



## MÉTHODOLOGIES DE DÉVELOPPEMENT

# CONDUITE DE PROJETS INFORMATIQUES

Auteur: Bernard GIACOMONI

Autoentreprise GIACOMONI Bernard

<b>VERSIONS &amp; DATES</b>	<b>OBJET</b>	<b>AUTEUR(S)</b>
<i>Version 1.0 Décembre 2008</i>	<i>Création</i>	<i>Bernard GIACOMONI - Association A.T.L.A.N.T.I.C</i>
<i>Version 1.1</i>	<i>Prise en compte des méthodes agiles</i>	<i>Bernard GIACOMONI Autoentreprise GIACOMONI Bernard</i>

# SOMMAIRE

## Table des matières

<b>I. INTRODUCTION:</b> .....	<b>5</b>
<b>I.1. OBJET DU DOCUMENT:</b> .....	<b>5</b>
<b>I.2. CARACTÉRISTIQUES GÉNÉRALES DE LA DÉMARCHE:</b> .....	<b>5</b>
<b>II. GÉNÉRALITÉS SUR LA CONDUITE DE PROJETS:</b> .....	<b>6</b>
<b>II.1. QU'EST-CE QU'UN PROJET?</b> .....	<b>6</b>
<b>II.2. QUE SIGNIFIE «CONDUIRE» UN PROJET ?</b> .....	<b>6</b>
<b>II.3. CRITÈRES DE CONTRÔLE D'UN PROJET:</b> .....	<b>6</b>
II.3.1. LA NOTION DE QUALITÉ:.....	6
II.3.2. RESPECT DU PLANNING:.....	7
II.3.3. MAÎTRISE DES COÛTS:.....	7
II.3.4. INTERDÉPENDANCE DE CES TROIS CRITÈRES:.....	7
<b>II.4. LA STRATÉGIE: DIVISER POUR RÉGNER:</b> .....	<b>8</b>
I.1.1. LE CONTRÔLE DES COÛTS ET DES DÉLAIS:.....	9
I.1.2. LE CONTRÔLE DE LA QUALITÉ:.....	9
I.1.3. LA NOTION DE VALEUR:.....	9
<b>III. PRINCIPALES DÉMARCHES DE DÉVELOPPEMENT:</b> .....	<b>10</b>
<b>III.1. PRINCIPAUX TYPES D'ACTIVITÉS LIÉES A UN PROJET:</b> .....	<b>10</b>
III.1.1. INTRODUCTION:.....	10
III.1.2. SPÉCIFIER CE QUE L'ON DOIT FAIRE:.....	10
III.1.3. CONCEVOIR COMMENT ON VA LE FAIRE:.....	10
III.1.4. RÉALISER L'OBJET DU PROJET:.....	10
III.1.5. VÉRIFIER QUE L'OBJET PRODUIT EST BIEN CONFORME A CE QUI EST ATTENDU :.....	11
<b>III.2. MÉTHODES CLASSIQUES ET MÉTHODES AGILES :</b> .....	<b>11</b>
III.2.1. PRINCIPE DIRECTEUR DES MÉTHODES CLASSIQUES :.....	11
III.2.2. PRINCIPE DIRECTEUR DES MÉTHODES AGILES:.....	12
<b>IV. LES DÉMARCHES EN CASCADES - CYCLE EN V:</b> .....	<b>13</b>
<b>IV.1. RAPPELS SUR LE CYCLE EN V:</b> .....	<b>13</b>
IV.1.1. LES PRINCIPALES PHASES D'UN PROJET:.....	13
IV.1.2. CARACTÉRISTIQUES DU CYCLE EN V:.....	14
IV.1.2.1. UN CYCLE EN DEUX TEMPS:.....	14
IV.1.2.2. SYMÉTRIE ENTRE PHASES:.....	14
IV.1.2.3. RÉCURSIVITÉ DE LA DÉMARCHE:.....	14
IV.1.3. INCONVÉNIENTS DU CYCLE EN V:.....	15
IV.1.3.1. RISQUES LIES AU RETARD DE DÉTECTION DES DÉFAUTS DE SPÉCIFICATION ET DE CONCEPTION PRÉLIMINAIRE:.....	15
IV.1.3.2. RISQUES LIES A LA NON IMPLICATION DU CLIENT LORS DES PHASES DE RÉALISATION:.....	15
IV.1.4. AVANTAGES DU CYCLE EN V:.....	16
IV.1.4.1. ADAPTABLE A TOUS LES TYPES ET TOUTES LES TAILLE DE PROJET:.....	16
IV.1.4.2. UNE GESTION RIGOREUSE DES ASPECTS CONTRACTUELS:.....	16
IV.1.4.3. UNE PARTICIPATION DU CLIENT RÉDUITE AU STRICT NÉCESSAIRE:.....	16
IV.1.5. LE CYCLE EN V AVEC DÉVELOPPEMENT INCRÉMENTAL:.....	16
<b>IV.2. DESCRIPTION DE LA DÉMARCHE GÉNÉRALE:</b> .....	<b>18</b>
IV.2.2. DIFFÉRENTS TYPES D'ACTIVITÉS:.....	18
IV.2.3. LES TRAVAUX PRÉALABLES:.....	19
IV.2.3.1. GÉNÉRALITÉS:.....	19
IV.2.3.2. LES ÉTAPES DES TRAVAUX PRÉALABLES:.....	19
IV.2.3.3. RECUEIL ET ANALYSE DU BESOIN:.....	20

IV.2.3.4. SPÉCIFICATIONS TECHNIQUES:.....	20
IV.2.3.5. CONTRACTUALISATION:.....	21
IV.2.3.6. ORGANISATION ET PLANIFICATION DU PROJET:.....	22
IV.2.4. TRAVAUX DU PROCESSUS DE DÉVELOPPEMENT:.....	23
IV.2.4.1. ÉTAPES DES TRAVAUX DU PROCESSUS DE DÉVELOPPEMENT:.....	23
IV.2.4.2. CONCEPTION PRÉLIMINAIRE:.....	24
IV.2.4.3. CONCEPTION DÉTAILLÉE-ORGANISATION DE LA RÉALISATION:.....	26
IV.2.4.4. CONCEPTION DÉTAILLÉE-RÉALISATION:.....	26
IV.2.4.5. TESTS UNITAIRES:.....	26
IV.2.4.6. INTÉGRATION:.....	27
IV.2.5. TRAVAUX COMPLÉMENTAIRES:.....	28
IV.2.5.1. ÉTAPES DES TRAVAUX COMPLÉMENTAIRES:.....	28
IV.2.5.2. PRÉPARATION DE LA RÉCEPTION DURANT LA PHASE DE SPÉCIFICATION TECHNIQUE:.....	28
IV.2.5.3. PRÉPARATION DE LA RÉCEPTION DURANT LES PHASES DE DÉVELOPPEMENT:.....	29
IV.2.5.4. VALIDATION:.....	29
IV.2.5.5. QUALIFICATION OPÉRATIONNELLE:.....	30
<b>V. LES DÉMARCHES ITÉRATIVES - MÉTHODES AGILES:.....</b>	<b>31</b>
<b>V.1. HISTORIQUE:.....</b>	<b>31</b>
<b>V.2. PRINCIPES FONDAMENTAUX DES MÉTHODES AGILES:.....</b>	<b>31</b>
V.2.1. INTRODUCTION:.....	31
V.2.2. PREMIER PRINCIPE: DÉVELOPPEMENT ITÉRATIF:.....	31
V.2.2.1. DESCRIPTION:.....	31
V.2.2.2. AVANTAGES:.....	32
V.2.2.3. INCONVÉNIENTS:.....	33
V.2.3. DEUXIÈME PRINCIPE: LIVRER AU PLUS TÔT CE QUI PEUT L'ÊTRE:.....	33
V.2.3.1. PRÉSENTATION:.....	33
V.2.3.2. AVANTAGES:.....	33
V.2.3.3. INCONVÉNIENTS:.....	33
V.2.4. AUTRES PRINCIPES LIÉS AUX MÉTHODES AGILES:.....	33
<b>V.3. MÉTHODES AGILES- DÉROULEMENT GÉNÉRAL:.....</b>	<b>35</b>
V.3.1. INITIALISATION DU PROJET ET PREMIÈRE ITÉRATION:.....	35
V.3.2. ITÉRATIONS SUIVANTES:.....	35
V.3.3. RECUEIL DES EXIGENCES DU CLIENT:.....	35
V.3.4. CONSTITUTION DES CAHIERS DE RÉCEPTION:.....	36
V.3.5. NÉGOCIATION DES EXIGENCES:.....	36
V.3.6. DÉMARCHE DE PLANIFICATION:.....	36
V.3.6.1. POSITION DU PROBLÈME:.....	36
V.3.6.2. ÉTABLISSEMENT D'UN PLAN DE DÉVELOPPEMENT:.....	36
<b>VI. SYNTHÈSE SUR LES DEUX TYPES DÉMARCHES:.....</b>	<b>38</b>
<b>VII. ANNEXES:.....</b>	<b>39</b>
<b>VII.1. CONTENUS ET PLANS DES DOCUMENTS (CAS D'UN CYCLE EN V):.....</b>	<b>39</b>
VII.1.1. GÉNÉRALITÉS:.....	39
VII.1.2. CAHIER DES CHARGES FONCTIONNEL (CDCF):.....	40
VII.1.2.1. But du Document:.....	40
VII.1.2.2. Plan et contenu:.....	40
VII.1.3. SPÉCIFICATION TECHNIQUE DE BESOIN ET DOSSIER JUSTIFICATIF DES EXIGENCES: .....	42
VII.1.3.1. But du document:.....	42
VII.1.3.2. Plan et contenu du document:.....	42
VII.1.4. MANUEL OPÉRATEUR:.....	44
VII.1.4.1. But du document:.....	44
VII.1.4.2. Contenu:.....	44
VII.1.5. ORGANIGRAMME DES TACHES (OT):.....	46
VII.1.5.1. But du document:.....	46
.....	46
VII.1.5.2. Démarche de construction:.....	46
VII.1.6. PLAN DE DÉVELOPPEMENT (PDv):.....	48
VII.1.7. DOCUMENTS DE CONCEPTION PRÉLIMINAIRE:.....	50

VII.1.7.1. But du document:.....	50
VII.1.7.2. Plan et contenu:.....	50
VII.1.8. PLAN D'INTÉGRATION:.....	51
VII.1.8.1. But du document:.....	51
VII.1.8.2. Plan et contenu:.....	51
VII.1.9. CAHIER DE RÉCEPTION GÉNÉRAL:.....	53
VII.1.9.1. But du document:.....	53
VII.1.9.2. Contenu:.....	53
VII.1.10. DOSSIERS DE DÉFINITION D'UN COMPOSANT:.....	54
VII.1.10.1. But du document:.....	54
VII.1.10.2. Contenu.....	54
VII.1.11. DOSSIER DE TEST UNITAIRE D'UN COMPOSANT:.....	55
VII.1.11.1. But du document:.....	55
VII.1.11.2. Contenu:.....	55
VII.1.12. COMPTES-RENDUS DE TESTS UNITAIRES:.....	55
VII.1.13. CAHIERS DE RÉCEPTION DÉTAILLÉS:.....	56
VII.1.13.1. But du document:.....	56
VII.1.13.2. Contenu:.....	56
VII.1.13.3. COMPTES-RENDUS DE TESTS DE RÉCEPTION:.....	56
<b>VII.2. PRINCIPES DE DÉCOMPOSITION ORGANIQUE:.....</b>	<b>57</b>

## I.INTRODUCTION:

### I.1.OBJET DU DOCUMENT:

Le présent document constitue un guide méthodologique pour la conduite de projets dont l'objet est le développement d'un **systèmes informatique**.

Ce guide s'adresse à des lecteurs connaissant les **concepts de base** de la conduite de projets et du contrôle qualité. Il s'attache à définir les objectifs et le contenu de chaque phase du projet et de décrire le mécanisme général d'enchaînement de ces phases, indépendamment de toute méthodologie particulière.

Il peut s'appliquer à la fois à des projets dont tout ou partie de la réalisation est sous-traitée et à des projets entièrement menés en interne.

Un système informatique peut être centralisé sur une seule machine ou réparti sur plusieurs postes en réseau. Il peut concerner à la fois des composants logiciels et des composants matériels. La démarche exposée ici essaie d'englober ces différents possibilités.

### I.2.CARACTÉRISTIQUES GÉNÉRALES DE LA DÉMARCHE:

- La démarche est globalement basée sur le schéma du «cycle en V», dont les principes sont brièvement rappelés dans le chapitre suivant: cette particularité permet de profiter de l'adaptabilité de ce cycle à toutes les tailles et tous les types de projets. Cependant, afin de corriger certains défauts de ce type de démarche, des itérations ont été ajoutées dans chaque phase et sous-phase.
- Dans chaque phase, les activités ainsi que les fournitures consommées et produites sont décrites d'une manière le plus indépendante possible du domaine de réalisation.
- La démarche n'imposant l'utilisation d'aucun outil ou méthode spécifique, le réalisateur garde la liberté de leur choix, en fonction des caractéristiques de la réalisation (nature, taille, domaine technique, etc...), et de la formation des développeurs.
- La documentation associée au projet est considérée comme partie intégrante de l'activité de développement, au même titre que le code objet ou les composants matériels réalisés. En effet, les données d'entrée de chaque phase sont le plus souvent constituées par des documents produits dans les phases précédentes: la documentation précède donc la réalisation et lui sert de base. Ceci permet de s'assurer qu'elle est effectivement produite et qu'elle correspond bien à ce qui a été réalisé.

## II.GÉNÉRALITÉS SUR LA CONDUITE DE PROJETS:

### II.1.QU'EST-CE QU'UN PROJET?

Dans le vocabulaire courant, la notion de projet se rapporte plus à l'**intention** d'accomplir quelque chose qu'à l'**accomplissement** lui-même. Cependant, dire «j'ai le projet d'aller en Chine l'an prochain» implique plus qu'une simple intention. On comprend qu'un certain nombre d'activités préliminaires (prise de renseignements, par exemple) ont été effectuées ou sont en cours (peut-être même un début de planification). En revanche, on peut noter que si la personne a déjà réservé ses billets d'avion, on ne parlera plus vraiment de projet, car il y a un début de réalisation de l'**objet** du projet.

Dans le langage professionnel, le sens a évolué pour englober toutes les phases de la réalisation de l'objet (cet objet étant un **produit** ou un **service**), depuis la définition du besoin jusqu'à sa livraison. Le mot PROJET désigne donc une entreprise humaine individuelle ou collective de durée limitée dans le temps dont le but unique est de créer un objet spécifique (produit ou service). Un projet possède une date de début et une date de fin (correspondant à l'**acceptation** de l'objet par le client), qui est souvent impérative. Enfin, à un projet donné correspond une **organisation** qui lui est propre: des moyens humains, matériels, financiers lui sont alloués pour la durée du projet (des collaborateurs peuvent être engagés pour tout ou partie de sa durée).

### II.2.QUE SIGNIFIE «CONDUIRE» UN PROJET ?

De même que conduire un véhicule consiste en premier lieu à contrôler sa vitesse et sa direction et éventuellement à modifier ces paramètres en fonction des circonstances, conduire un projet consiste principalement à contrôler son avancement et à prendre en cas de besoin des mesures correctives.

Le conducteur de projet (ou mieux: **directeur de projet**) doit donc disposer d'outils de contrôle lui permettant de vérifier à un instant donné que le projet progresse vers les résultats attendus et à l'allure prévue. Il a aussi besoin de moyens de commande lui permettant d'agir sur l'activité des différents intervenants afin de prendre des mesures correctives.

Les moyens de commande sont d'une part l'autorité qui lui est conférée en tant que **directeur de projet** sur les différents intervenants de ce projet, et d'autre part le pouvoir de disposer des ressources matérielles allouées en fonction des besoins. Il n'y a là rien de spécifique par rapport à n'importe quelle responsabilité de commandement.

Pour contrôler un projet, il faut disposer de critères permettant de caractériser son bon déroulement. On peut globalement classer ces critères en trois catégories:

1. La **QUALITÉ** du résultat.
2. Le respect du **PLANNING**.
3. La maîtrise des **COÛTS**.

### II.3.CRITÈRES DE CONTRÔLE D'UN PROJET:

#### II.3.1.LA NOTION DE QUALITÉ:

Dans la vie courante, la qualité d'un produit est le plus souvent considérée comme une notion absolue: par exemple, une chaussure sera reconnue de bonne ou de mauvaise qualité en fonction des caractéristiques des matières premières employées, de sa finition, de sa coupe, etc... En général, un produit de qualité supérieure sera plus coûteux qu'un produit «bas de gamme». Cependant, on peut remarquer que cette manière (courante) de voir les choses ne préjuge en rien de l'efficacité du produit: une chaussure jugée de qualité supérieure peut très bien s'avérer beaucoup moins adaptée à la boue et aux grands froids que de simples bottes de caoutchoux.

Dans le domaine de la conduite de projet, la notion de QUALITÉ est toute différente: elle mesure ***l'écart qui existe entre les services que le produit peut rendre à l'utilisateur et ce que cet utilisateur en attend***. (Cette notion est donc assez proche de la notion d'**efficacité**). Dans cette optique, pour un utilisateur qui désire en priorité posséder un véhicule pour se déplacer en ville, une petite voiture de début de gamme (s'il existe un besoin de transporter de petites charges ou des passagers), voire un scooter (dans le cas contraire) peuvent être jugés de meilleure qualité qu'une berline luxueuse: tout dépend de la **définition du besoin** de l'utilisateur.

La **QUALITÉ** d'un produit mesure donc l'écart qui existe entre les **SERVICES** que le produit peut rendre à l'utilisateur et le **BESOIN** défini par cet utilisateur (évidemment, moins l'écart est grand, plus la qualité est bonne).

### II.3.2.RESPECT DU PLANNING:

Garantir la livraison à temps du résultat d'un projet peut être aussi important (et même plus important dans certains cas) que garantir la qualité. Prenons par exemple le cas de la réalisation d'un équipement devant servir aux prochains jeux olympiques: il est évidemment préférable de livrer à temps un équipement inachevé que de le livrer achevé, mais après la cérémonie de clôture!. Dans ce cas, une prise de retard dans la réalisation amènera la direction de projet à sacrifier quelques travaux non essentiels (donc à sacrifier la QUALITÉ). Cette décision permettra d'accélérer le reste de la réalisation en y consacrant tous les moyens disponibles afin de livrer à temps.

### II.3.3.MAÎTRISE DES COÛTS:

Quelle que soit la nature d'un projet, sa réalisation exige la consommation de RESSOURCES:

- Le temps de travail fourni par les réalisateurs.
- Le temps d'utilisation de machines, de véhicules...
- La consommation de produits (carburant, papier, matériaux divers)
- Le concours éventuel de sous-traitants
- La délivrance d'autorisations, le paiement de loyers, de taxes, de royalties, etc...

La plupart de ces ressources sont PAYANTES. Leur consommation influe donc sur le COÛT DE RÉALISATION.

Dans un environnement marchand, il est évidemment primordial que ce coût de réalisation soit inférieur au prix convenu avec le client (sous peine de travailler à perte). Il est donc de première importance de maîtriser les coûts de réalisation. Cette maîtrise passe par le contrôle de l'utilisation des ressources.

### II.3.4.INTERDÉPENDANCE DE CES TROIS CRITÈRES:

En fait, ces trois critères interagissent fortement entre eux:

- La recherche de la qualité consomme des ressources: corriger un écart existant entre les caractéristiques du produit et les caractéristiques demandées demande d'y allouer du temps (temps humain, temps machine), ce qui augmente les coûts et peut retarder l'achèvement.
- Pour respecter la date d'achèvement, on peut être amené soit à sacrifier la qualité du produit en ne réalisant pas certaines fonctions, soit à engager des collaborateurs ou recourir à la sous-traitance, ce qui augmente les coûts.
- Enfin, pour tenir les coûts, on peut être amené à réduire la consommation des ressources (se séparer de collaborateurs ou de sous-traitants, réduire le budget d'équipement, etc...) ce qui peut influencer soit sur la qualité (si on veut tenir les délais), soit sur la date d'achèvement (si on veut conserver la qualité).

## II.4.LA STRATÉGIE: DIVISER POUR RÉGNER:

**Piloter un projet, c'est CONTRÔLER EN PERMANENCE son avancement et éventuellement CORRIGER quand cela s'impose.**

Pour pouvoir exercer ce contrôle permanent, il est indispensable de ménager, tout au long du déroulement du projet, des «points de test». A chacun de ces points doit être associée une «référence» décrivant l'état attendu à ce niveau en terme d'avancement de la production, de coût et de délais. Ces points de test doivent évidemment être assez nombreux pour pouvoir repérer et corriger les écarts avant qu'ils puissent trop s'aggraver.

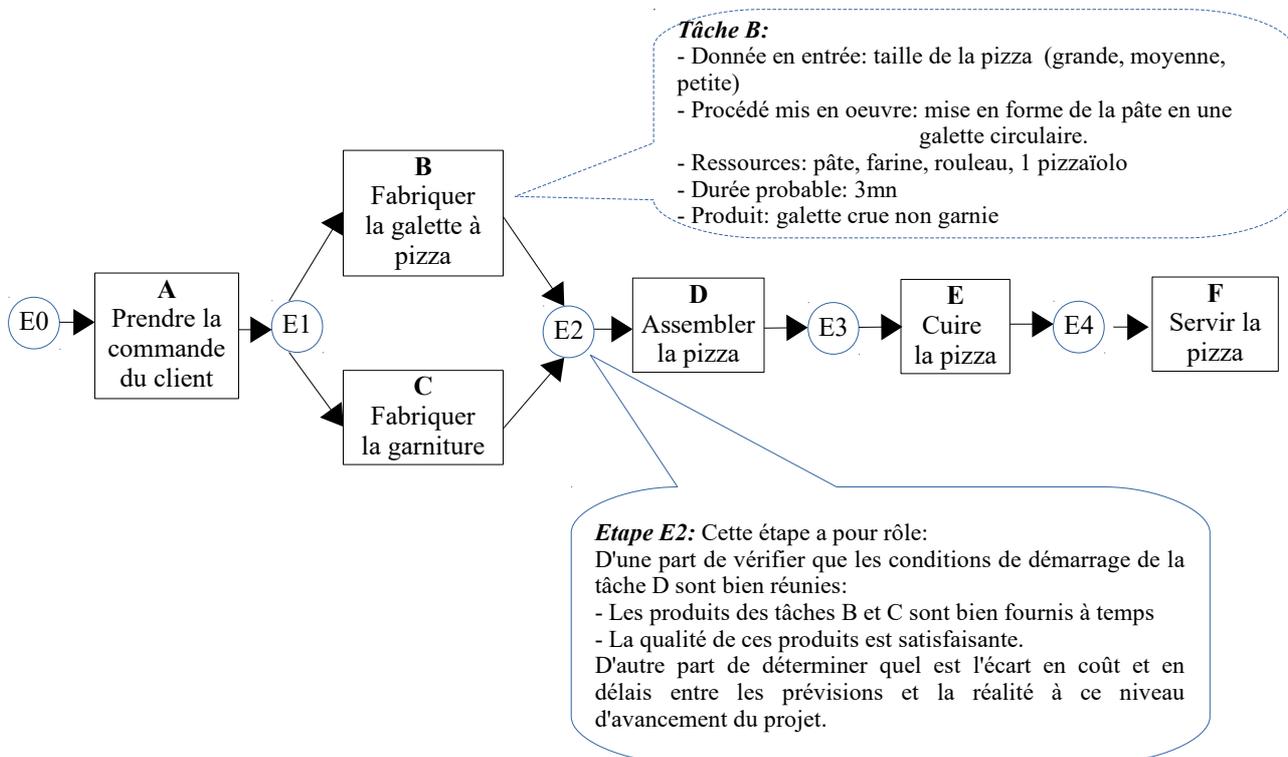
La stratégie adoptée, quelle que soient les méthodes employées, consiste donc toujours en une décomposition du projet en **tâches** élémentaires, d'un volume, d'un degré de complexité et d'une durée suffisamment faibles pour qu'il soit facile d'en déterminer:

- Les conditions de départ (données et commandes d'entrée).
- Le procédé à mettre en œuvre.
- Les ressources nécessaires.
- La durée probable.
- Les résultats attendus

On définit également entre les tâches des «étapes»: ce sont des points de rendez-vous que l'on se fixe pour vérifier l'évolution du projet (ce sont les points de test évoqués plus haut). Les tâches représentent donc des activités de production et les étapes des activités de contrôle.

Cette décomposition peut être représentée par un **graphe** dont les noeuds sont soit des tâches soit des étapes et dont les arcs représentent les liens de précedence entre les **tâches** et les **étapes**. Ce graphe est orienté, puisque les arcs ont un sens de parcours (le sens temporel). Ce type de graphe, dont les nœuds représentent des activités et dont les arcs symbolisent la communication entre ces activités (échange de données ou de commandes) est à la base de nombreux outils d'analyse (diagrammes de flux):

### EXEMPLE:



### I.1.1.LE CONTRÔLE DES COÛTS ET DES DÉLAIS:

En particulier, ces graphes sont à la base de la technique de **planification** la plus répandue, le PERT (Program Evaluation Review Technique). Cette technique, qui date des années 1960, a pour principe:

- La décomposition d'un projet en tâches élémentaires.
- La détermination, pour chacune de ces tâches, de ses conditions de départ, de son résultat, de sa durée probable (en fait, durée minimale, durée maximale et durée la plus probable).
- L'évaluation des ressources dont chaque tâche a besoin (heures de travail, fournitures...)
- L'Organisation de ces tâches en un graphe orienté suivant leur enchaînement logique.

En étudiant ce graphe, il est possible de prévoir assez précisément la durée probable du projet.

Il est également possible de visualiser en permanence l'avancement d'un projet, de détecter les écarts et de prévoir les conséquences d'un retard dans la réalisation d'une tâche sur la date de livraison finale.

On peut également contrôler la consommation des ressources, ce qui permet de prévoir toute dérive des coûts et les conséquences du manque de certaines ressources en terme de délais ou de qualité. Le PERT peut être appliqué avec profit à toutes les tailles de projets. Jusqu'à une vingtaine de tâches, un PERT peut être géré manuellement. Au dessus, il faut utiliser un logiciel.

Actuellement un certain nombre de logiciels s'appuyant sur le principe du PERT sont disponibles. Un des plus employés est le gratuiciel GANTT PROJECT, logiciel sous licence publique GPL.

### I.1.2.LE CONTRÔLE DE LA QUALITÉ:

Nous avons vu plus haut que pour contrôler, il faut d'une part évaluer, mesurer et d'autre part comparer le résultat avec une référence. Lorsqu'il s'agit d'une valeur numérique, comme la durée d'une tâche, son coût ou une quantité produite, la démarche tombe sous le sens: on évalue l'écart numérique entre la valeur mesurée et la référence, et on obtient par exemple un nombre de jours de retard par rapport au planning ou un montant de dépassement du budget.

Lorsqu'il s'agit de contrôler la qualité d'un produit, la démarche est beaucoup moins évidente. En effet, nous avons vu plus haut que la qualité d'un produit exprimait **l'écart qui existe entre les services que le produit peut rendre à l'utilisateur et ce que l'utilisateur en attend**. La référence à prendre est donc «ce que l'utilisateur attend du produit»

Si l'on veut évaluer le produit final, la référence qu'il faut utiliser est donc **l'expression du besoin** (telle qu'elle a été acceptée par le réalisateur): on s'attachera à démontrer dans quelle mesure le produit répond à cette expression de besoin. Cependant, ce contrôle ne peut intervenir qu'en fin de projet, alors qu'il est bien trop tard pour corriger une dérive.

La solution passe par un contrôle de qualité à la fin de chaque nœud du graphe: les produits résultant de chaque tâche sont soumis à des tests pour déterminer s'ils sont conformes à ce qu'on attendait. Ces tests peuvent aller de la simple inspection au déroulement de «jeux d'essai» élaborés. Ceci permet de valider la fin d'une tâche ou de la relancer pour effectuer des actions correctives.

### I.1.3.LA NOTION DE VALEUR:

La notion de VALEUR d'un produit ou d'un service englobe la notion de qualité et la notion de coût de possession.

Le COÛT DE POSSESSION d'un produit comprend:

- Le coût d'acquisition
- Les coûts de mise en service, d'entretien et de retrait du service
- Le coût d'élimination ou de recyclage

La VALEUR d'un produit peut se définir comme le rapport:

$$\text{QUALITÉ} / \text{COÛT DE POSSESSION}$$

## III.PRINCIPALES DÉMARCHES DE DÉVELOPPEMENT:

### III.1.PRINCIPAUX TYPES D'ACTIVITÉS LIÉES A UN PROJET:

#### III.1.1.INTRODUCTION:

Le but ultime de tout projet est de satisfaire des besoins. La satisfaction de ces besoins va être obtenue par la livraison d'une FOURNITURE qui peut être:

- Un OBJET: un logiciel, un véhicule, un ensemble de réservations pour un voyage, etc;
- Un SERVICE: des prestation de formation, de ménage, de surveillance, etc.

Dans ce document, nous nous préoccuperons surtout de la fourniture de LOGICIELS, et plus précisément de logiciels développés pour l'occasion (par rapport à des logiciels existants, choisis "sur étagère"). La fourniture sera donc envisagée comme un OBJET.

Dans un premier temps, les activités liées à la réalisation d'un projet peuvent être définies comme suit :

- SPÉCIFIER ce que l'on doit faire;
- CONCEVOIR comment on va le faire;
- RÉALISER l'objet du projet;
- VÉRIFIER que l'objet produit est bien conforme a ce qui est attendu.

Ces activités sont décrites ci-après :

#### III.1.2.SPÉCIFIER CE QUE L'ON DOIT FAIRE:

L'étude des besoins permet d'exprimer un certain nombre d'EXIGENCES par rapport à l'OBJET du projets. Ces exigences peuvent appartenir à divers domaines:

- Exigences FONCTIONNELLES (relatives aux FONCTIONS de l'objet);
- Exigences OPÉRATIONNELLES (relatives à l'utilisation de cet objet);
- Exigences concernant la SÉCURITÉ (des personnes, des biens, de l'environnement);
- Exigences concernant le respect des NORMES et de la RÉGLEMENTATION;
- Etc.

Le résultat de cette étude fait l'objet d'un document dont l'appellation la plus courante est "CAHIER DES CHARGES".

#### III.1.3.CONCEVOIR COMMENT ON VA LE FAIRE:

Il est évident que la création d'un objet complexe (un véhicule automobile, une sculpture monumentale, un bâtiment, un logiciel, etc.) exige un minimum de réflexion "en amont" (par exemple, la construction d'un bâtiment exigera au préalable une étude concernant la faisabilité compte tenu des contraintes, l'architecture la plus adaptée, les techniques à employer, etc. Cette activité intermédiaire entre l'expression du besoin et la réalisation est appelée CONCEPTION (DESIGN en anglais).

**REMARQUE** : l'activité de conception est souvent divisée en deux activités :

- La CONCEPTION PRÉLIMINAIRE dont le but est de définir une ARCHITECTURE GLOBALE;
- La CONCEPTION DÉTAILLEE dont le but est de concevoir chacun des constituants définis par l'architecture globale.

#### III.1.4.RÉALISER L'OBJET DU PROJET:

Il s'agit maintenant de CONSTRUIRE l'objet en respectant à la fois le cahier des charges et les principes de conception définis. Pour un logiciel, il s'agit de CODER les différents composants, de

les TESTER, puis de les ASSEMBLER entre eux (INTÉGRATION) pour former l'objet final.

**REMARQUE:** Dans un projet logiciel, l'activité de RÉALISATION est divisée en deux activités :

- L'activité de CODAGE-TESTS UNITAIRES (toujours étroitement imbriqués dans un développement de logiciels) ;
- L'activité d'INTÉGRATION .

### III.1.5.VÉRIFIER QUE L'OBJET PRODUIT EST BIEN CONFORME A CE QUI EST ATTENDU :

Il s'agit d'ÉVALUER, à l'aide de TESTS, dans quelle mesure l'objet produit répond aux exigences des utilisateurs concernés par le projet. En théorie, si le recueil des besoins a été exhaustif et si leur analyse a été bien menée, il suffit de mesurer la conformité du produit avec les exigences du CAHIER DES CHARGES.

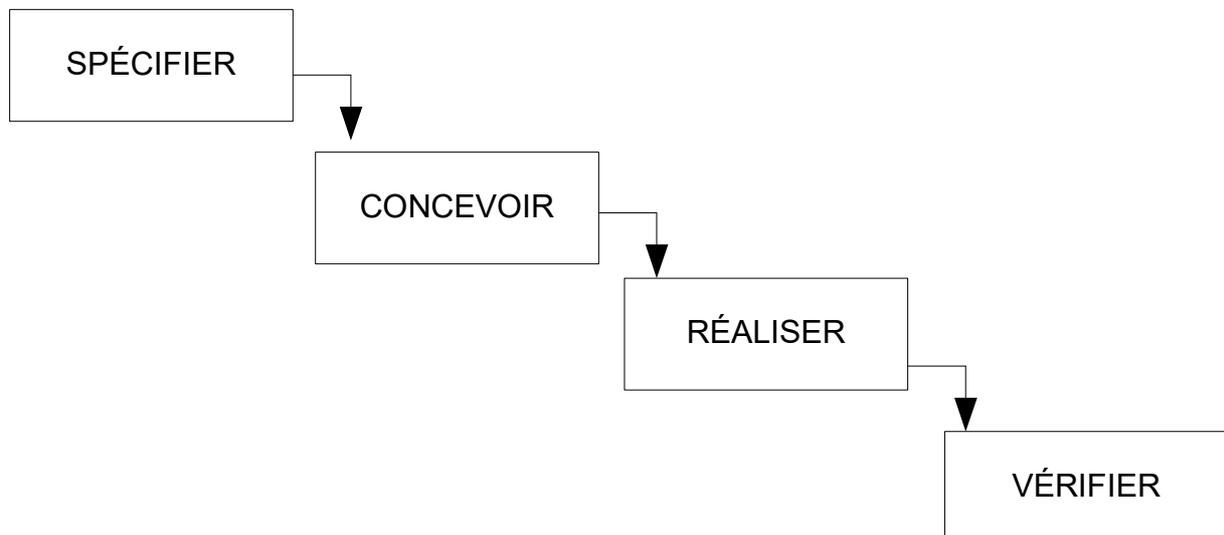
**REMARQUE:** on distingue en général deux phases de vérification :

- La RÉCEPTION du produit (vérification de la conformité avec le cahier des charges) ;
- la QUALIFICATION du produit (vérification de l'aptitude du produit à une utilisation dans un contexte d'utilisation opérationnel).

## III.2.MÉTHODES CLASSIQUES ET MÉTHODES AGILES :

### III.2.1.PRINCIPE DIRECTEUR DES MÉTHODES CLASSIQUES :

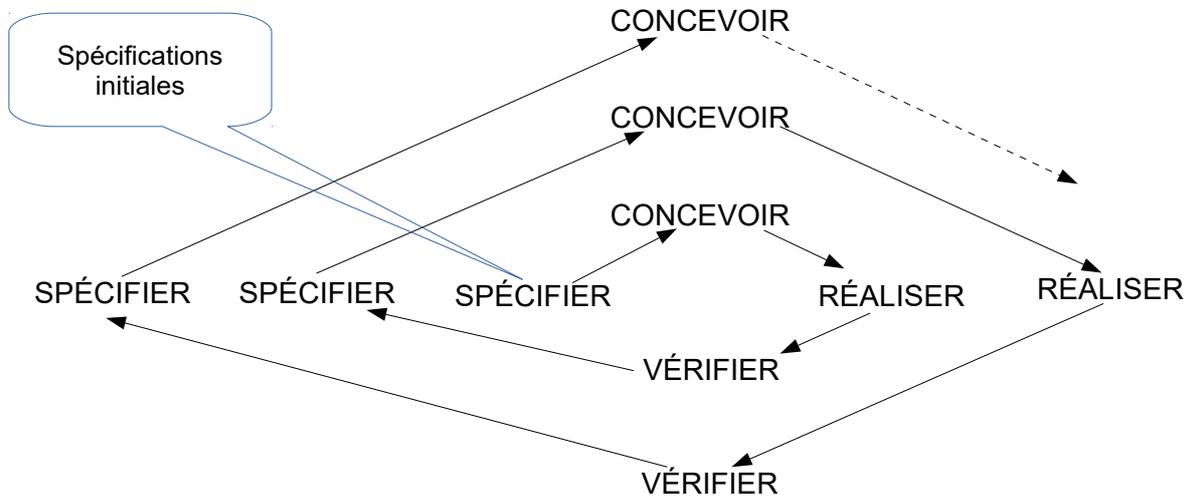
Jusqu'au début des années 2000, les méthodes de conduite de projet s'inspiraient pratiquement toutes du modèle "EN CASCADE". Selon ce modèle, l'idéal est qu'un projet soit réalisé par un enchaînement unique des 4 activités citées plus haut, sans superposition de ces activités ni retour en arrière :



Ceci implique que les travaux liés à une activités donnée soient menés d'une manière exhaustive et soient entièrement terminés avant de passer à l'activité suivante, **y compris la documentation afférente**. Le retour à l'activité précédente (et donc, la remise en cause de ses résultats) doit être **exceptionnel** et justifié par des **difficultés insurmontables** rencontrées dans l'activité en cours.

### III.2.2.PRINCIPE DIRECTEUR DES MÉTHODES AGILES:

Les méthodes agiles remettent en cause le principe de la cascade. Bien au contraire, elle prônent une coexistence permanente des 4 activités : l'objet du projet est atteint par une succession d'ITÉRATIONS du cycle [ SPÉCIFIER → CONCEVOIR → RÉALISER → VÉRIFIER ] :



A chaque cycle, de nouvelles exigences peuvent être rajoutées aux spécifications initiales (sur demande du client et avec l'accord du réalisateur). D'autre part, les spécifications déjà prises en compte peuvent être modifiées. Les travaux correspondant aux activités de spécification et de conception ne sont donc figés qu'à la fin du projet (et les documents correspondants également).

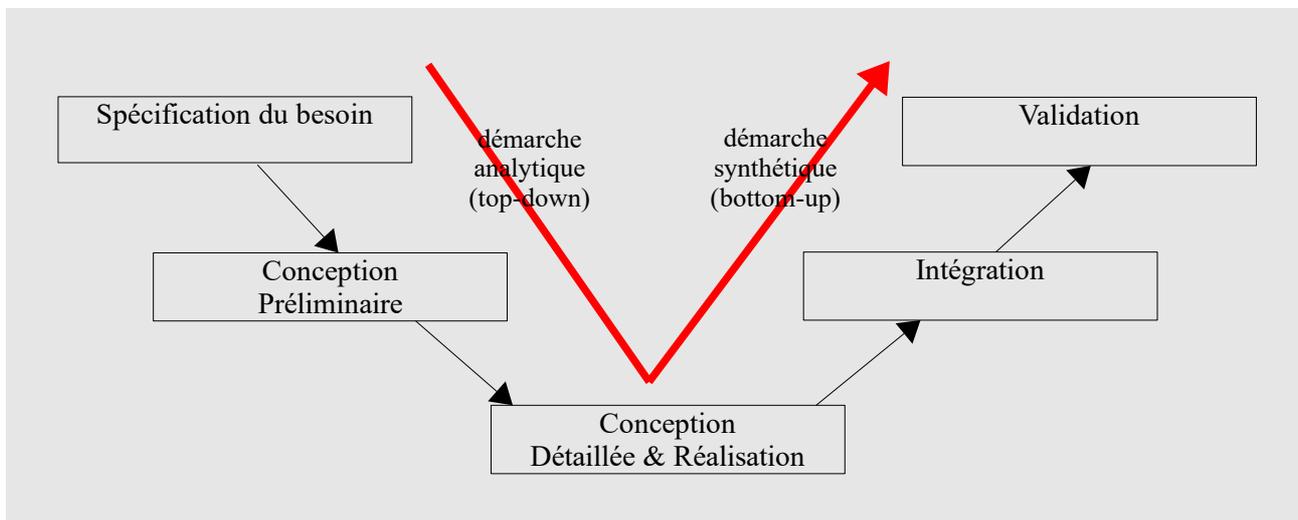
## IV. LES DÉMARCHES EN CASCADES - CYCLE EN V:

### IV.1. RAPPELS SUR LE CYCLE EN V:

#### IV.1.1. LES PRINCIPALES PHASES D'UN PROJET:

Le schéma ci-dessous résume le cycle en V en cinq phases fondamentales:

1. **La spécification du besoin:** l'objectif principal de cette phase est d'élaborer les clauses techniques du contrat entre le client et le réalisateur. Pour ce faire, il faut recueillir, classer, ordonner et caractériser les exigences du client vis à vis de l'objet du projet. L'objectif secondaire est, une fois le contrat de réalisation passé, d'évaluer et planifier la réalisation et de préparer la réception de l'objet par le donneur d'ordre (client).
2. **La conception préliminaire** définit les grandes lignes de l'architecture du système: décomposition en composants et définition des relations entre ces composants.
3. **La conception détaillée** est l'étude et la réalisation de chacun des composants définis lors de la phase précédente. Chaque composant réalisé est validé par des **tests unitaires**.
4. **L'intégration** consiste en l'incorporation de chacun des composants réalisés et validés dans l'environnement qui lui est assigné par l'architecture globale du système. Dans cette phase, l'intégration de chaque composant est validée par des **tests d'intégration**.
5. **La validation** permet de s'assurer que le système réalisé est bien conforme aux exigences du client telles qu'elles sont spécifiées dans le contrat. Pour ce faire, le système, après livraison au client, subit des tests «**de réception**», qui valident le système réalisé par rapport aux exigences du client, puis des tests «**de qualification**» qui permettent de s'assurer du bon fonctionnement du système en service régulier (c'est à dire dans les conditions d'utilisation réelle et dans la durée).



#### REMARQUE:

La phase de conception détaillée et réalisation est souvent éclatée en trois phases: conception détaillée proprement dite, réalisation et tests unitaires, ce qui porte à 7 le nombre de phases. Cette division n'a pas été retenue, car:

- Elle englobe difficilement le cas où un composant fait l'objet d'un sous-projet.
- Pour des raisons d'efficacité et de réduction des risques, la réalisation d'un composant se déroule rarement suivant un seul cycle «conception détaillée → réalisation → tests unitaires». En général, la démarche suivie est plutôt incrémentale. Elle implique un certain nombre d'itérations du cycle sur des versions de plus en plus complètes du composant.

Pour ces raisons, il nous a semblé préférable d'encapsuler la conception-réalisation des composants dans une seule phase.

## IV.1.2.CARACTÉRISTIQUES DU CYCLE EN V:

### IV.1.2.1.UN CYCLE EN DEUX TEMPS:

La dénomination «cycle en V» se justifie par le fait que la démarche se décompose en deux parties: la première partie est basée sur une démarche analytique qui tend à décomposer le problème en éléments simples (démarche que l'on qualifie souvent de «top-down» ou encore «descendante»), alors que la deuxième partie réalise la synthèse de la solution à partir de ces éléments simples (ce qui est plutôt caractéristique d'une démarche dite «bottom-up», ou encore «ascendante»).

La succession d'étapes d'analyse et d'étapes de synthèse n'est pas en elle-même caractéristique du cycle en V: on trouve cette dualité dans la plupart des démarches. Ce qui est caractéristique est le regroupement des activités d'analyse dans un premier temps et des activités de synthèse dans un deuxième temps.

### IV.1.2.2.SYMÉTRIE ENTRE PHASES:

On peut également remarquer que dans un cycle en V, chaque phase de décomposition (analyse) est associée à une phase de recomposition (synthèse):

- La phase de spécification du besoin, qui décompose le besoin en un certain nombre d'exigences peut être mise en correspondance avec la phase de validation, qui consiste à vérifier que la synthèses de ces exigences correspond bien au besoin initial.
- La phase de conception préliminaire, qui décompose l'objet du projet en un certain nombre de composants peut être mise en correspondance avec la phase d'intégration, qui consiste à assembler ces composants suivant les principes architecturaux définis en conception et à vérifier que cet assemblage aboutit bien à l'architecture prévue.

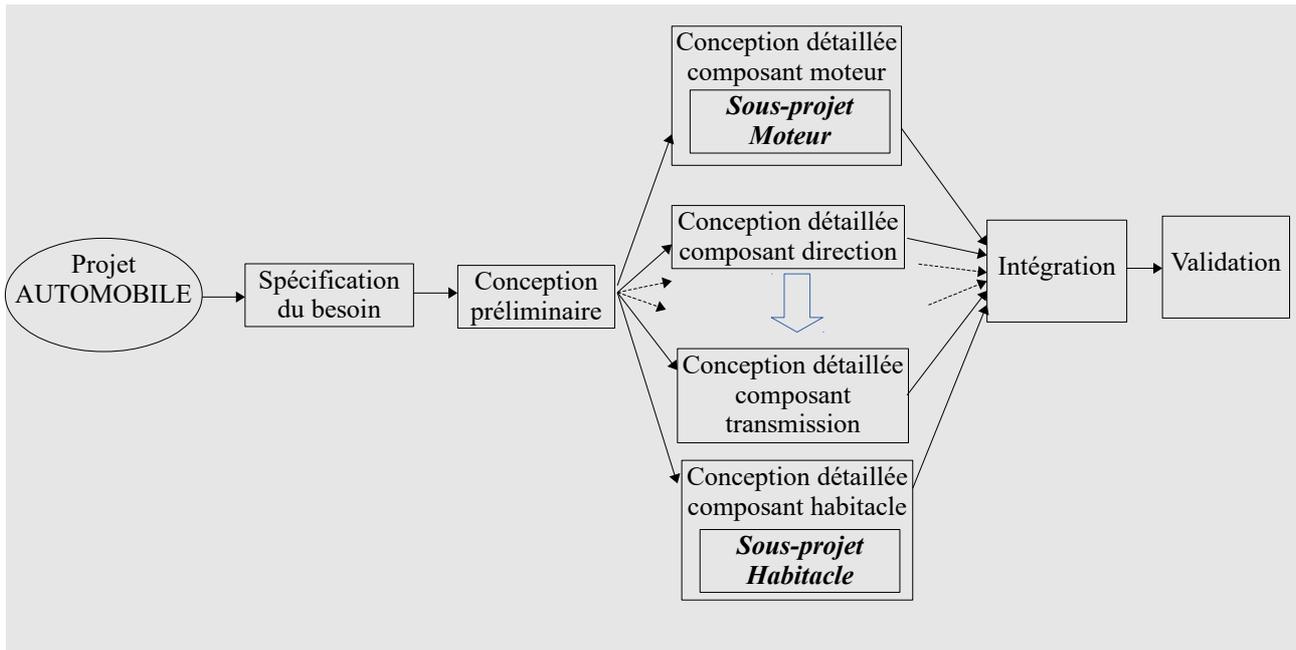
### IV.1.2.3.RÉCURSIVITÉ DE LA DÉMARCHE:

Lorsqu'un projet est très complexe ou très volumineux, les composants définis en conception préliminaire peuvent eux-même être très complexes ou très volumineux. La réalisation d'un tel composant peut faire l'objet d'un sous-projet (avec toutes les phases d'un projet). Ce sous projet pourra être confié à une équipe en interne ou à un sous-traitant.

#### **EXEMPLE:**

*Les composants d'un projet de développement d'une automobile pourraient être: le moteur, la direction, la transmission, etc. L'étude architecturale menée en phase de conception préliminaire va définir les services que l'on attend du composant moteur et ses relations avec le reste de la réalisation. Cependant, ce composant peut être jugé trop complexe pour baser sa réalisation uniquement sur ces éléments. De ce fait, il peut faire l'objet d'un sous-projet comprenant toutes les phases d'un projet. En particulier, on relance une phase de spécifications du besoin particulier concernant le moteur, qui va permettre d'ajouter aux exigences du client les exigences liées à l'intégration du moteur dans l'architecture choisie en conception préliminaire du projet principal.*

Le cycle en V peut donc être récursif, la réalisation de chaque composant pouvant donner lieu à un sous-projet. Le schéma ci-après représente graphiquement un exemple de planification du cycle en V:



### IV.1.3. INCONVÉNIENTS DU CYCLE EN V:

#### IV.1.3.1. RISQUES LIÉS AU RETARD DE DÉTECTION DES DÉFAUTS DE SPÉCIFICATION ET DE CONCEPTION PRÉLIMINAIRE:

Nous avons pu remarquer, à la lecture du contenu des phases, que si les travaux de conception détaillée donnent lieu à une validation immédiate (par les tests unitaires), les résultats des phases «spécification du besoin» et «conception préliminaire» ne sont validés qu'avec un certain retard:

- La Conception préliminaire n'est vraiment validée qu'en fin de phase «intégration», par les tests d'intégration (donc après le développement de tous les composants).
- La Spécification du besoin n'est vraiment validée que par la phase de validation (tests de réception et de qualification), donc en fin de projet.

La détection tardive des défauts dans les spécifications et la conception préliminaire peut entraîner la remise en cause d'une grande partie des travaux accomplis. Par exemple:

- Un défaut de conception préliminaire peut entraîner l'invalidation du développement de plusieurs composants.
- Un défaut de spécification peut entraîner la **remise en cause de l'architecture générale** (pouvant elle-même entraîner l'invalidation du développement de **nombreux composants**).

Les conséquences en matière de coûts et de délais peuvent donc être considérables.

Pour réduire les risques, les deux phases en question prévoient en général des «revues» (revues de spécifications, revues de conception) au cours desquelles les résultats des travaux sont inspectés. Des validations par modélisation peuvent également être effectuées. Cependant, si elles réduisent les risques, ces vérifications n'ont pas la même fiabilité qu'un test réalisé sur un composant ou un système réels.

#### IV.1.3.2. RISQUES LIÉS À LA NON IMPLICATION DU CLIENT LORS DES PHASES DE RÉALISATION:

Le cycle en V n'implique le client qu'au cours des phases «spécification du besoin» et «validation». En effet, le contrat de réalisation est signé en fin de spécification du besoin: de ce fait, dans les phases suivantes, les clauses techniques de ce contrat s'imposent aux deux parties. Il n'est donc, en théorie, plus question d'y changer quoi que ce soit sans remettre en cause ce contrat, même si l'une ou l'autre des parties découvre que des améliorations pourraient être apportées.

La livraison du produit se fait donc d'un bloc, en début de phase «validation», et ce n'est qu'à ce moment là que le client le découvre en réalité. Le risque est donc qu'il existe une grande divergence entre ce que le client attendait et ce qu'on lui présente. Cette divergence peut apparaître même si le produit se révèle conforme aux clauses techniques du contrat, car elle peut avoir des causes subjectives difficiles à cerner (aspect réel du produit, refus ou crainte du changement, incompréhension de certaines spécifications, etc).

Pour réduire ce risque, on intègre en phase de spécification du besoin la présentation au client de maquettes plus ou moins élaborées du produit (ou de certaines parties de ce produit), qui lui permettent d'une part de se familiariser avec son aspect et son ergonomie et d'autre part de reformuler ses exigences en conséquence (ceci est particulièrement vrai pour les interfaces homme-machine, qui doivent faire l'objet de maquettes élaborées).

#### **IV.1.4.AVANTAGES DU CYCLE EN V:**

En revanche, le cycle en V présente au moins trois avantages:

##### ***IV.1.4.1.ADAPTABLE A TOUS LES TYPES ET TOUTES LES TAILLE DE PROJET:***

La récursivité permet de définir des composants de premier niveau relativement complexes, puisque chacun d'entre eux pourra, si besoin est, faire l'objet d'un sous-projet délégué à un responsables de sous-projet, voire à des sous-traitants. Cette capacité permet de limiter le nombre de ces composants. De ce fait, quelles que soient la taille et la complexité du projet, le modèle architectural de premier niveau peut rester relativement simple, ce qui facilite beaucoup le contrôle de l'avancement.

##### ***IV.1.4.2.UNE GESTION RIGOUREUSE DES ASPECTS CONTRACTUELS:***

Si la non implication du client pendant les phases de réalisation a des inconvénients, elle a aussi l'avantage de permettre une gestion rigoureuse des rapports contractuels, car après la phase de spécification, les obligations respectives du client et du réalisateur sont bien définies et ne varient pas dans le temps. De plus, le planning de la réalisation ne peut être perturbé par la non disponibilités du client ou par un changement d'interlocuteur.

##### ***IV.1.4.3.UNE PARTICIPATION DU CLIENT RÉDUITE AU STRICT NÉCESSAIRE:***

Le client n'a pas forcément la volonté, les moyens humains ou les compétences pour s'impliquer dans les phases de réalisation (c'est d'ailleurs pour une de ces raisons qu'il a choisit de ne pas développer par lui-même).

L'implication du client n'est fructueuse que dans les domaines où celui-ci peut apporter une expertise réelle (en général: ergonomie d'utilisation du produit ou bien aspects se rapportant directement au «métier» du client). En revanche, pour tous les autres aspects de la réalisation, il se révélera probablement incompetent. Son implication «obligée» risque d'être perçue comme une perte de temps, voire une source de conflits.

De ce point de vue, le cycle en V libère le client pendant la phase de réalisation. Ceci lui permet de se concentrer sur la préparation de la phase de validation, où il se trouvera de nouveau très impliqué.

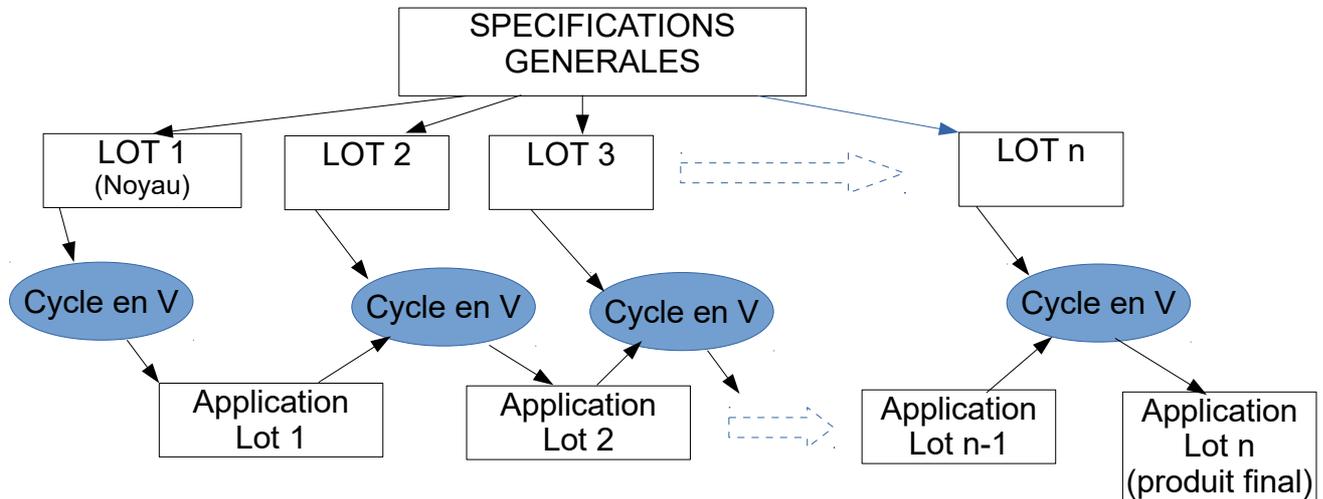
#### **IV.1.5.LE CYCLE EN V AVEC DÉVELOPPEMENT INCRÉMENTAL:**

Cette variante permet d'atténuer certains inconvénients du cycle en V "classique". En particulier, elle permet de beaucoup plus impliquer le client dans la vie du projet.

Dans cette démarche, après l'élaboration de SPÉCIFICATIONS classiques, on sépare (en accord avec le client), les exigences fonctionnelles en plusieurs LOTS. Le premier lot correspond au NOYAU de l'application. Celui-ci implémente les mécanismes généraux de l'application et les fonctions qui paraissent essentielles aux yeux du client. Ce lot est donc FONCTIONNEL (il implémente complètement certaines fonctions). Ce premier lot est réalisé et livré suivant la démarche du cycle en V classique.

Puis on relance un cycle en V pour réaliser le deuxième lot de spécifications, et ainsi de suite jusqu'à ce que le produit implémente toutes les exigences listées par les SPÉCIFICATIONS.

D'autre part, les lots de rang supérieur à 1 peuvent être CONDITIONNELS: ceci veut dire qu'après la livraison de chaque lot, le client se réserve le droit, de sa seule initiative, de terminer le processus de développement. En corollaire, les spécifications concernant les lots de rang supérieur à 1 NE SONT PAS FORCÉMENT FERMES ET DÉFINITIVES, laissant au client et au réalisateur des latitudes d'adaptation en cours de projet:



Cette démarche de réalisation est compatible avec le type de marché public appelé MARCHÉ A TRANCHES CONDITIONNELLES.

## IV.2.DESCRPTION DE LA DÉMARCHE GÉNÉRALE:

### IV.2.2.DIFFÉRENTS TYPES D'ACTIVITÉS:

Nous avons, plus haut, découpé le cycle en V en cinq phases principales. Chacune de ces phases recouvre des activités de nature différente que nous allons séparer en trois catégories:

- Les TRAVAUX PRÉALABLES
- Les TRAVAUX DU PROCESSUS DE DÉVELOPPEMENT
- Les TRAVAUX COMPLÉMENTAIRES.

Les **travaux préalables** rassemblent les activités dont le but est de préparer directement les travaux de réalisation de l'objet du projet (spécification du besoin, contractualisation, planification de la réalisation). Toutes ces activités appartiennent à la phase de **spécification du besoin**. Elles sont sous la responsabilité principale du **maître d'ouvrage**, qui représente les intérêts du **client** (que l'on appelle aussi «donneur d'ordre»).

Comme leur nom l'indique, les travaux du processus de développement comprennent toutes les activités de conception et de réalisation (conception préliminaire et détaillée, intégration). Elles sont sous la responsabilité principale du **maître d'oeuvre**, qui représente les intérêts du **fournisseur** (ou **réalisateur**).

Les **travaux complémentaires** regroupent les activités de préparation et de réalisation de la livraison du produit au client: validation du produit par client et qualification opérationnelle.

#### REMARQUES:

- *Chaque fourniture issue des différentes phases et des différents types d'activités fait l'objet d'une procédure de validation. En cas d'échec de la validation, le cycle réalisation-validation de la fourniture est itéré jusqu'à ce que la fourniture soit jugé satisfaisante. Ce mécanisme s'applique à toutes les fournitures, y compris la documentation.*
- *La description détaillée des documents et produits issus des différentes phases et sous-phases figure dans les annexes.*

## IV.2.3.LES TRAVAUX PRÉALABLES:

### IV.2.3.1.GÉNÉRALITÉS:

L'objectif principal des travaux préalables est la définition **externe** du produit. C'est la description du produit du point de vue des utilisateurs et du client: services attendus, performances, aspect, forme, encombrement, niveau de bruit acceptable, etc). Cette définition est à la base de la rédaction des clauses techniques du contrat de fourniture ou de réalisation passé entre le **maître d'ouvrage** et le **maître d'Oeuvre**. Bien distinguer les travaux préalables du reste du projet permet de ne pas préjuger de la solution de réalisation avant d'avoir une idée complète du problème: les travaux préalables définissent **ce que l'on doit faire** et non **comment** on doit le faire.

Les autres objectifs des travaux préalables sont d'organiser et de planifier la réalisation (dans les limites prévues par le contrat) et de fixer les modalités contractuelles de la réception de l'objet.

Ces activités exigent une collaboration étroite du **maître d'ouvrage** avec le **maître d'oeuvre**.

### IV.2.3.2.LES ÉTAPES DES TRAVAUX PRÉALABLES:

Les travaux préalables appartiennent entièrement à la phase de spécification du besoin. Ils font partie de la **maîtrise d'ouvrage**. Le tableau ci-après décrit les activités entrant dans le cadre des travaux préalables. La nature et le contenu des différents documents et produits sont explicités en annexe.

ÉTAPES	ACTIVITÉS	FOURNITURES CONSOMMÉES	RESPONSABLE	VALIDATION	PRODUITS
Recueil et Analyse du besoin	Énoncer, caractériser et justifier les besoins externes sous forme d'exigences, dans le langage du client et des utilisateurs potentiels.	Recueil des besoins exprimés par le client et les différents utilisateurs du produit.	Maître d'ouvrage	Par revue de document	Cahier Des Charges Fonctionnel (C.D.C.F)
Spécifications techniques	Exprimer les besoins sous forme d'exigences techniques, caractérisées par des critères.  Décrire et caractériser les interfaces du système avec la périphérie.  Décrire les modalités générales de la réception du produit  Justifier les exigences.  Décrire les différents modes d'utilisation du produit et les procédures associées.	C.D.C.F  Documentation des systèmes destinés à être connectés (spécification des interfaces)  Contributions des utilisateurs pour compléments d'information et travaux sur maquettes	Maître d'ouvrage	Par revue de documents  Par maquettage, modélisation, simulation.	Spécification Technique de besoin (S.T.B).  Dossier de Justification des Exigences (D.J.E).  Manuel Opérateur (M.Op)
Contractualisation	Rédiger le <b>cahier des charges</b> de la réalisation.  Finaliser le <b>contrat de réalisation</b>	S.T.B  soutien juridique	Maître d'ouvrage et Maître d'œuvre	Sans objet	Cahier des Charges  Contrat de réalisation
Organisation et planification du projet	Déterminer les différentes tâches à accomplir, évaluer leur durée et les ressources nécessaires à leur réalisation  Établir un planning de la réalisation.	S.T.B D.J.E M.Op	Maître d'ouvrage et maître d'œuvre.	Par Revue	Organigramme des tâches (O.T)  Plan de Développement (P.Dv)

### IV.2.3.3.RECUEIL ET ANALYSE DU BESOIN:

#### IV.2.3.3.1 .OBJECTIFS:

Déterminer et caractériser les exigences des utilisateurs vis à vis de l'objet du projet. Ces exigences peuvent concerner en particulier:

- Les fonctionnalités attendues de l'objet et les performances associées (exigences fonctionnelles).
- Les conditions d'utilisation de l'objet (exigences opérationnelles).
- La conception ou la réalisation (normes techniques ou environnementales à respecter, méthodes imposées, etc..)

#### IV.2.3.3.2 .CONTENU:

En premier lieu, les besoins des différents utilisateurs potentiels sont recueillis. Puis ces informations sont analysées, regroupées et classées afin d'établir une liste exhaustive des exigences et de les caractériser le plus précisément possible. Le résultat de ces travaux constitue la substance du Cahier des Charges Fonctionnel (**C.D.C.F**). Il est également recommandé de recueillir la justification des besoins exprimés.

Des versions successives du C.D.C.F sont présentées au client et aux utilisateurs pour critiques et propositions d'amélioration jusqu'à ce que le résultat soit jugé satisfaisant.

Pour cette activité, il est recommandé de s'appuyer sur une démarche méthodologique relativement simple et peu formalisée, afin que les utilisateurs amenés à collaborer puissent la comprendre sans être obligé de suivre une formation importante. Le C.D.C.F doit être rédigé **dans des termes appartenant aux métiers des utilisateurs et du client** et non en termes «informatiques».

### IV.2.3.4.SPÉCIFICATIONS TECHNIQUES:

#### IV.2.3.4.1 .OBJECTIFS:

- Exprimer, caractériser et justifier les exigences en termes techniques (ici, en termes informatiques).
- Définir les modalités de la réception de l'objet
- Créer le Manuel Opérateur du produit.

#### IV.2.3.4.2 .CONTENU:

#### SPÉCIFICATION TECHNIQUE DU BESOIN:

La Spécification Technique de Besoin se déduit en grande partie du C.D.C.F. Elle traduit en termes techniques les exigences exprimées par le C.D.C.F.

#### **Exemple de différence entre le C.D.C.F et la S.T.B:**

*Soit un projet concernant la création d'un système de stockage de fichiers vidéo:*

- *Au niveau du C.D.C.F, une exigence pourrait être: «le produit doit être capable de stocker au moins cinquante films de long métrage». Cette exigence a un sens précis pour des utilisateurs potentiels, car elle est exprimée dans des termes qu'ils comprennent. Par contre, elle est peu exploitable pour la réalisation, car trop vague (quelle est la taille d'un film ? Faut-il prendre 50 fois la taille du plus grand long métrage possible ou 50 fois la taille moyenne? etc)*
- *Au niveau de la S.T.B, cette exigence deviendra: «le produit doit être capable de stocker jusqu'à 50 fichiers vidéo compressés de 50 mégabits chacun. Cette rédaction a beaucoup moins de sens pour les utilisateurs. Par contre, elle est beaucoup plus compréhensible et exploitable pour les réalisateurs.*

*D'où l'utilité de disposer des deux documents. ...*

De plus, la S.T.B définit et caractérise également les exigences et contraintes induites par les

interfaces physiques du produit.

Enfin, elle décrit les modalités de la réception de l'objet (plan de réception):

- Procédure générale de réception.
- Liste des fournitures devant faire l'objet d'une réception (fournitures dûes par le client et dûes par le réalisateur)
- Planification globale des travaux de réception (préparation et réalisation).

La S.T.B. est souvent validée par un procédé de modélisation (Exemple: modélisation fonctionnelle S.A.D.T.).

#### *JUSTIFICATION DES EXIGENCES:*

La S.T.B. est parfois accompagnée d'un document justifiant chaque exigence, le Dossier Justificatif des Exigences (**D.J.E**). Ce document permet de garder la traçabilité du besoin. La S.T.B. et le D.J.E. peuvent souvent être fusionnés.

#### *CRÉATION DU MANUEL OPÉRATEUR:*

Nous avons vu que la S.T.B. spécifie les exigences techniques concernant les interfaces du système avec la périphérie, et en particulier les interfaces homme-machine. Ce travail engendre en grande partie la substance du Manuel Opérateur, dont l'objectif est de décrire les différents modes d'utilisation du produit et les procédures associées. La rédaction du manuel opérateur (**MOp**) conjointement avec la S.T.B. permet donc de rationaliser l'effort de production de ces documents: la S.T.B. peut faire référence au manuel opérateur pour décrire les interfaces homme-machine, ce qui minimise l'effort de rédaction. De plus, pour les futurs utilisateurs, le manuel opérateur est souvent beaucoup plus «parlant» que la S.T.B. Il sera donc un utile complément à celle-ci quand il s'agira de faire valider la phase de recueil et analyse du besoin par les utilisateurs et le client.

### *IV.2.3.5.CONTRACTUALISATION:*

#### *IV.2.3.5.1 .LE CAHIER DES CHARGES ET LE CONTRAT*

Le contrat entre le client et le réalisateur est conclu sur la base d'un **cahier des charges** de la réalisation. Ce document comprend des **clauses techniques** qui définissent l'engagement du réalisateur dans le domaine des fonctionnalités et des performances du produit, et des **clauses administratives et commerciales** qui concernent les conditions de sa réalisation et de sa livraison (coûts, délais, pénalités de retard, régime juridique, etc). Les clauses techniques sont directement tirées de la S.T.B. (Parfois même, la S.T.B. est simplement annexée au document).

#### *PROCÉDURE PAR APPEL D'OFFRES:*

Le fournisseur (ou réalisateur) peut être choisi au moyen d'un **appel d'offre**: Les candidats à la réalisation répondent au cahier des charges par des **propositions techniques et commerciales** qui détaillent les solution techniques proposées, leur coût, leurs délais de réalisation, etc. Le client aura à choisir entre ces différentes propositions.

#### *PROCÉDURE DE GRÈS A GRÈS:*

Le contrat peut également être passé «**de grès à grès**», après des négociations entre le client et ses fournisseurs potentiels sur les termes du cahier des charges. De ce fait, les clauses techniques finalement adoptées peuvent différer du contenu de la S.T.B, qui devra être adaptée en conséquence. C'est le cas en particulier quand un appel d'offre est resté infructueux, obligeant le client à revoir ses exigences.

#### **IV.2.3.6. ORGANISATION ET PLANIFICATION DU PROJET:**

##### **IV.2.3.6.1 .OBJECTIFS:**

- Identification de l'ensemble des travaux à accomplir dans le cadre du projet et organisation de leur exécution.
- Planification de ces travaux (dans le respect des clauses administratives et commerciales concernant les délais)
- Mise en place des structures de contrôle de l'avancement et de la gestion des ressources.

##### **IV.2.3.6.2 .CONTENU:**

Cette sous-phase se situe directement après la signature du contrat de réalisation. Elle concerne à la fois le client (maître d'ouvrage) et le réalisateur (maître d'œuvre). Le maître d'œuvre s'occupe de l'organisation et de la planification des phases de développement, le maître d'ouvrage intègre ces données dans l'organisation et la planification globale.

Dans un premier temps, l'ensemble des tâches qui concourent à la réalisation du projet sont identifiées, y compris les tâches de direction de projet et de soutien. Le résultat de cette activité est souvent présenté sous une forme graphique, appelée «Organigramme des Tâches (**O.T**)».

Dans un deuxième temps, chaque tâche est décrite, organisée et attribuée à un responsable. Des ressources (humaines, matérielles, financières, etc) lui sont allouées. Le planning global du projet est élaboré.

Ces activités sont consignées dans le document Plan de Développement (**P.Dv**).

#### IV.2.4. TRAVAUX DU PROCESSUS DE DÉVELOPPEMENT:

Ces travaux se situent après la passation du contrat avec le réalisateur. Ils font partie de la **maîtrise d'œuvre**. Leurs objectifs sont:

- La **définition interne** du produit (c'est à dire l'étude et le choix d'une solution architecturale pour le produit)
- Sa **réalisation, et son intégration** dans son environnement d'utilisation.

##### IV.2.4.1. ÉTAPES DES TRAVAUX DU PROCESSUS DE DÉVELOPPEMENT:

Les travaux du processus de développement englobent 3 phases du cycle en V: La conception préliminaire, la conception détaillée-réalisation et l'intégration. Le tableau ci-après détaille les activités correspondantes. La nature et le contenu des différents documents et produits sont explicités en annexe.

PHASE / ÉTAPE	ACTIVITÉS	FOURNITURES CONSOMMÉES	RESPONSABLES	VALIDATION	DOCUMENTS ET FOURNITURES PRODUITS
<b>Conception Préliminaire</b> / Détermination de la solution architecturale	Identification et caractérisation des différents composants architecturaux et des relations entre ces composants.  Validation de la solution architecturale: - du point de vue fonctionnel (vérification du recouvrement des exigences par les composants) - du point de vue des performances (modélisation, maquetages)	<b>S.T.B</b>	Maître d'œuvre	Revue de documents  Maquetages et modélisation	Document de Conception Préliminaire ( <b>D.C.P.</b> )
<b>Conception détaillée/</b> organisation de la réalisation	Regroupement des composants en lots de réalisation, attribution de chaque lot à un réalisateur (développeur ou équipe de développeurs interne ou sous-traitant)  Travaux préalables correspondant à tous les lots nécessitant le lancement d'un sous-projet	<b>D.C.P</b>	Maître d'œuvre  Responsables de lot	Revue de documents	Notes d'organisation nommant les responsables de lots de réalisation  Plan d'intégration ( <b>P.I</b> )  <b>S.T.B</b> de sous-projet dans le cas d'un lot qui le justifie ou d'un lot sous-traité
<b>Conception détaillée/</b> Réalisation	Conception et réalisation des composants contenus dans chaque lot de réalisation et tests unitaires de ces composants	<b>D.C.P.</b>  <b>S.T.B</b> de sous-projet	Responsables de lots	Tests unitaires de composants	Dossier de Définition ( <b>D.D</b> ) de chaque composant ou documents afférents au sous-projet  Composants développés
<b>Intégration</b>	Intégration de chaque composant à l'architecture globale et tests d'intégration	Plan d'intégration ( <b>P.I</b> )	Maître d'œuvre  Responsables de lots	Tests d'intégration	Résultats des tests d'intégration  Objet prêt pour la validation

#### IV.2.4.2. CONCEPTION PRÉLIMINAIRE:

##### IV.2.4.2.1 .OBJECTIF:

Concevoir une architecture répondant aux exigences exprimées par la S.T.B.

##### DÉFINITIONS:

L'**architecture d'un système** quelconque décrit et caractérise les **composants** de ce système, et les **relations** qui existent entre eux.

Un **composant** est un sous-ensemble d'un système dont les fonctions propres et les interactions avec le reste du système sont entièrement définies, et qui peut, de ce fait, faire l'objet d'un développement autonome.

Dans un **système informatique**, les composants pourront être des logiciels, des matériels, ou un mélange des deux (composants matériels munis de logiciels «embarqués»). La nature des composants logiciels pourra varier suivant la démarche de conception adoptée (fonctions, lots, objets, agents, etc.). Les **relations** qui existent entre les composants peuvent être divisées en deux catégories:

- Les relations de nature **statique**, qui correspondent à des liens structurels entre les composants (dépendance fonctionnelle d'un composant vis à vis d'un autre, relations d'héritage, de composition, etc.)
- Les relations de nature **dynamique**, qui caractérisent le fonctionnement du système (flots de données ou de contrôle, signaux, messages, etc.)

##### IV.2.4.2.2 .CONTENU:

La phase de spécification des besoins tente de répondre à la question «que faut-il réaliser ?». Elle effectue une définition **externe** de l'objet. La phase de conception préliminaire, tente de répondre à la question «comment l'objet du projet doit-il être réalisé pour répondre aux exigences?». Il s'agit donc de la définition **interne** de cet objet, c'est à dire de son architecture. Le problème est de passer de la définition externe à la définition interne sans trahir les exigences exprimées par la S.T.B.

Dans certaines méthodes de conception, ce passage est réalisé par une démarche de décomposition fonctionnelle: les fonctionnalités sont décomposées en sous-fonctionnalités qui sont à leur tour décomposées, et ainsi de suite jusqu'à ce qu'il soit possible de concevoir, pour chaque sous-fonctionnalité résultante, un composant capable de la satisfaire (exemple: outil F.A.S.T). Cependant, ce type de démarche a pour inconvénient de produire des architectures peu évolutives et peu performantes en termes de rapidité et de réactivité.

De ce fait, une démarche par prototypages successifs est souvent préférée:

- Dans un premier temps, on tente d'encapsuler les **traitements** nécessaires à la réalisation des exigences fonctionnelles de l'objet et les **données** sur lesquelles ces traitements agissent dans des **composants**, puis on identifie les relations entre ces composants (on a donc élaboré une solution architecturale).
- Dans un deuxième temps, on vérifie que la solution satisfait bien aux exigences (par revues, modélisation, etc).
- On itère la démarche jusqu'à ce que la solution soit satisfaisante.

L'identification des composants est guidée et contrainte par la démarche de conception choisie et par la satisfaction des exigences.

##### REMARQUES:

L'architecture d'un système doit être étudiée suivant au moins deux points de vue:

- Le point de vue **dynamique** qui modélisera le fonctionnement interne de l'objet, correspondant aux services qu'il est sensé rendre à ses utilisateurs.
- Le point de vue **statique** qui étudiera la manière dont il est réalisé.

A part la recherche d'une architecture adaptée à la satisfaction des exigences, l'un des objectifs les plus importants de la conception préliminaire est de permettre une évolution harmonieuse de l'objet, sans dégradation. Il est donc extrêmement important d'exposer et de justifier les principes directeurs de la conception.

La description de la solution architecturale fait l'objet du Document de Conception Préliminaire (D.C.P).

**IV.2.4.3. CONCEPTION DÉTAILLÉE-ORGANISATION DE LA RÉALISATION:****IV.2.4.3.1 . OBJECTIF:**

Organiser les travaux de réalisation des différents composants et préparer leur intégration.

**IV.2.4.3.2 .CONTENU:**

Le maître d'oeuvre regroupe les composants en **lots de réalisation**, pouvant comprendre un ou plusieurs composants. Chaque composant va être attribué à un **responsable de lot**. Celui-ci peut être un développeur seul ou le chef d'une équipe. Il peut s'agir d'un sous-traitant.

D'autre part, le maître d'oeuvre rédige le Plan d'intégration (**P.I**), dont le but est d'organiser l'intégration des différents composants à l'architecture globale, au fur et à mesure de leur livraison par les responsables de lots. Le plan d'intégration définit l'ordre dans lequel l'intégration doit se dérouler, ainsi que les modalités d'intégration de chaque composant (en particuliers, les «tests d'intégration» qui permettent de valider l'intégration de chaque composant et les tests de non régression correspondants). En fin d'intégration, le produit aura été entièrement testé par rapport aux spécifications du D.C.P.

**IV.2.4.3.3 .DÉCOUPAGE EN LOTS DE RÉALISATION:**

Les principes qui président au regroupement en lots dépendent de la stratégie adoptée par le chef de projet. ils peuvent être d'ordre technique (de mêmes natures que ceux qui président au regroupement des activités en lots), d'ordre opérationnel (regrouper des composants dont le développement exigent les mêmes compétences ou utilisent les mêmes outils), d'ordre stratégique (sous-traiter des développement dont l'acquisition de l'expertise en interne n'est pas jugée utile), etc.

**IV.2.4.4. CONCEPTION DÉTAILLÉE-RÉALISATION:****IV.2.4.4.1 . OBJECTIF:**

Développer les composants, conformément aux spécifications contenues dans le document de Conception Préliminaire (**D.C.P**).

**IV.2.4.4.2 . CONTENU:****CAS DES COMPOSANTS NON SOUS-TRAITES:**

Les composants ne présentant pas de difficultés particulières peuvent être réalisés sur la base des spécifications contenues dans le Document de Conception Préliminaire (D.C.P).

Dans le cas de composants plus complexes, ils peuvent nécessiter une étude architecturale interne pouvant aboutir à la définition de sous-composants et nécessiter une phase d'intégration (ceci correspond à un mini-projet réduit aux travaux du processus de développement).

**CAS DE COMPOSANTS SOUS-TRAITES:**

Dès l'instant qu'un composant est sous-traité, il fait l'objet d'un contrat de réalisation. De ce fait, il est nécessaire de le gérer dans le cadre d'un sous-projet comprenant toutes les phases du cycle en V.

**IV.2.4.5. TESTS UNITAIRES:**

Les tests unitaires valident un composant par rapport aux spécifications architecturales contenues dans le D.C.P. La composition de ces tests (décrite dans le Dossier de Tests Unitaires) doit faire l'objet d'un accord entre le maître d'œuvre et le responsable de lot.

#### **IV.2.4.5.1 .FOURNITURES PRODUITES:**

- Cas de composants ne faisant pas l'objet de sous-projets: Les Dossiers de Définition (**D.D**) de chaque composant: définition et justification de l'architecture interne, description des sous-composants.
- Cas de composants faisant l'objet de sous-projets: Le Dossier de Définition est constitué par la documentation attachée au sous-projet.
- Les composants réalisés (composants matériels ou codes sources et exécutables)
- Les Dossiers de Tests Unitaires, accompagnées des compte-rendus de tests.

#### **IV.2.4.6.INTÉGRATION:**

##### **IV.2.4.6.1 . OBJECTIF:**

Intégrer chaque composant développé à l'architecture globale et valider son fonctionnement dans cet environnement.

##### **IV.2.4.6.2 . CONTENU:**

Chaque composant livré par les responsables de composants est intégré dans l'architecture du produit, puis son fonctionnement est validé par des tests d'intégration. Des tests de non régression sont ensuite effectués pour s'assurer que le fonctionnement des composants précédemment intégrés n'est pas affecté par la présence du nouveau composant.

Pour des raisons d'optimisation de la durée du développement, on n'attend pas la fin des travaux de conception détaillée pour commencer l'intégration: chaque composant doit être intégré le plus rapidement possible après sa livraison. De ce fait, si globalement, l'activité se déroule conformément aux directives fournies par le Plan d'Intégration (**P.I**), réalisé lors de la sous-phase «organisation de la réalisation», ce plan est susceptible d'être aménagé en fonction des aléas du développement des lots.

## IV.2.5. TRAVAUX COMPLÉMENTAIRES:

Ces travaux se répartissent entre la phase de spécification du besoin et la phase de réception. Ils font partie de la **maîtrise d'ouvrage**. Leur objectif est la préparation et l'exécution de la **réception** de l'objet par le client.

### IV.2.5.1. ÉTAPES DES TRAVAUX COMPLÉMENTAIRES:

PHASE / ÉTAPE	ACTIVITÉS	FOURNITURES CONSOMMÉES	RESPONSABLES	VALIDATION	DOCUMENTS ET FOURNITURES PRODUITS
<b>Spécifications techniques/</b> Préparation de la réception	Identification des tests de réception à effectuer, description du principe de ces tests, planification.	<b>S.T.B</b> Manuel opérateur ( <b>M.Op</b> )	Maître d'ouvrage et maître d'œuvre	Par revue	Cahier de Réception Général ( <b>CRG</b> )
<b>Phases de développement/</b> Préparation de la réception	Pour chaque point à tester, déterminer: – Les scénarios de test – Les outils de test – Les jeux d'essai et les résultats attendus	<b>CRG</b> <b>M.Op</b>	Maître d'ouvrage et maître d'œuvre	Par revue	Cahiers de Réception Détaillés ( <b>CRDi</b> )
<b>Réception /</b> Validation	– Déroulement des tests de réception – Relevé des anomalies et réserves éventuelles.  <b>Nota:</b> une réserve majeure invalide la réception toute entière, occasionnant un retour en développement pour correction de l'anomalie, puis une reprise de la validation.	<b>CRDi</b>	Maître d'ouvrage	Par tests de réception prévus aux CRDi	Comptes-rendus de tests de réception (annexés aux CRDi)
<b>Réception /</b> Qualification opérationnelle	Cette phase consiste à tester le produit en utilisation opérationnelle, afin de détecter éventuellement les problèmes d'exploitation et d'y remédier.  Durant cette phase, les réserves mineures émises lors de la réception doivent être levées par le développeur.	<b>S.T.B</b> <b>M.Op</b> <b>Comptes-rendus de tests de réception</b>	Maître d'ouvrage  Responsable d'exploitation	Par utilisation en condition opérationnelle suivant une gamme de scénarios représentatifs.	Comptes-rendus de tests de qualification

### IV.2.5.2. PRÉPARATION DE LA RÉCEPTION DURANT LA PHASE DE SPÉCIFICATION TECHNIQUE:

#### IV.2.5.2.1 .OBJECTIF:

Fixer, d'un commun accord entre le maître d'ouvrage et le maître d'œuvre, les principes généraux de la procédure de réception de l'objet et préparer celle-ci.

#### IV.2.5.2.2 .CONTENU:

Cette activité prend place après la passation du contrat de réalisation. Elle a pour but de déterminer et décrire les modalités du déroulement de la réception et de la qualification opérationnelle de l'objet:

- Détermination des modalités de la validation et de la qualification opérationnelle (procédure générale, responsabilités des différents acteurs)
- Pour chaque fourniture concernée par la réception, détermination des tests de validation à

effectuer, du principe de réalisation de ces tests et des ressources nécessaires à leur mise en oeuvre (la liste des fournitures a été fixée par la S.T.B).

- Organisation et planification du déroulement des opérations de réception (enchaînement optimal des tests).

Cette activité s'appuie directement sur les résultats de la phase de spécification technique: les points de test sont déterminés à partir des critères issus de la caractérisation des exigences (S.T.B.). Le résultat de ces travaux est consigné dans le Cahier de Réception Général (C.R.G).

#### **IV.2.5.2.3 .PÉRIMÈTRE D'UNE RÉCEPTION:**

La liste des fournitures concernées par la réception doit être définie par la S.T.B. Ceci peut concerner non seulement l'objet lui-même, mais aussi:

- L'ensemble de la documentation du projet.
- Les fournitures nécessaires à l'intégration de l'objet dans son environnement opérationnel.
- Les prestations de formation des personnels du client.
- Et, en fait, toute autre fourniture prévue par le contrat.

Le périmètre d'une réception ne comprend pas uniquement des fournitures dues par le réalisateur: la plupart du temps, l'intégration de l'objet dans son environnement opérationnel implique que le **client** s'engage sur un certain nombre de points qui conditionnent son fonctionnement: configuration des locaux d'accueil, conditions d'accès à l'énergie, climatisation, interfaces avec l'environnement, etc. Ces points doivent faire l'objet d'une validation, au même titre que les fournitures du réalisateur.

#### **IV.2.5.3.PRÉPARATION DE LA RÉCEPTION DURANT LES PHASES DE DÉVELOPPEMENT:**

##### **IV.2.5.3.1 .OBJECTIF:**

Décrire entièrement les procédures permettant de valider chacun des points de test prévus dans le C.R.G.

##### **IV.2.5.3.2 .CONTENU:**

Chaque point de test donne lieu à la rédaction d'un Cahier de Réception Détaillée. Ce document décrit la ou les procédures de validation associées. Doivent y figurer la description de chaque scénario de test, les jeux d'essai (Description des configurations, valeur des données sur lesquelles les tests seront appliquées), les résultats attendus (valeurs ou configurations produites). L'ensemble des C.R.D. est annexé au C.R.G.

#### **IV.2.5.4.VALIDATION:**

##### **IV.2.5.4.1 .OBJECTIF:**

Valider la fourniture par rapport aux exigences exprimées par la S.T.B.

##### **IV.2.5.4.2 .CONTENU:**

Les tests décrits dans les CRD sont effectués conformément aux directives contenues dans le CRG. Les résultats de chaque test sont consignés dans des comptes-rendus.

Chaque cas de non conformité avec les résultats attendus peut amener le client à notifier une «réserve» quant à l'acceptation de la réception. Dans tous les cas, l'acceptation définitive de l'objet est conditionnée à la levée de toutes les réserves en fin de qualification opérationnelle. Cependant, si une réserve est jugée «mineure» par le client, elle n'interdit pas de valider la réception de l'objet. Le réalisateur aura alors la possibilité de lever cette réserve pendant la qualification opérationnelle. En revanche, une réserve majeure doit être levée avant la fin de la phase de validation: elle implique donc un retour en phase de développement (pour correction de la non conformité), puis une reprise de la procédure de validation.

#### **IV.2.5.5.QUALIFICATION OPÉRATIONNELLE:**

##### **IV.2.5.5.1 .OBJECTIF:**

Valider la fourniture en utilisation courante.

##### **IV.2.5.5.2 .CONTENU:**

La fourniture est remise au client, à charge pour lui de l'utiliser dans son contexte normal pendant une durée définie. Les cas de non conformité par rapport au comportement attendu sont consignés dans des comptes-rendus. La qualification opérationnelle est conditionnée à la levée des réserves mineures émises lors de la validation et à la correction des cas de non conformités découverts en cours de qualification.

##### **IV.2.5.5.3 .RÉCEPTION DÉFINITIVE:**

En général, les contrats prévoient une durée maximale pour la phase de qualification. A l'issue de cette durée, et en l'absence de réserves non levées ou de non conformités découvertes en phase de qualification, la réception devient définitive. Sinon, des pénalités peuvent être appliquées au réalisateur.

## V.LES DÉMARCHES ITÉRATIVES - MÉTHODES AGILES:

### V.1.HISTORIQUE:

Les méthodes de gestion de projets par ITÉRATIONS successives (et non plus "EN CASCADE", comme dans le cycle en V ) datent du début des années 1980. Le concept de MÉTHODE AGILE, qui est rattaché à ce type de démarche date du début des années 2000. Ces techniques ont donc été largement confrontées aux réalités du développement de logiciel: elles peuvent donc être considérées comme matures.

### V.2.PRINCIPES FONDAMENTAUX DES MÉTHODES AGILES:

#### V.2.1.INTRODUCTION:

Les méthodes agiles découlent du «MANIFESTE AGILE» élaboré au début des années 2000. Ce manifeste met en avant quatre "PRINCIPES FONDAMENTAUX" que l'on peut résumer par les quatre assertions suivantes

1. Primauté de l'humain et de la communication dans le groupe par rapport à la mise en place de processus et d'outils de développement;
2. Primauté aux impératifs opérationnels par rapport aux exigences de qualité, de traçabilité et de documentation;
3. Primauté à une collaboration libre avec les clients par rapport au respect d'exigences contractuelle;
4. Primauté aux capacités d'adaptation au changement plutôt qu'au respect rigoureux d'une planification.

La première assertion a pour conséquence d'affaiblir le caractère impératif des exigences liées à la planification et au respect des démarches de développement: ces exigences ne constituent plus que des "cadres de travail" souples auxquels le groupe de réalisation est autorisé à déroger à chaque fois qu'il estime justifié de le faire.

La deuxième assertions établit que le respect des exigences en matière de qualité de réalisation, de traçabilité et de fourniture de documentation peuvent (et même doivent) être transgressées chaque fois que des impératifs opérationnels l'exigent (respect de délais, difficultés techniques, etc.). Le premier impératif est de LIVRER AU PLUS TÔT CE QUI PEUT ÊTRE LIVRÉ (c'est à dire ce qui fonctionne), même si cette livraison n'est pas accompagnée des documentation afférentes;

Des assertions 3 et 4 résulte la possibilité, pour le client de modifier ses exigences tout au long du projet, par une négociation libre avec le client. De là découle le caractère ITÉRATIF de la démarche agile: la cible est atteinte non pas par une démarche linéaire, mais par une succession d'itérations de la démarche de développement "convergeant" vers le produit final.

#### V.2.2.PREMIER PRINCIPE: DÉVELOPPEMENT ITÉRATIF:

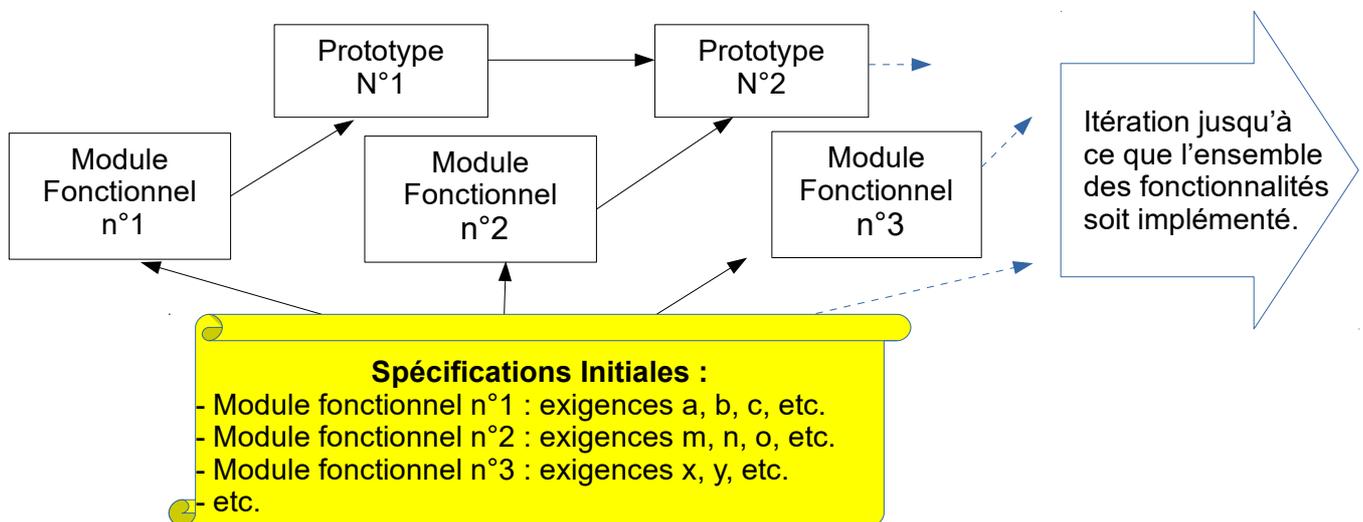
##### V.2.2.1.DESCRPTION:

1. A partir des exigences (fonctionnelles, opérationnelles, etc.) exprimées par le client, des SPÉCIFICATIONS sont élaborées en concertation avec celui-ci. Contrairement au cycle en V, ces exigences N'ONT PAS UN CARACTÈRE CONTRACTUEL:
2. A partir de ces spécifications, le développement se déroule sous la forme d'un certain nombre d'ITÉRATIONS. Chacune de ces itérations concerne le développement complet d'un sous-ensemble des exigences prévues par le document de spécifications initial. Nous appellerons cet ensemble d'exigences un MODULE FONCTIONNEL. Les itérations se

déroulent comme suit:

3. Un premier MODULE FONCTIONNEL, choisi par le client, fait l'objet d'un développement complet qui aboutit à la création d'un premier PROTOTYPE. Celui-ci fait l'objet d'une validation par le CLIENT. Si le client valide le prototype, celui-ci peut lui être livré, sinon, le prototype est renvoyé en phase de développement;
4. Le client choisit alors un deuxième MODULE FONCTIONNEL comme objectif de la deuxième itération. Les logiciels correspondant à ce module fonctionnel sont développés;
5. Le premier prototype est alors complété et modifié pour intégrer les fonctions spécifiées par le deuxième MODULE FONCTIONNEL. On aboutit ainsi à un deuxième prototype qui fait également l'objet d'une évaluation par le client. Si le client valide le nouveau prototype, celui-ci peut lui être livré, sinon, le prototype est renvoyé en phase de développement;
6. On choisit alors un troisième MODULE FONCTIONNEL qui conduira à un troisième prototype, et ainsi de suite, l'application se construit par itérations, chacune de ces itérations incrémentant dans le prototype de nouvelles fonctions.

Le schéma ci-après résume le principe des itérations:



#### REMARQUES:

- Les contenus et la priorité des modules fonctionnels à réaliser à chaque itération sont décidés par le représentant du CLIENT;
- Entre deux itérations le CLIENT peut modifier les SPÉCIFICATIONS du produit en accord avec le RÉALISATEUR. En revanche, pendant une itération, le sous-ensemble de fonctions qui constitue l'objectif de cette itération ne peut être modifié: il est «sanctuarisé»;
- Les itérations s'arrêtent dans deux cas:
  - L'ensemble des exigences fonctionnelles a été implémenté;
  - A un moment donné, le client se déclare satisfait du prototype tel qu'il est et abandonne ce qui reste à faire.

#### V.2.2.2.AVANTAGES:

Cette procédure a l'avantage de maintenir le client au plus près de la réalisation et de lui permettre d'affiner progressivement ses exigences en collaboration avec le développeur. Elle minimise donc les risques d'incompréhension ou d'imprécision dans l'expression des exigences.

**V.2.2.3.INCONVÉNIENTS:**

Les inconvénients sont de deux ordres:

- La procédure exige beaucoup d'engagement du client tout au long du projet en termes de mise à disposition de ses personnels;
- La procédure permet difficilement de prévoir la durée du projet: la plus ou moins grande disponibilité des personnels du client, mais aussi leur connaissance plus ou moins grande de l'environnement d'utilisation du produit interviendront beaucoup dans cette durée.

**V.2.3.DEUXIÈME PRINCIPE: LIVRER AU PLUS TÔT CE QUI PEUT L'ÊTRE:****V.2.3.1.PRÉSENTATION:**

Contrairement au cycle en V classique où l'ensemble du logiciel est dans la plupart des cas livré en une seule fois (sauf dans le cas d'un cycle incrémental), les méthodes agiles conseillent de livrer en plusieurs lots, chaque lot étant FONCTIONNEL, c'est à dire implémentant complètement une ou plusieurs des FONCTIONS du produit.

**Exemple:** Supposons que le produit final soit un site marchand supportant les fonctions suivantes:

1. Présentation du catalogue des produits;
2. Gestion du panier d'achat;
3. Gestion des comptes clients;
4. Gestion facturation et expédition;
5. Etc.

Il peut être prévu de livrer un premier lot supportant la fonction n°1 dès que le logiciel correspondant sera entièrement réalisé et validé, alors que les autres lots seront toujours en développement.

**V.2.3.2.AVANTAGES:**

Ces livraisons «dès que possible» permettent d'entretenir la relation client-fournisseur, de repérer le plus tôt possible les difficultés éventuelles et de former «en avance» les équipes d'utilisateurs.

**V.2.3.3.INCONVÉNIENTS:**

Développer séparément les différentes fonctions d'une application sans avoir au préalable étudié l'architecture générale du système pose fatalement de nombreux problèmes, dont les plus courants sont:

- Les incompatibilités et effets de bord entre modules fonctionnels (conflits de ressources);
- Les redondance: certains mécanismes communs risquent d'être développés plusieurs fois ou bien refondus et complétés à chaque introduction d'une nouvelle fonctions;
- Une architecture générale peu efficiente car résultant d'adaptations successives.

**V.2.4.AUTRES PRINCIPES LIÉS AUX MÉTHODES AGILES:**

Outre les deux principes exposés plus haut, la plupart des méthodes agiles (comme SCRUM ou XP) préconisent le respect d'un certain nombre de principes concernant la formation et l'animation des groupes chargés du développement. Elles concernent:

- **LA DURÉE DES ITÉRATIONS:** une itération doit être de COURTE DURÉE (1 à 4 semaines suivant les méthodes);

- **LA TAILLE DE L'ÉQUIPE DE RÉALISATION:** pour beaucoup de méthodes, l'équipe de réalisation doit être UNIQUE (pas de fragmentation des tâches entre plusieurs équipes) et l'effectif ne doit pas dépasser 8 ou 9 personnes;
- **LA HIÉRARCHIE AU SEIN DE L'ÉQUIPE DE RÉALISATION:** les membres de l'équipe de réalisation ne sont ni soumis à une hiérarchie ni spécialisés: chacun des membre est amené à intervenir sur l'ensemble du projet. Un responsable du groupe existe cependant, mais il intervient surtout comme animateur ou facilitateur et comme chargé de la communication avec le reste de l'entreprise.
- **LE TRAVAIL EN BINÔMES:** le travail en binôme est conseillé, mais les binômes doivent changer très fréquemment pour empêcher toute spécialisation;
- **LES RÉUNIONS:** outre les réunions marquant le début et la fin des itérations, des réunions JOURNALIÈRES (mais de très courte durée) sont conseillées pour contrôler l'avancement des travaux et détecter les difficultés éventuelles.

## V.3.MÉTHODES AGILES- DÉROULEMENT GÉNÉRAL:

### V.3.1.INITIALISATION DU PROJET ET PREMIÈRE ITÉRATION:

Les développements de logiciels menés suivant les principes de l'agilité débutent en général par une première itération qui concerne:

- L'étude du besoin initial aboutissant (souvent, mais pas toujours) à un document de SPÉCIFICATIONS non contractuel ;
- L'élaboration d'un PLAN DE RÉCEPTION indicatif et de CAHIERS DE RÉCEPTION;
- Une PLANIFICATION INITIALE. Cette planification est basée sur l'évaluation des charges de travail respectives liées au développement des fonctions à réaliser. La procédure est détaillée plus loin.

**REMARQUE:** des travaux de CONCEPTION PRÉLIMINAIRE destinés à élaborer des principes d'architecture globaux peuvent être effectués pendant cette première itération. Cependant, certains spécialistes considèrent qu'une telle démarche est inutile: pour eux, l'architecture globale doit émerger par retouches et adaptation successives des différentes itérations. Cette pratique est appelée REFACTORING (ou réusinage, remaniement, refactorisation).

Les travaux liés à cette phases initiales constituent une première itération. Après celle-ci, le processus de construction incrémentale des fonctions se déroule comme décrit plus haut.

### V.3.2.ITÉRATIONS SUIVANTES:

Chaque itération, qui en principe doit durer entre deux et quatre semaines, correspond en fait à un cycle complet de développement (au sens du cycle en V) comprenant:

- Une phase de **spécifications** (choix des fonctions à réaliser par l'itération, planning et estimation du volume de réalisation de chaque tâche parcellaire);
- Une phase de **Conception préliminaire** de l'incrément;
- Une phase de **Conception détaillée** de chacun des composants qui composent cet incrément (conception, codage, tests unitaires);
- Une phase d'**Intégration** des composants au prototype en cours (avec tests d'intégration);
- Une phase de **Réception** du nouveau prototype (test des fonctions ajoutées par l'itération et tests de non régression, validation par le client).

A la fin du développement, une RÉCEPTION GLOBALE est effectuée avant de réaliser la QUALIFICATION opérationnelle du produit.

### V.3.3.RECUEIL DES EXIGENCES DU CLIENT:

Le CLIENT exprime ses exigences (fonctionnelles, opérationnelles, méthodologiques, normatives, techniques, etc.) sous la forme de phrases simples, éventuellement accompagnées de critères de réalisation. Par exemple pour un site web marchand:

#### Exigence:

L'Administrateur du système doit pouvoir visualiser le nombre de visiteurs du site.

#### Critères:

- Nombre de visiteurs connectées;
- Nombre de premières connexion par heures, par journées et par semaines;
- Sous forme numérique et graphique.

Éventuellement, ces exigences sont analysées avec le concours du réalisateur. Elles peuvent être regroupées ou scindées suivant les besoins.

Dans certaines démarches agiles, cette activité n'aboutit pas à la rédaction d'un document formel de SPÉCIFICATIONS, mais à de simples FICHES.

#### **V.3.4.CONSTITUTION DES CAHIERS DE RÉCEPTION:**

Pour chaque exigence du client une PROCÉDURE DE TESTS DE RÉCEPTION est élaborée. Autant que possible, ces procédures doivent être automatisées sous la forme de logiciels de test, mais ce n'est pas toujours possible. La procédure peut alors s'appuyer sur:

- Pour les fonctions fournissant des résultats numériques ou alphanumériques: un TABLEAU fournissant pour chaque jeu de données d'entrée les données de sorties correspondantes (tableur excell ou autre).
- Pour les autres fonctions (IHM, sorties graphiques, sonores ou autres): des scripts peuvent parfois être utilisés (par exemple, pour simuler des interactions de l'utilisateur avec un interface graphique).

#### **V.3.5.NÉGOCIATION DES EXIGENCES:**

Avant chaque itération, client et réalisateurs peuvent NÉGOCIER les exigences contenues dans le recueil initial des spécifications (STB). Ceci concerne:

- L'abandon d'exigences par le client;
- La modification de la caractérisation de certaines exigences;
- L'introduction d'exigences supplémentaires, soit avec augmentation du prix, soit en compensant par l'abandon d'autre exigences (TROC des exigences).

#### **V.3.6.DÉMARCHE DE PLANIFICATION:**

##### **V.3.6.1.POSITION DU PROBLÈME:**

Dans le contexte des méthodes classiques, la planification initiale du projet est effectuée à la suite de l'activité de conception préliminaire. Or, dans le contexte des méthodes agiles, nous ne disposons pas, en général de résultats de conception préliminaire initiale (celle-ci n'est pas faite ou bien elle est forcément incomplète). En revanche, la liste des exigences exprimées par le client à ce stade du projet est connue, même si elle peut évoluer. La planification se fait alors à partir de cette liste.

##### **V.3.6.2.ÉTABLISSEMENT D'UN PLAN DE DÉVELOPPEMENT:**

Le plan est donc établi à partir de la liste des exigences du client et avec sa collaboration. La procédure est la suivante:

- Pour chaque exigence, un COÛT D'IMPLÉMENTATION est évalué en POINTS par les réalisateurs. La valeur du point n'a pas d'importance en soi: le tout est que le rapport entre les nombres de points attribués à deux exigences reflète le rapport entre les charges de travail qu'elles représentent;
- Une PRIORITÉ est également affectée à chaque exigence, en rapport avec leur importance vis à vis du client;
- Ces informations sont alors collationnées sous la forme d'un tableau dont chaque ligne correspondra à une exigence;
- Ce tableau sera exploité par le client et les réalisateurs pour déterminer et ajuster la planification du projet.

**V.3.6.2.1 .EXEMPLE SIMPLE DE PLAN DE DÉVELOPPEMENT:**

Cas d'un site web marchand:

EXIGENCE	COÛT D'IMPLÉMENTATION	PRIORITÉ
Les internautes doivent pouvoir visualiser la liste des articles à la vente.	25	1
Les internautes doivent pouvoir acheter un article de la liste visualisable	15	1
Les administrateurs doivent pouvoir ajouter ou retirer un article de la liste	20	2
Les internautes doivent pouvoir ouvrir et gérer un compte client	15	3
Les clients titulaires d'un compte doivent recevoir un newsletter.	12	3

## VI.SYNTÈSE SUR LES DEUX TYPES DÉMARCHES:

A première vue, la comparaison entre les démarches classiques "en cascade" (comme le cycle en V) et les démarches agiles peut sembler très déconcertante, tant les deux types de démarches semblent opposés, tant dans leurs principes que dans leurs traductions concrètes. En effet:

- Alors que les démarches classiques mettent l'accent sur la nécessité d'effectuer un recueil et une analyse du besoin exhaustifs afin de bien définir le lien contractuel entre client et réalisateur, les démarches agiles se contente d'une analyse du besoin sommaire et non contractuelle qui laisse ouverte les possibilité d'évolution de ce besoin;
- Alors que les démarches classiques insistent sur l'importance d'élaborer une conception préliminaire en préalable à toute conception de détail ou réalisation, les méthodes agiles font plus ou moins l'impasse sur toute étude préalable de l'architecture d'ensemble;
- Alors que les démarches classiques insistent sur la nécessité d'une planification détaillée, élaborée le plus tôt possible, les méthodes agiles se contente d'une planification souple et non contraignante dont le rôle est plus de mesurer les dérives éventuelles que de les empêcher.

En fait, les méthodes en cascade tendent à établir entre le client et le fournisseur une relation très contrainte par les engagements que les deux parties sont amenés à contracter et à finaliser par écrit dans le cahier des charge et les documents de réception. Une fois ces documents signés, il reste très peu de place pour une négociation sur d'éventuels ajouts ou aménagements. Cette rigidité est contraignante mais en contrepartie, elle fixe bien les rôles de chacun, assure une bonne traçabilité et permet de contrôler les dérives trop importantes en matière de coûts et de délais. Notons que le cycle en V incrémental permet de corriger en partie cette rigidité.

Les méthodes agiles, au contraire, privilégient la négociation client-fournisseur. Ce point, associé au caractère itératif des travaux, permet d'éviter ou de corriger les mauvaises spécifications de départ. L'inconvénient est la difficulté de contrôler les coûts et les délais, surtout dans le cas ou un partenaire est de mauvaise foi. Cet inconvénient les rend peu compatibles avec les exigences de la réglementation des marchés publics.

D'autre part, les méthodes agiles, du fait de l'organisation des équipes de développement et leur taille limitée, conviennent peu à des projets informatiques très volumineux, exigeant un découpage en plusieurs équipes, sur plusieurs sites et le respect d'un planning rigoureux.

## VII.ANNEXES:

### VII.1.CONTENUS ET PLANS DES DOCUMENTS (CAS D'UN CYCLE EN V):

#### VII.1.1.GÉNÉRALITÉS:

Dans le cadre d'une démarche de projet rationnelle, la documentation ne constitue pas une fourniture annexe destinée à commenter à posteriori la réalisation. Bien au contraire:

- Chaque document est un constituant essentiel de la réalisation au même titre que les composants matériels ou logiciels produits. L'ensemble des documents associés à un projet fait partie du référentiel qualité associé à ce projet. A ce titre, il doit faire l'objet d'une procédure d'acceptation par le client.
- La réalisation de la documentation associée à une fourniture matérielle ou logicielle doit **précéder** la réalisation de cette fourniture et non lui succéder. En effet, la documentation ne se contente pas de décrire la fourniture: elle expose et justifie les principes et contraintes que le réalisateur sera tenu de respecter.
- Chaque document fait l'objet d'une validation (en général sous forme de revue) **préalablement** à la réalisation de la fourniture qu'il décrit.

Chaque documentation est donc un élément de la traçabilité du projet. La traçabilité est une condition essentielle pour permettre la bonne utilisation de l'objet et rendre possible son évolution.

## VII.1.2.CAHIER DES CHARGES FONCTIONNEL (CDCF):

### VII.1.2.1.But du Document:

Le Cahier des Charges Fonctionnel rend compte de l'étude du besoin, du point de vue de l'utilisateur. Il énonce et caractérise les **fonctions externes** que le produit doit offrir dans toutes ses **phases de vie** (exemple: livraison, transport, utilisation, retrait, destruction, etc).

### VII.1.2.2.Plan et contenu:

#### I.OBJET:

##### I.1.Objet du document:

Cahier des Charges fonctionnel concernant le sous-projet X du projet y

##### I.2.Objet du projet:

Définition sommaire de la réalisation: description globale du besoin, origine de ce besoin.

#### DOCUMENTS APPLICABLES:

Titres et identification des documents auxquels il est fait référence dans le présent document.

#### DÉFINITION ET VALIDATION DU BESOIN:

##### 3.1.Validation du besoin du système d'ordre supérieur:

Expliquer dans quel contexte l'objet du projet doit s'intégrer (c'est le «système d'ordre supérieur»): à qui (ou à quoi) rend-t-il service ? dans quel but? quelles sont les causes qui pourraient faire disparaître ce besoin ?

Cette activité permet d'identifier les **ENTITÉS** de l'environnement qui sont en relation avec l'objet, c'est à dire celles qui **utilisent** l'objet ou bien **sont utilisées** par lui.

##### 3.2.Identification des fonctions principales:

###### 3.2.1.Identification des fonctions principales:

Les fonctions principales de l'objet sont celles qui lui permettent de mettre en relation deux entités de son environnement. On s'attachera à les exprimer par un verbe ou un groupe verbal à l'infinitif, suivi de deux compléments qui font référence à des entités de l'environnement.

###### **Exemple 1:**

*l'objet «lunette de vue» a pour entité d'environnement «le porteur» (de lunettes) et «la vision» (de ce porteur). Une fonction principale sera: **Améliorer la vision du porteur.***

###### **Exemple 2:**

*Dans la liste des entités d'environnement d'un objet «site internet de commerce en ligne», on peut trouver les entités «client» et «catalogue en ligne». Une fonction principale sera: **Permettre la consultation du catalogue en ligne par les clients.***

###### 3.2.2.Identification des fonctions contraintes:

Les fonctions contraintes sont celles qui mettent en relation une seule entité avec l'objet (l'autre entité étant en fait l'objet lui-même). Elles peuvent s'exprimer par un verbe à l'infinitif, suivi d'un seul complément qui fait référence à une entité de l'environnement.

###### **Exemple:**

*l'objet « automobile » peut avoir pour entité d'environnement le « niveau de carburant ». Une fonction contrainte*

*sera: **Signaler l'insuffisance du niveau de carburant***

Les fonctions contraintes sont ainsi appelées car elles correspondent à des besoins induits par l'objet lui-même et non par les utilisateurs. De ce fait, on les appelle également « fonctions de conception ».

**REMARQUES:**

- Dans le cadre du CDCF, les énoncés de fonctions doivent être effectués dans un « langage » accessible au client et aux utilisateurs
- Il faut étudier les fonctions de l'objet dans toutes les phases de la vie de celui-ci. Ces phases peuvent être: la production, le transport, l'utilisation, la maintenance, le retrait de service, etc. Seule la phase d'utilisation engendre des fonctions principales

**3.3. Décomposition des fonctions:**

Lorsqu'une fonction est très complexe, il peut s'avérer difficile de la caractériser directement, car les critères à définir serait trop nombreux ou trop hétérogènes. Dans ce cas, il peut être intéressant de décomposer cette fonction en sous-fonctions.

**Exemple:** La fonction principale de l'objet « lunette de vue »: **Améliorer la vision du porteur** pourrait se décomposer en deux sous-fonctions: **Améliorer la vision de près du porteur et Améliorer la vision de loin du porteur.**

**Attention:** lorsqu'on décompose une fonction, il faut s'assurer que les sous-fonctions obtenues sont indépendantes les unes des autres, c'est à dire que chacune de ces fonctions peut exister indépendamment des autres.

**3.4. Caractérisation des fonctions:**

La caractérisation permet de préciser les critères de rendu du service, et pour chaque critère, les performances attendues et les tolérances acceptables (limite ou fourchette d'acceptation du critère). On peut également joindre une brève justification du besoin. Dans le cadre du CDCF, cette caractérisation doit être effectuée dans un « langage » accessible au client. Le tableau suivant donne un exemple de présentation de cette étude:

<b>FONCTION:</b>		<b>Permettre la consultation du catalogue en ligne par les clients.</b>		
<b>CRITÈRE</b>	<b>NIVEAU ATTENDU</b>	<b>TOLÉRANCE</b>	<b>JUSTIFICATION</b>	
Présentation d'une photographie de chaque produit.	Les détails >= 1/100e de la dimension maximale du produit doivent pouvoir être vus par un client dont l'acuité visuelle (après correction) est au moins égale à 4/10e	Poste client muni d'une définition d'écran >= 1024x768	Le client doit avoir la même impression visuelle que s'il observait un produit dans un rayon de magasin.	

Les critères ainsi définies seront à la base de la définition des tests de réception de l'objet par le client. Il est donc indispensable de définir des critères **testables**.

## VII.1.3.SPÉCIFICATION TECHNIQUE DE BESOIN ET DOSSIER JUSTIFICATIF DES EXIGENCES:

### VII.1.3.1.But du document:

La Spécification Technique de Besoin (S.T.B) constitue le document de référence pour tout ce qui concerne le développement et les modalités générales de réception de l'objet.

Comme le C.D.C.F, elle décrit et caractérise les exigences du client vis à vis du produit. Cependant, elle approfondit ces spécifications en les traduisant en exigences et en critères techniques (informatique) qui les rendent immédiatement opérationnelles pour le réalisateur.

De plus, la S.T.B. décrit l'environnement d'accueil de l'objet et les conditions de son intégration dans cet environnement. En particulier, elle définit et caractérise en termes techniques les interfaces de l'objet avec son environnement.

Enfin, la S.T.B fixe les conditions générales de la réception de l'objet.

### VII.1.3.2.Plan et contenu du document:

#### I.OBJET:

##### I.1.Objet du document:

Spécifications Techniques de Besoin concernant le sous-projet X du projet y

##### I.2.Objet du projet:

Définition sommaire de la réalisation: description sommaire du besoin, origine de ce besoin (recopie du CDCF ou référence à celui-ci)

#### II.DOCUMENTS APPLICABLES:

Titres et identification des documents auxquels il est fait référence dans le présent document (en particulier, le CDCF)

#### III.EXIGENCES:

les exigences identifiées et caractérisées par le C.D.C.F sont ici reclassées en 3 catégories:

##### 3.1.Exigences fonctionnelles:

##### 3.2.Exigences opérationnelles:

##### 3.3.Exigences de conception et de réalisation:

Ces exigences et les critères de caractérisation qui leur ont été associés sont reprises du CDCF et reformulées en termes techniques. Par exemple, la caractérisation de l'exigence du CDCF: «Permettre la consultation du catalogue en ligne par les clients» pourrait être reformulée de la manière suivante:

<b>FONCTION:</b>	<b>Permettre la consultation du catalogue en ligne par les clients.</b>		
<b>CRITÈRE</b>	<b>NIVEAU ATTENDU</b>	<b>TOLÉRANCE</b>	<b>JUSTIFICATION</b>
La présentation de chaque produit sur l'écran client doit inclure une vue photographique associée au texte.	Le 1/100e de la dimension maximale du produit doit pouvoir occuper au moins 5 pixels sur l'écran du client	- Poste client muni d'une définition d'écran >= 1024x768 - Pour satisfaire le critère, l'utilisation d'une fonction «zoom» associée à chaque photographie est acceptée.	Le client doit avoir la même impression visuelle que s'il observait un produit dans un rayon de magasin.

#### IV.INTERFACES:

Décrire chaque interface du système avec son environnement:

- Fonctionnalités
- Performances
- Protocoles de communication
- Structure des messages

- Structure et enchaînement des menus de commande et de paramétrage
- etc.

#### **V.RÉCEPTION:**

Décrire les modalités générales de la réception (plan de réception):

- Procédure générale de réception
- Liste des fournitures devant faire l'objet d'une réception (fournitures dûes par le client et dûes par le réalisateur)
- Planification globale des travaux de réception (préparation et réalisation).

#### **VI.ANNESES:**

- Descriptions fines des algorithmes et des calculs mathématiques que le client veut utiliser
- Modélisation et maquetages en vue de valider la S.T.B. (en particulier, maquetages des IHM)
- etc.

## VII.1.4.MANUEL OPÉRATEUR:

### VII.1.4.1.But du document:

Le Manuel Opérateur doit permettre l'utilisation de l'application par les utilisateurs prévus par le client.

### VII.1.4.2.Contenu:

Le manuel opérateur doit décrire les procédures d'utilisation du produit dans phases de vie (installation, configuration, mise en service, utilisation opérationnelle, etc) et, dans chaque phase, pour chaque mode d'utilisation du produit (Par exemple dans sa phase de vie «utilisation opérationnelle» un produit peut inclure les modes d'utilisation «préparation-paramétrage», «exécution», «maintenance», etc.)

Pour décrire les procédures d'utilisation, le manuel opérateur reprend les informations concernant les interfaces homme-machine contenues dans la S.T.B en les détaillant et les approfondissant. Il doit tenir compte de la formation des utilisateurs prévus par le client: le manuel opérateur doit être rédigé d'une manière compréhensible par eux.

### I.OBJET:

Manuel opérateur du produit X, version Y ....

### II.DOCUMENTS APPLICABLES:

Titres et identification des documents auxquels il est fait référence dans le présent document (en particulier, la S.T.B.)

### III.DESCRPTION DU PRODUIT:

#### 3.1.II. GÉNÉRALITÉS:

Description du produit, de son environnement et de ses missions.

#### 3.2.INTERFACES D'EXPLOITATION:

Décrire les interfaces d'exploitation (menus et enchaînements de menus, synoptiques de surveillance, pupitres de commande): aspect visuel et fonctionnalités.

### IV.UTILISATION:

Pour chacune des phases de vie du produit, on décrit ses modes d'utilisation. Par exemple, si le produit est un logiciel interactif:

#### 4.1.INSTALLATION:

##### 4.1.1.Procédure d'installation sous windows:

Décrire la procédure d'installation sous windows...

##### 4.1.2.Procédure d'installation sous linux:

Décrire la procédure d'installation sous linux...

#### 4.2.CONFIGURATION:

##### 4.2.1.Choix des polices de caractères:

Décrire les procédures permettant de choisir les polices de caractères d'installation sous linux...

##### 4.2.2.Choix des couleurs:

#### 4.3.UTILISATION OPÉRATIONNELLE:

##### 4.3.1.Préparation d'une utilisation:

Décrire les procédures permettant de paramétrer l'utilisation de l'application...

**4.3.1.Utilisation en mode xxxx:**

.....

**4.4.DÉSINSTALLATION:**

.....

.....

**V.ANNEXES:**

## VII.1.5.ORGANIGRAMME DES TACHES (OT):

### VII.1.5.1.But du document:

Déterminer et visualiser graphiquement la liste des tâches nécessaires à la réalisation du projet.

### VII.1.5.2.Démarche de construction:

La démarche est basée sur une décomposition arborescente des activités. L'organigramme des tâches est un document évolutif qui s'enrichit et s'affine tout le long du projet. La démarche présentée ci-après est donnée uniquement pour exemple. Elle a le mérite d'être la plus générale possible.

#### VII.1.5.2.1 .Premiers niveaux de décomposition:

En première analyse, on peut décomposer les activités en 3 catégories:

- Les activités directement liées à la réalisation du projet.
- Les activités de direction du projet.
- Les activités de soutien au projet.

Les activités directement liées à la réalisation sont celles qui composent le cycle en V de développement du produit. Nous avons largement détaillé leur décomposition dans cet ouvrage. Dans un premier temps, nous les avons décomposées en 3 sous-catégories:

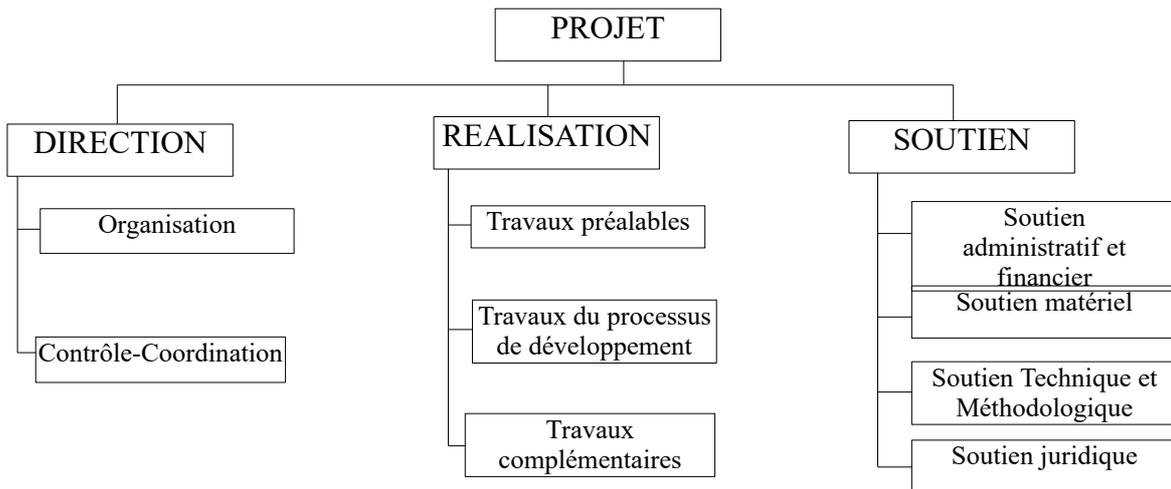
- Travaux préalables
- Travaux du processus de développement
- Travaux complémentaires

Les activités de direction du projet peuvent être décomposées en 2 sous-catégories:

- Les activités d'organisation (organisation des équipes, nomination des responsables, attribution des tâches)
- Les activités de contrôle et de coordination: contrôle d'avancement, contrôle d'utilisation des ressources, contrôle de qualité, coordination, arbitrages.

Les activités de soutien sont celles qui permettent de fournir aux autres activités les ressources qui leur sont nécessaires. Il s'agit de se procurer ces ressources et de les gérer au mieux. Les activités de soutien peuvent dans un premier temps être décomposées en 4 sous-catégories.

- Soutien administratif et financier: embauches de collaborateurs, gestion des crédits, emprunts, achats, etc...
- Soutien matériel: mettre à disposition des réalisateurs les moyens matériels nécessaires à leur mission: locaux, plates-formes de développement, matériels divers, etc.
- Soutien technique et méthodologique: prestations de conseil, d'assistance technique, de formation, etc.
- Soutien juridique: conseils pour la rédaction des contrats, pour la gestion des contentieux, etc.



#### VII.1.5.2.2 .Suite de la démarche:

Cette première décomposition peut s'appliquer à tous les projets. Après ce niveau, l'organigramme va se particulariser de plus en plus en fonction de la taille du projet, de sa nature, de sa complexité, etc.

#### EXEMPLES:

- Les travaux du processus de développement d'un projet complexe pourront se décomposer en un projet principal et plusieurs sous-projets
- Le soutien technique et méthodologique pourra comprendre la gestion de stages de formation, l'achat de documentations, de services de consultants, etc..

#### VII.1.5.2.3 .REMARQUE:

Nous pouvons constater que la représentation graphique sous forme d'organigramme est très difficile à gérer sans outil. Si l'on ne dispose pas de ceux-ci, on peut se contenter d'une description sous la forme d'un texte indenté, d'un maniement beaucoup plus aisé:

```

PROJET
I.DIRECTION
  1.1.Organisation
    1.1.1.Equipes de projets
    1.1.2.Planification des activités
  1.2.Contrôle-coordination
    1.2.1.Contrôle d'avancement
    1.2.2.Coordination et arbitrage.
    1.2.3.Contrôle qualité.
    1.2.4.Contrôle finances et ressources.
II.RÉALISATION
  2.1.Travaux préalables
    2.1.1.Expression du besoin
    2.1.2.Contract
    2.1.3.Planification réalisation
  2.2.Travaux du processus de développement
    2.2.1.Conception préliminaire
    2.2.2.Conception détaillée projet principal
    2.2.3.Sous projet 1
    2.2.4.Sous projet 2
    2.2.5.Intégration projet principal
    2.2.6.Intégration sous-projet 1
    2.2.5.Intégration sous-projet 2
  2.3.Travaux complémentaires
    
```

- 2.3.1.Réception
- 2.3.2.Qualification

### III.SOUTIEN

- 3.1.Soutien administratif et financier
  - 3.1.1.Embauche collaborateurs
  - 3.1.1.Gestion des crédits
  - 3.1.1.Achats de matériels
- 3.2.Soutien materiel
  - 3.2.1.Création et gestion de la plate-forme de développement
  - 3.2.2.Maintenance des matériels et systèmes d'exploitation
- 3.3.Soutien technique et méthodologique
  - 3.3.1.Assistance qualité
  - 3.3.1.Organisation des formations
- 3.4.Soutien juridique
  - 3.4.1.Conseil pour rédaction du cahier des charges.

Le degré de décomposition optimal dépend en grande partie de la complexité du projet et des effectifs en personnels disponibles. Il est, d'autre part, indispensable de définir des tâches cohérentes, centrées sur un type d'activité bien déterminé et exigeant des compétences et des ressources bien identifiables. Enfin, il faut se souvenir qu'une trop grande parcellisation des tâches peut être une source de démotivation pour ceux qui en sont chargés tout en compliquant beaucoup la coordination et le contrôle.

## VII.1.6.PLAN DE DÉVELOPPEMENT (PDV):

### a)-But du document:

Bien définir chaque tâche et construire un calendrier prévisionnel (par exemple, sous forme de PERT)

### b)- Contenu:

#### I.OBJET:

Plan de développement du sous-projet x du projet y...

#### II.DOCUMENTS APPLICABLES:

Titres et identification des documents auxquels il est fait référence dans le présent document (en particulier D.C.P. et de référence).

#### III.TERMINOLOGIE ET SIGLES UTILISES:

#### IV.ORGANIGRAMME DES TACHES:

L'organigramme des tâches peut être inclus dans ce document.

#### V.DESCRPTION DES TACHES:

le Plan de Développement reprend les tâches définies dans l'O.T. et définit pour chacune d'elles:

- Le responsable de cette tâche.
- Le contenu de la tâche (ressources consommées, ressources produites, description des activités)
- L'estimation de sa durée (s'il ne s'agit pas d'une activité permanente, du type contrôle ou soutien)
- L'identification qualitative et quantitative des ressources nécessaires (humaines, financières, techniques, matérielles, etc).
- Les contraintes associées (Disponibilité des ressources, confidentialité, etc ...)

#### VI.PLANIFICATION DU PROJET:

Une première planification des tâches est réalisée à partir de ces données. Cette planification

sera constamment actualisée en cours de projet pour tenir compte d'une part de l'affinage de certains paramètres (par exemple, l'estimation de la durée d'une tâche peut être affinée au cours de sa conception détaillée), et d'autre part des aléas éventuels.

## VII.1.7.DOCUMENTS DE CONCEPTION PRÉLIMINAIRE:

### VII.1.7.1.But du document:

Exposer et justifier l'architecture globale de l'application.

### VII.1.7.2.Plan et contenu:

#### I.OBJET:

Document de conception préliminaire du sous-projet x du projet y...

#### II.DOCUMENTS APPLICABLES:

Titres et identification des documents auxquels il est fait référence dans le présent document (en particulier, la S.T.B. de référence).

#### III.TERMINOLOGIE ET SIGLES UTILISES:

#### IV.CONTRAINTES ET PERFORMANCES A RESPECTER ET CHOIX DE CONCEPTION RESULTANTS:

Exposer les contraintes majeures qui ont des conséquences sur les choix de conception de l'objet. (ces contraintes sont déduites des exigences et des critères associés). Exposer les choix de conception résultants.

#### V.ANALYSE DU PROBLÈME:

Dans la forme, la rédaction de ce chapitre dépend beaucoup de la méthode de conception utilisée. Cependant, quelle que soit cette méthode, la première étape consiste toujours en une identification des **traitements** nécessaires à la réalisation des exigences, des **données rémanentes** concernées par ces traitements et des **interactions** entre ces traitements.

#### VI.DESCRPTION DE LA SOLUTION:

La solution architecturale doit être décrite du point de vue dynamique (fonctionnement) et du point de vue statique (structure).

Chaque composant identifié doit recevoir une description suffisante pour permettre sa réalisation. En particulier, les interfaces du composant avec le reste du système doivent être entièrement décrit:

- Forme et contenu des messages échangés.
- Protocoles d'échange.
- Traitements activés à partir de ces interfaces.

#### VII.VALIDATION DE LA SOLUTION:

##### 7.1.Recouvrement des fonctions par les composants:

Ce sous-chapitre rend compte du recouvrement des exigences par l'architecture choisie. Ceci revient à vérifier que les activités encapsulées dans les différents composants permettent bien de satisfaire l'ensemble des exigences de la S.T.B. (on pourra représenter cette étude sous la forme d'un tableau Exigence x Composant:

	Exigence n° 1	Exigence n° 2	Exigence n° 3		Exigence n° n
Composant n° 1	Traitement t1.i	Traitement t1.j	Traitement t1.k	----->	Traitement t1.x
Composant n° 2	Traitement t2.i	Traitement t2.j	Traitement t2.k		-----
-----	-----	-----	-----		Traitement tn.x
Composant n° n	Traitement tn.i	Traitement tn.j	Traitement tn.k		

(On vérifie par exemple, que les traitements Traitement t1.j, Traitement t2.j, ... , Traitement tn.j permettent bien de satisfaire l'exigence n° 1).

### 6.2. Validation du fonctionnement dynamique:

Ce sous-chapitre rend compte de la vérification du fonctionnement dynamique de l'architecture choisie (on utilise en général un maquettage ou une modélisation partielle ou complète sous forme d'automates, de réseaux de Pétri, etc...).

## VII.1.8. PLAN D'INTÉGRATION:

### VII.1.8.1. But du document:

Exposer la procédure d'intégration des divers composants (planification de l'intégration) et les tests de validation associés à chaque étape.

### VII.1.8.2. Plan et contenu:

#### I. OBJET:

plan d'intégration du sous-projet x du projet y...

#### II. DOCUMENTS APPLICABLES:

Titres et identification des documents auxquels il est fait référence dans le présent document (en particulier, la D.C.P. de référence).

#### III. TERMINOLOGIE ET SIGLES UTILISES:

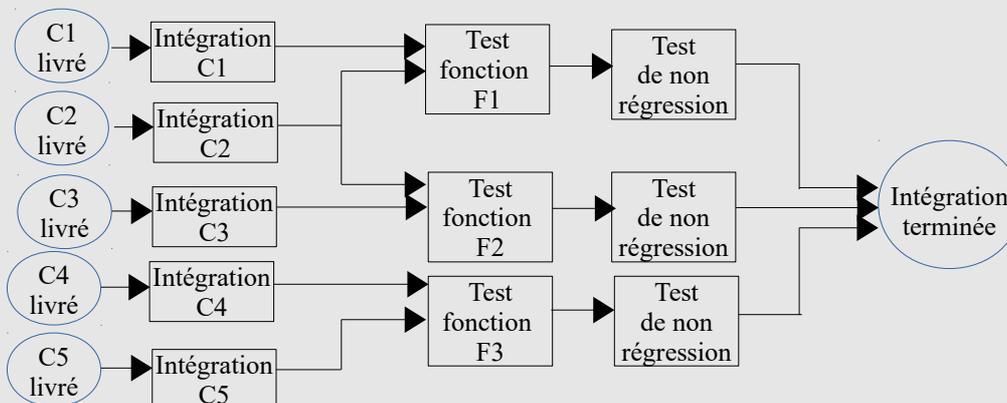
#### IV. DESCRIPTION DE LA PROCÉDURE D'INTÉGRATION:

Une procédure d'intégration assez pratique consiste à réaliser (en fin de conception préliminaire) une maquette générale de l'architecture où tous les composants sont «émulés» (c'est à dire: remplacés par un composant réduit à leurs interfaces externes, le fonctionnement interne étant grossièrement simulé). Cette maquette peut également servir à la validation de la solution architecturale.

Chaque fois qu'un composant est livré, il est implanté dans l'architecture à la place de son émulateur. Chaque fois que l'ensemble des composants réalisant une fonction donnée sont intégrés, on exécute les tests de réception prévus par le Cahier de Réception Général pour valider cette fonction. On revalide également les fonctions précédemment intégrées (tests de non régression). A la fin du processus, l'intégration est entièrement validée.

#### V. PLANIFICATION GÉNÉRALE DE L'INTÉGRATION:

Décrit l'ordonnancement général prévu pour l'intégration des lots. Dans le cas de la procédure proposée au paragraphe IV (intégration fonction par fonction), le plan d'intégration pourra ressembler à ce qui suit:



NOTA:

*Le planning d'intégration ci-dessus traite de l'intégration des lots C1, C2, C3, C4 et C5. La fonction F1 est remplie par C1 et C2, la fonction F2 par C2 et C3, la fonction F3 par C4 et C5. Les tests sont déclenchés chaque fois que tous les composants concourant à une fonction donnée sont intégrés. Ces tests sont ceux prévus dans le cahier de réception général.*

## VII.1.9.CAHIER DE RÉCEPTION GÉNÉRAL:

### VII.1.9.1.But du document:

Décrire les modalités générales de la réception (validation et qualification opérationnelle), définir la liste des tests de validation et le principe de chacun de ces tests, planifier la validation.

### VII.1.9.2.Contenu:

#### I.OBJET:

Cahier de réception général du sous-projet x du projet y...

#### II.DOCUMENTS APPLICABLES:

Titres et identification des documents auxquels il est fait référence dans le présent document (en particulier, la S.T.B. de référence).

#### III.TERMINOLOGIE ET SIGLES UTILISES:

#### IV.MODALITÉS GÉNÉRALES DE LA RÉCEPTION:

- Décrire les principales étapes de la procédure de préparation et d'exécution de la réception et définir les responsabilités et les acteurs.
- Énumérer les fournitures devant faire l'objet d'une validation

**REMARQUE:** ces renseignements devraient figurer dans la S.T.B. Il peut suffire d'y faire référence.

#### V.DESCRPTION DES TESTS DE RÉCEPTION:

Pour chaque fourniture, définir les tests de validation à effectuer et les principes de réalisation de ces tests, définir les conditions de leur réalisation et les ressources nécessaires

#### VI.PLANIFICATION:

Décrire l'enchaînement logique des tests de réception.

## VII.1.10.DOSSIERS DE DÉFINITION D'UN COMPOSANT:

### VII.1.10.1.But du document:

Définir la structure interne d'un composant, exposer et justifier les principes de sa conception interne et de sa réalisation. Le document doit permettre la réalisation immédiate du composant, sans étude supplémentaire. Il doit également être suffisant pour permettre la maintenance et l'évolution du composant.

### VII.1.10.2.Contenu.

#### **I.OBJET:**

Dossier de définition du composant C du sous-projet x du projet y...

#### **II.DOCUMENTS APPLICABLES:**

Titres et identification des documents auxquels il est fait référence dans le présent document (en particulier, la D.C.P. de référence).

#### **III.TERMINOLOGIE ET SIGLES UTILISES:**

#### **IV.EXPOSE DES CONTRAINTES ET CONSEQUENCES SUR LA CONCEPTION:**

Exposer les contraintes de réalisation particulières, résultant des exigences fonctionnelles, opérationnelles, de conception et de réalisation, etc.

#### **V.DESCRPTION ARCHITECTURALE DU COMPOSANT:**

Description de l'architecture interne (sous-composants et relations entre sous-composant)

- Du point de vue statique (structuration): la forme de cette description dépend de la méthode de développement adoptée.
- Du point de vue dynamique (fonctionnement): décrire les différents modes de fonctionnement, les traitements correspondants (algorithmes, ordonnancement, etc.)

#### **VI.INTERFACES:**

Décrire les interfaces externes du composant (points d'activation, forme des appels, protocoles de communication, traitements associés).

## VII.1.11.DOSSIER DE TEST UNITAIRE D'UN COMPOSANT:

### VII.1.11.1.But du document:

Définir entièrement les modalités d'exécution des tests unitaires d'un composant..

### VII.1.11.2.Contenu:

#### I.OBJET:

Dossier de tests unitaires du composant xxx du sous-projet x du projet y...

#### II.DOCUMENTS APPLICABLES:

Titres et identification des documents auxquels il est fait référence dans le présent document (en particulier, le D.C.P. du projet et le D.D. du composant).

#### III.PRÉSENTATION GÉNÉRALE DES TESTS:

Décrire et justifier l'organisation et l'enchaînement des opérations de test.

#### II.DESCRPTION DES TESTS:

Les tests doivent avoir pour but de valider le composant par rapport aux spécifications contenues dans le Dossier de Conception Préliminaire. Il faut s'assurer que le comportement du composant, soumis aux différentes sollicitations et contraintes prévues par le D.C.P, réagit bien comme ce document le prévoit (traitements effectués, performances, robustesse, résistance aux erreurs, etc.

Pour chaque test, décrire:

- Le dispositif de test et les moyens nécessaires.
- La procédure de préparation et d'exécution du test
- Les jeux d'essais
- Les résultats attendus.

## VII.1.12.COMPTES-RENDUS DE TESTS UNITAIRES:

Ces comptes rendus sont des fiches, permettant aux responsables de noter, pour chaque test unitaire, les résultats du test et les écarts par rapport aux résultats attendus. Ces fiches sont annexées à chaque dossier de test unitaire.

## **VII.1.13.CAHIERS DE RÉCEPTION DÉTAILLÉS:**

### ***VII.1.13.1.But du document:***

Pour chaque point de test défini dans le C.R.G, définir complètement les modalités de réalisation des tests associés.

### ***VII.1.13.2.Contenu:***

#### **I.OBJET:**

Cahier de Réception Détaillé du point de test n° xxx du sous-projet x du projet y...

#### **II.DOCUMENTS APPLICABLES:**

Titres et identification des documents auxquels il est fait référence dans le présent document (en particulier, le C.R.G. de référence).

#### **III.PRÉSENTATION GÉNÉRALE DES TESTS:**

Décrire et justifier l'organisation et l'enchaînement des opérations de test.

#### **II.DESCRPTION DES TESTS:**

Pour chaque test, décrire:

- Le dispositif de test et les moyens nécessaires.
- La procédure de préparation et d'exécution du test
- Les jeux d'essais
- Les résultats attendus.

### ***VII.1.13.3.COMPTES-RENDUS DE TESTS DE RÉCEPTION:***

Ces comptes rendus sont des fiches, permettant aux responsables de la réception de noter, pour chaque point de test soumis à validation, leurs conclusions et leurs réserves, à l'issue de chaque test de recette. Ces fiches sont annexées à chaque C.R.D.

## VII.2.PRINCIPES DE DÉCOMPOSITION ORGANIQUE:

En phase de conception préliminaire, la problématique principale attachée à l'élaboration d'une architecture adaptée aux exigences est la recherche d'un découpage en «composants de premier niveau» pertinent.

Nous avons vu plus haut que, sauf dans le cas d'une démarche purement fonctionnelle, la démarche consistait toujours à essayer d'encapsuler les traitements et les données rémanentes dans des composants, d'identifier les interactions entre ces composants, puis d'évaluer l'architecture obtenue par rapport aux exigences.

La démarche d'encapsulation doit être guidée en priorité par le soucis d'obtenir les qualités que l'on attend du produit. D'autres facteurs peuvent également intervenir, comme les contraintes dues à la réglementation, l'état de l'art, la politique du réalisateur en matière de méthodologie, de normes, de réutilisation des logiciels, etc.

Le tableau ci-après énumère les qualités que l'on peut attendre d'un produit informatique et les conséquences sur le choix des composants. Précisons qu'une qualité ne s'applique pas forcément à tout le système: elle peut ne concerner qu'un groupe de composants, impliqués dans un traitement donné.

Nous pourrions remarquer, à la lecture de ce tableau, que très souvent, la recherche d'une qualité amène des choix de conception incompatibles avec d'autres qualités. Par exemple, la recherche de la rapidité ou de la réactivité est peu compatible avec la lisibilité ou la maintenabilité. De ce fait, une solution architecturale repose toujours sur le choix des qualités à privilégier. Ce choix est, bien sûr, guidé en priorité par les exigences du client. Cependant, d'une part, ces qualités peuvent présenter des incompatibilités, d'autre part, il est la plupart du temps impossible de négliger complètement les autres qualités, sous peine de rencontrer de grosses difficultés lors du développement ou de la réception. Il faut donc trouver un compromis acceptable.

### **REMARQUES:**

- *De nombreuses méthodes de conception architecturale des logiciels sont disponibles (fonctionnelles, par les données, par objets, etc...). Il est très recommandé de les utiliser dans cette phase. Cependant, ces méthodes sont peu applicables au développement de composants matériels. De ce fait, lorsqu'un projet comporte le développement d'éléments matériels, ceux-ci font l'objet de sous-projets gérés suivant des méthodes adaptées.*
- *Une «politique» de réutilisation de composants consiste à mettre en place des pratiques favorisant, d'une part la production de composants réutilisables à l'occasion de projets et d'autre part la réutilisation de ces composants lors de projets ultérieurs. Ceci amène souvent à réaliser des composants plus élaborés que ce qui est strictement nécessaire pour le projet en cours. En revanche, la réutilisation de ce composant dans d'autres projets s'en trouve facilitée. La réutilisation, si elle est bien gérée, peut amener une réduction très importante des coûts et des délais.*

<b>QUALITÉS D'UN PRODUIT INFORMATIQUE</b>		
<b>QUALITÉ</b>	<b>OBJECTIF</b>	<b>CONSÉQUENCES SUR LA CONCEPTION</b>
Rapidité d'exécution	Privilégier la rapidité d'exécution des traitements d'une fonction donnée.	Les composants concourant à la réalisation de l'exigence doivent être définis de façon à minimiser leur nombre et à optimiser la circulation des données entre eux. En particulier, on minimisera le nombre de couches logicielles traversées
Réactivité	La réponse à un événement externe donnée doit intervenir le plus rapidement possible.	A la contrainte de rapidité d'exécution du traitement (interaction-réponse) doit être ajoutée la contrainte de rapidité de la prise en compte de l'événement lui-même. Le traitement de l'événement doit être déclenché par un gestionnaire d'événement (programmation événementielle). Le choix d'un système d'exploitation multitâches à priorités préemptives peut se révéler nécessaire.
Déterminisme temporel	La réponse à un événement externe donnée doit intervenir dans un intervalle de temps déterminé.	Le déterminisme temporel implique la réactivité pour la prise en compte des événements. Elle n'implique pas forcément la rapidité, qui dépend du temps de réponse minimal.
Lisibilité	Les principes architecturaux doivent être facilement compréhensibles.	Limiter le nombre de niveaux de décomposition et de composants par niveau. Les bonnes proportions pour une architecture statique sont: <ul style="list-style-type: none"> <li>• De 3 à 5 couches de composants</li> <li>• Moins d'une dizaine de composants par niveau pour les couches intermédiaires</li> <li>• Les couches supérieures et inférieures doivent être moins remplies que les couches intermédiaires</li> </ul>
Fiabilité	Le fonctionnement global doit pouvoir résister à la défaillance d'un composant matériel ou logiciel.	<ul style="list-style-type: none"> <li>• Éviter de lier le fonctionnement d'un composant à celui d'un autre composant: en particulier, éviter le partage de données</li> <li>• Préférer les systèmes «répartis» aux systèmes centralisés.</li> <li>• Prévoir des redondances pour les composants jouant un rôle «critique».</li> <li>• Utiliser des composants à la fiabilité éprouvée (composants standards, composants réutilisés, etc)</li> </ul>
Ré utilisabilité	L'idée est de développer certains composants de façon à pouvoir les réutiliser pour d'autres projets	Les composants concernés doivent faire l'objet d'études supplémentaires: ils doivent être évolutifs et adaptables à d'autres environnements. Ceci implique que ces composants puissent supporter des fonctionnalités non exigées par le client, ce qui augmente leur complexité et leur coût de réalisation.
Réutilisation	L'idée est de réutiliser des composants développés à l'occasion d'autres projets	La décomposition architecturale (statique ou dynamique) doit être menée de façon à faire apparaître ces opportunités de réutilisation. Ceci peut conduire à des architectures moins optimisées en taille et en rapidité. En revanche, la réutilisation permet de faire baisser les coûts et délais de réalisation.
Maintenabilité	Lors de défaillances du système, les composants en cause doivent pouvoir être repérés rapidement.	Il faut éviter de distribuer les traitements correspondant à une fonction dans trop de composants et minimiser les interactions entre composants. Ceci implique souvent une augmentation du nombre de composants et de couches logicielles.

**QUALITÉS D'UN PRODUIT INFORMATIQUE**

Évolutivité	<ul style="list-style-type: none"> <li>• L'ajout de fonctions nouvelles ou la modification de fonctions existantes doit demander un minimum de travail d'intégration.</li> <li>• Les risques d'effets de bord avec le reste du système doivent être minimisés.</li> </ul>	<ul style="list-style-type: none"> <li>• Respecter une structuration des composants en plusieurs couches.</li> <li>• Encapsuler données et traitements d'accès à ces données dans le même composant (conception objet)</li> </ul>
-------------	---	---