiciel. Lorsque les exigences système allouées au logiciel sont modifiées, les plans, produits de travail et activités logiciel affectés sont adaptés de façon à rester cohérents avec les exigences modifiées.

1.Objectifs

Objectif 1 : Les exigences système allouées au logiciel sont contrôlées de façon à établir un référentiel pour les activités d'ingénierie logiciel et de gestion.

Objectif 2 : Les plans, produits et activités logiciel sont maintenus cohérents avec les exigences système allouées au logiciel.

2.Engagement de réalisation

Engagement 1 : Le projet suit une directive écrite de l'organisation pour la gestion des exigences système allouées au logiciel.

- ❖ Les exigences système allouées au logiciel sont appelées «exigences allouées» dans ces pratiques.
- ❖ Les exigences allouées sont le sous-ensemble des exigences système qui doit être pris en charge par les composants logiciel du système.
- ❖ Les exigences allouées constituent une des principales entrées du plan de développement logiciel. L'analyse des exigences logiciel définit et raffine les exigences allouées et donne comme résultat des exigences logiciel documentées.

Typiquement, cette directive stipule ce qui suit :

1. Les exigences allouées sont documentées.
2. Les exigences allouées sont passées en revue par :
   - les responsables logiciel, et
   - les autres groupes concernés.
     - ✓ Les groupes concernés comprennent, par exemple :
       - les tests système,
       - l'ingénierie logiciel (y compris tous les sous-groupes, par exemple, la conception logiciel),
       - l'ingénierie système,
       - l'assurance-qualité logiciel,
       - la gestion des configurations logiciel, et

- le soutien à la documentation.

3. Les plans, activités et produits de travail logiciel sont modifiés pour rester cohérents avec les changements apportés aux exigences allouées.

3.Capacité de réalisation

Capacité 1 : La responsabilité d'analyse des exigences système et de leur allocation aux composants logiciel, matériels et autres composants du système est assignée pour chaque projet.

- ✓ L'analyse et l'allocation des exigences système ne relèvent pas du groupe d'ingénierie logiciel mais constituent un préalable à son travail.

Cette responsabilité couvre :

1. La gestion et la documentation des exigences système et leur allocation tout au long du projet.
2. La mise en oeuvre des changements apportés à ces exigences et à leur allocation.

Capacité 2 : Les exigences allouées sont documentées.

- ✓ Les exigences allouées comprennent :

1.Les exigences non techniques (c'est-à-dire les accords, les conditions et/ou les clauses contractuelles) affectant et déterminant les activités du projet logiciel.

- Les accords, conditions et clauses contractuelles comprennent, par exemple :
  - les produits à livrer,
  - les dates de livraison, et
  - les jalons.

2. Les exigences techniques quant au logiciel.

- ✓ Les exigences techniques comprennent, par exemple :
  - les fonctions utilisateur final, opérateur, soutien ou intégration;
  - les exigences de performance;
  - les contraintes de conception;
  - le langage de programmation; et
  - les exigences d'interface.

3. Les critères d'acceptation à utiliser pour vérifier si les produits logiciel satisfont aux exigences allouées.

Capacité 3 : Des ressources et un financement suffisants sont fournis pour la gestion des exigences allouées.

1. Des personnes possédant l'expérience et les compétences voulues dans le domaine d'application et en ingénierie logiciel sont affectées à la gestion des exigences allouées.
2. Des outils de soutien aux activités de gestion des exigences sont rendus disponibles.

Les outils de soutien comprennent, par exemple :

- les tableurs,
- les outils de gestion des configurations,
- les outils de traçabilité, et
- les outils de gestion des tests.

Capacité 4 : Les membres du groupe d'ingénierie logiciel et des autres groupes de support logiciel sont formés pour réaliser leurs activités de gestion des exigences.

Cette formation peut couvrir, par exemple :

- les méthodes, les normes et les procédures utilisées dans le cadre du projet, et
- le domaine d'application.

4. Activités réalisées

Activité 1 : Le groupe d'ingénierie logiciel passe en revue les exigences allouées avant leur prise en charge par le projet logiciel.

1. Les exigences allouées incomplètes ou manquantes sont repérées.
2. Les exigences allouées sont passées en revue pour déterminer :
   - si elles sont réalisables et appropriées à une solution logiciel,
   - si elles sont énoncées de façon claire et correcte,
   - si elles sont cohérentes les unes avec les autres, et
   - si elles sont testables.

3. Les exigences allouées pour lesquelles un problème potentiel a été identifié sont passées en revue avec le groupe chargé de l'analyse l'allocation des exigences système et les changements nécessaires effectués.

4. Les engagements résultant des exigences allouées sont négociés les groupes concernés :

✓ Les groupes concernés comprennent, par exemple :

- l'ingénierie logiciel (y compris tous les sous-groupes, dont celui de la conception Logiciel),
- l'estimation du projet logiciel,
- l'ingénierie système,
- les tests système,
- l'assurance-qualité logiciel,
- la gestion des configurations logiciel,
- la gestion des contrats, et
- le soutien à la documentation.

Activité 2 : Le groupe d'ingénierie logiciel utilise les exigences allouées comme base pour les plans, produits de travail et activités logiciel.

✓ Les exigences allouées :

1. Sont gérées et contrôlées.

L'expression «géré et contrôlé» implique que la version du produit utilisé à un moment donné (passé ou présent) est connue (contrôle de version) et que les modifications y sont incorporées de façon contrôlée (contrôle des changements).

Lorsqu'on souhaite un formalisme plus poussé que celui impliqué par «géré et contrôlé», le produit de travail logiciel peut être soumis à la discipline intégrale de gestion des configurations, telle que décrite dans le secteur clé Gestion de configuration logiciel.

2. Constituent la base du plan de développement logiciel.

3. Constituent la base du développement des exigences logiciel.

Activité 3 : Les changements aux exigences allouées sont passés en revue et pris en charge par le projet logiciel.

1. L'impact sur les engagements actuels est évalué et les changements sont négociés, au besoin.
   - Les changements aux engagements pris envers les personnes et les groupes externes à l'organisation sont passés en revue avec la Direction.
   - Les changements aux engagements pris au sein de l'organisation sont négociés avec les groupes concernés.

2. Les changements qui doivent être apportés aux plans, produits de travail et activités logiciel par suite de changements apportés aux exigences allouées sont :
   - identifiés,
   - évalués,
   - analysés en termes de risques,
   - documentés,
   - planifiés,
   - communiqués aux personnes et aux groupes concernés,
   - soumis à un suivi jusqu'à l'achèvement des travaux.

5.Mesures et analyse

Mesure 1 : Des mesures sont effectuées et utilisées afin de déterminer l'état des activités de gestion des exigences allouées.

Les mesures couvrent, par exemple :

- l'état de chacune des exigences allouées;
- le changement des exigences allouées; et
- le nombre total cumulatif de changements apportés aux exigences allouées, y compris le nombre total de changements proposés, à l'étude, approuvés et intégrés au référentiel du système.

6.Vérification de mise en oeuvre

Vérification 1 : Les activités de gestion des exigences allouées sont passées en revue avec la Direction sur une base périodique.

L'objet principal des revues périodiques par la Direction est de lui fournir des renseignements et une vue non superficielle sur les activités du processus logiciel au niveau d'abstraction qui convient et en temps voulu. L'intervalle entre les revues devrait être fonction des besoins de l'organisation et peut être relativement long, dès lors que des mécanismes appropriés de rapport des cas d'exception sont disponibles.

Consulter la Vérification 1 du secteur clé Suivi et supervision de projet logiciel pour les pratiques décrivant la teneur habituelle des revues de supervision par la Direction.

Vérification 2 : Les activités de gestion des exigences allouées sont passées en revue avec le chef de projet autant sur une base périodique que sur événement.

Consulter la Vérification 2 du secteur clé Suivi et supervision de projet logiciel pour les pratiques décrivant la teneur habituelle des revues de supervision par le chef de projet.

Vérification 3 : Le groupe d'assurance-qualité logiciel passe en revue les activités et les produits de travail pour la gestion des exigences allouées et/ou effectue un audit de ceux-ci et rend compte des résultats obtenus.

Consulter à ce sujet le secteur clé Assurance-qualité logiciel.

- ✓ Au minimum, ces revues et/ou audits vérifient que :

1. Les exigences allouées sont passées en revue et les problèmes résolus avant que le groupe d'ingénierie logiciel ne s'engage à leur prise en charge.
2. Les plans, produits de travail et activités logiciel sont révisés de manière appropriée lorsque les exigences allouées sont modifiées.
3. Les changements aux engagements découlant de changements aux exigences allouées sont négociés avec les groupes concernés.