La démarche Qualité

IA Corporation est résolument engagé dans le développement de la “Qualité”. Une cellule Qualité, composée de sept personnes, a pour mission de formaliser, de former et de contrôler la Charte de la Qualité et les Plans Qualité de chaque projet.

La Charte de la Qualité est décrite au moyen de trois manuels de référence qui concentrent 10 années d’expertise et de savoir-faire accumulés par IA Corporation.

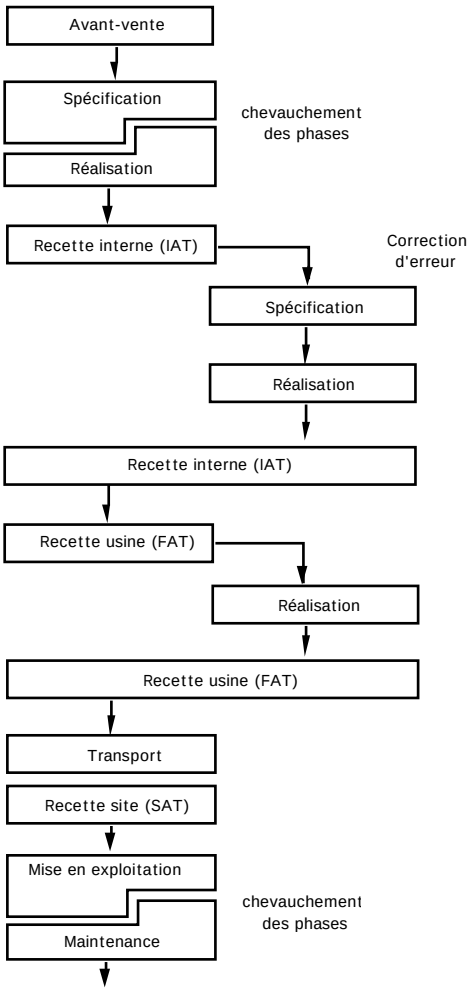
◊ “*Plan de Développement Système*”
Ce manuel présente le plan de développement d’un système à IA Corporation et décrit le cycle de vie ainsi que les productions par phase du cycle de vie d’un projet depuis l’avant-vente jusqu’à la maintenance, au suivi et extensions.

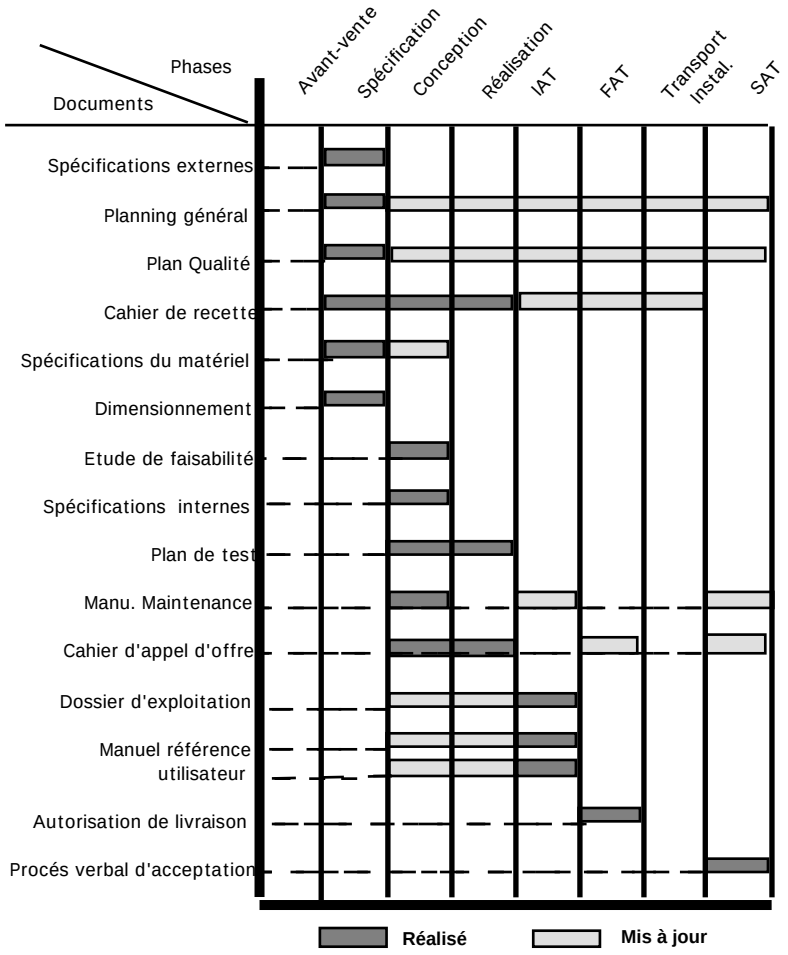
◊ *“Manuel d’Assurance Qualité Système*”
Ce manuel s’appuie sur le *Plan de Développement Système,* pour décrire les différents intervenants et les contrôles qualité. Il est rédigé conformément à la recommandation_PAQL_VO du 23 Mars 1989 de l’AFCIQ (Association Française de Contrôle Industriel de la Qualité).

◊ “*Plans Types*”
Ce manuel reprend l’ensemble des productions documentaires cités dans le *Plan de Développement Système,* décrivant pour chaque document, son objectif et son plan détaillé.

Un Plan Qualité est défini par projet sur la base du *Manuel d’Assurance Qualité Système*. Sa mise en œuvre est faîtes sous la responsabilité du Chef de Projet et d’un Chargé de Qualité affecté au projet. Le Plan Qualité détaille:

◊ le but, le domaine d’application et les responsables,
◊ les documents applicables et les documents de référence,
◊ la terminology,
◊ la démarche d’assurance, de contrôle et d’évaluation,
◊ la présentation Qualité du projet logiciel
(par référence au *Plan de Développement Système)*,
◊ la description détaillée des activités de Qualité,
◊ l’évaluation des méthodes et moyens du projet,
◊ les documents de suivi Qualité du projet, et
◊ le bilan Qualité.





Exemple d'un cycle de vie et échéancier des productions pour une "Qualité Normale"

QUALITÉ

Plan de Développement Système

Manuel de référence

Version 1.2

Référence	DT/8012/M.002
Révision	1.2 - 6 Mars 1992
Auteur	Pierre-Yves MONNET
Diffusion	MC2
Accès	MC2 uniquement
Contrat	
Pages	
Résumé	Description des procédures et moyens pour définir et obtenir la Qualité MC2
Mots Clés	Qualité

Versions

Date	*Version*	*Auteur*	*Commentaires*
9 Sept. 1991	0.1	P.Y. Monnet	Création du document
25 Oct 1991	0.2	P.Y. Monnet	Définition du projet / Suppressions des activités qualités par phase.
21 Nov 1991	0.3	P.Y.Monnet	Ajout de conseils pour chaque phase, Définition de la phase Extension, Remise en forme de la partie “définition du projet” Validation du document La phase de test disparait
27 Nov 1991	1.0	P.Y. Monnet	Validation de la version et ajout de remarques de la part de C. Malka
11 Jan 1992	1.1	P.Y. Monnet	Ajout de conseils, Modification des buts et responsabilités de la recette interne, La phase "Mise en maintenance" est renommée en "mise en exploitation" Remise en forme de la bibliographie.
6 Mars 1992	1.2	P.Y. Monnet	Schéma de production de la DCAR Définition de la validation d’aptitude (VSR)/ SAT Intégration des remarques 3 & 4 concernant la DCAR et la réalisation du cahier de recette par le client.

Note au lecteur

Ce document présente le plan de développement d'un système à MC2. Il décrit le cycle de vie et les productions par phase du cycle de vie.

Le but de ce document est double : d'une part donner un référentiel à la direction technique quant à la manière de conduire un projet, d'autre part d'ouvrir le dialogue vis-à-vis des autres services de MC2. Il est donc appelé à évoluer.

Les différents contrôles sont explicités dans le **Manuel D'Assurance Qualité Système**, décrivant les différents intervenants, les contrôles qualité.

Pour chaque document à produire au cours du développement, le lecteur se reportera aux **Plans Types**, donnant un guide de rédaction.

Ces trois documents (Plan de développement Système, Plans types, Manuel D'Assurance Qualité Système) peuvent être lus de manière indépendante.

Préface

MC2 s'est engagé dans le développement de la "Qualité".

C'est une action volontariste et permanente ; elle est nécessaire pour assurer l'évolution et la croissance de l'entreprise.

MC2 a un métier, "l'intégration de systèmes" pour lequel l'impact qualité est déterminant :

◊ Sur le plan transfert de connaissance
Notre "savoir-faire" est un des éléments clés de notre métier, il nous différencie et nous positionne dans l'environnement concurrentiel. Mettre en oeuvre une démarche qui assure l'évolution et le transfert de ce "savoir-faire" est essentiel.

◊ Sur le plan coût
Chaque système que nous réalisons est unique, c'est un investissement conséquent pour le client qui supporte développement et évolution.
La contrainte budgétaire et de délai est très forte. Intégrer ce contexte et mettre en oeuvre un modèle avec pour objectif d'assurer qualité et évolution dans les contraintes précitées est un des résultats les plus directs de la démarche qualité.

Ce premier document contribue à présenter le modèle de l'entreprise pour la réalisation de sa production : "le système".

Ce modèle est en constante évolution, et un des bénéfices attendus de ce Manuel qualité est d'avoir un document qui répercute cette évolution, dans les délais les plus courts, et de permettre à toute l'entreprise de se référer à un même modèle.

Charles Malka
Directeur Général de MC2

Table des matières

Versions i

Note au lecteur ii

Préface iii

Table des matières iv

Liste des figures vi

1. Introduction 1

2. Définition du projet 2

3. Phases du cycle de vie 3

3.1. AVANT VENTE 4
3.1.1. Définition 4
3.1.2. Conseils 5

3.2. SPÉCIFICATION 5
3.2.1. Définition 5
3.2.2. Conseils 7

3.3. CONCEPTION 8
3.3.1. Définition 8
3.3.2. Conseils 8

3.4. RÉALISATION 9
3.4.1. Définition 9
3.4.2. Conseils 11

3.5. RECETTE INTERNE (IAT) 12
3.5.1. Définition 12
3.5.2. Conseils 13

3.6. RECETTE USINE (FAT) 14
3.6.1. Définition 14
3.6.2. Conseils 14

3.7. TRANSPORT - INSTALLATION 15
3.7.1. Définition 15
3.7.2. Conseils 15

3.8. RECETTE SITE (SAT) 16
3.8.1. Définition 16
3.8.2. Conseils 16

3.9. MISE EN EXPLOITATION 16
3.9.1. Définition 16
3.9.2. Conseils 17

3.10. MAINTENANCE 17
3.10.1. Définition 17

3.11. EXTENSIONS 17
3.11.1. Définition 17
3.11.2. Conseils 17

3.12. RÉSUMÉ 18

Annexe A - Fiche d'Anomalie 20

Annexe B - Rapport d'anomalie 22

Glossaire 24

Index 27

Fiche bibliographique 29

Formulaire Qualité 30

Liste des figures

Exemple d'un cycle de vie 4
Exemple d'une évolution 4
Document de suivi de projet financier 6
Cycle en V 10
Productions / cycle de vie 18
Charge des participants 19
Rapport d’anomalie 23

1. Introduction

Le présent Plan de Développement Système a deux buts :

- définir le cycle de vie utilisé à MC2 pour le développement de ses systèmes,
- fournir un référentiel à l'équipe projet du consultant technique à l'équipe support .

La décomposition du document est la suivante :

Le chapitre 2 donne la définition d'un projet, et ce qu'il recouvre.

Le chapitre 3 présente le cycle de vie, ainsi que les différentes productions.

Ce document est le résultat d'un groupe composé de J. Bouchet, G. Fournet, J.M.Marcastel, P.Y. Monnet, F. Olléon, J.P. Pantel, S. Sontag, et sera amené à évoluer.

2. Définition du projet

Chaque projet MC2 doit fournir au client un **outil** pour améliorer sa gestion de document. Cet outil peut être une **étude préliminaire** , une **étude complète** (qui sont des outils d'aide à la décision) ou un **système** , c'est-à-dire un ensemble cohérent de matériel et de logiciel fabriqués, adaptés ou intégrés par MC2, ou l' extension et plus généralement la **maintenance** d'un tel système. MC2 peut se trouver présent à toutes les étapes de fabrication de cet outil ou n'être en charge que de quelques étapes.

Un **projet** consiste à réaliser un outil (réalisation d'une étude, d'un système...).Les exigences de chaque projet sont :

- le respect des demandes du client, définies par les fonctionnalités demandées, mais également par les ressources données par le client, en terme de délai et de budget ,
- la rentabilité du projet pour MC2,
- l'intégration du produit final dans les systèmes informatiques du client présents ou à venir (donc respect des standards),
- le respect des standards du marché,
- la réutilisabilité des composants,
- l'avance technologique et la recherche de partenaires.

La définition pouvant être : "Fournir le bon système pour son juste prix agréé par les deux parties".

Dans le cas général, un projet suit les trois étapes : étude - système - maintenance, mais il arrive que MC2 ne vende qu'une étude, ou commence directement au niveau du développement du système

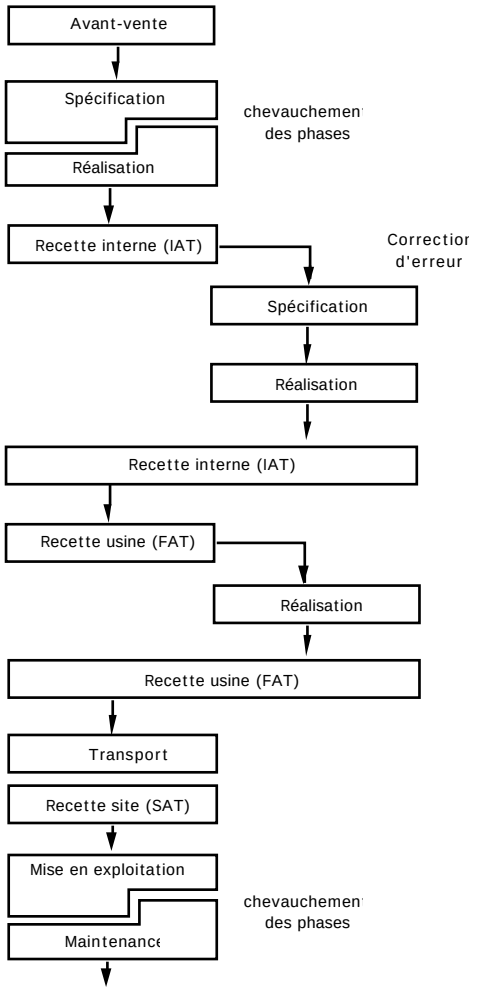
3. Phases du cycle de vie

Le cycle de vie du logiciel est décomposé en phase (étapes). Pour chaque phase, on décrit l'activité, les productions et on donne des conseils de réalisation

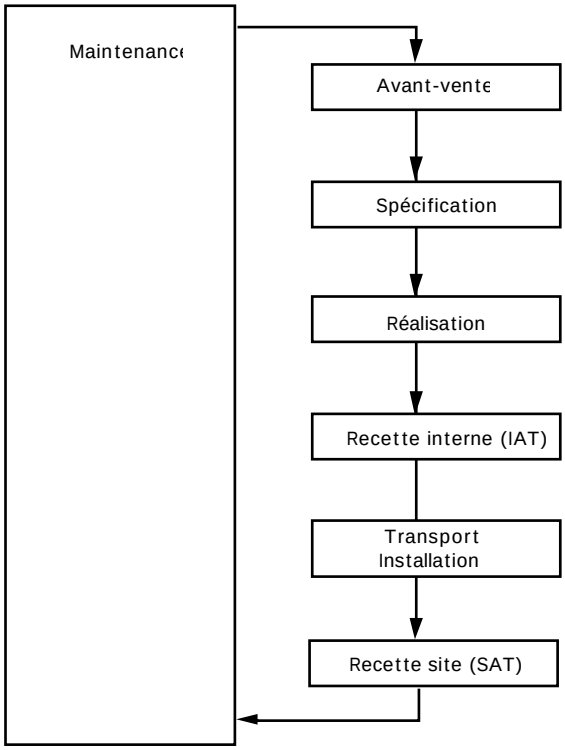
Cette partie décrit les **phases** du cycle de vie, et non pas le cycle de vie en lui même. Ainsi, l'équipe de développement est-elle libre de choisir entre un cycle de vie séquentiel, ou avec retour possible en arrière, ou avec chevauchement de phases, etc...

Plusieurs techniques et critiques peuvent être lues, telle la critique du cycle de vie en cascade de [Meyer 1990], disponible en annexe.

Voici par exemple comment les phases peuvent s'enchaîner. Le cycle de vie dépend de la gestion de projet.



Exemple d'un cycle de vie



Exemple d'une évolution

Toutes les productions de documents doivent se référer à un plan type MC2 (confère Qualité de la documentation). Le nom des différents documents à produire est donné en *italique* . Dans le cas où le plan est donné par le client, alors le rédacteur doit vérifier la complétude de ce plan avec le plan MC2 : le document doit avoir les mêmes renseignements. Il doit de plus indiquer dans le *plan Qualité* du projet le changement et la motivation de ce changement.

3.1. Avant vente

3.1.1. Définition

L'avant vente réunit un consultant technique , un commercial , le directeur technique et un chef de groupe. Elle aboutit à l'établissement d'une offre et d'un contrat . Nous ne parlons ici que de la proposition qui aboutit à la réalisation d'un projet.

Elle doit définir :

- le contexte technique, commercial et relationnel du projet,
- les ressources affectées (en temps et en argent),
- le système cible (ce qu'il faut réaliser), ainsi que ces extension s futures (qui ne font pas partie du premier système vendu).
- les données (structure, terminologie, cycle de vie, volume, sécurité, règles d'exploitation...),
- les fonctions du système,
- les interfaces avec le monde extérieur (logiciel, matériel, humain),
- les exigences opérationnelles (Qualité, performance, capacité, sécurité...),
- le dimensionnement du système (performance, volume).

L'ensemble doit être validé par la Direction Technique. La fin de phase est caractérisée par la signature du contrat.

Cette phase est détaillée dans la note interne DG/91035 , décrivant la procédure que doit suivre une offre (Sketch/Comité rouge/Envoi).

La production de cette phase doit être fixée par une réunion entre les différents acteurs (direction financière , direction commerciale , direction technique , consultant technique).

Production	La proposition de système produit : - une proposition technique , - une proposition financière , - un Document de Chiffrage Affaire (D.C.A.):
Responsable	Cette phase est sous la responsabilité d'un consultant technique , ou d'un chef de groupe dans le cas d'une évolution
Condition de passage	Le contrat est signé

3.1.2. Conseils

Pour éviter un retour arrière sur la phase, et donc sur le contrat, la proposition technique doit être complète, et vérifiée en détail par une personne de la direction technique. En effet, celle-ci s'engage sur la réalisation du projet.

3.2. Spécification

3.2.1. Définition

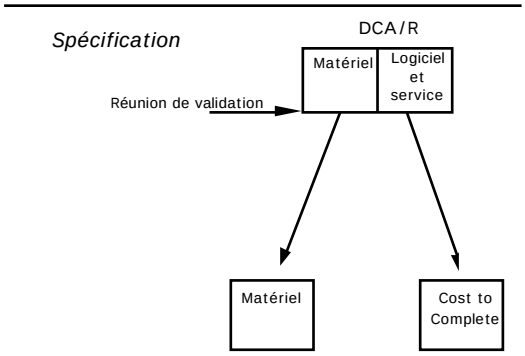
La phase de spécification décrit les prestations que doit assurer le système . Elle fournit une représentation complète et détaillée de ce que fera le système, et non pas comment il sera fait.
A ce niveau, le matériel, qui fait partie du contrat, est déterminé.

Les principaux points à raffiner (car ils doivent être présents dans l'avant vente) sont :

- les objectifs et les hypothèses du logiciel à réaliser,
- les contraintes liées à l'environnement (techniques, culturelles),
- les données (structure, terminologie, cycle de vie, volume, sécurité, règles d'exploitation...),
- les fonctions du système,
- les interfaces avec le monde extérieur (logiciel, matériel, humain),
- les exigences opérationnelles (Qualité, performance, capacité, sécurité...),
- les méthodes de conception et de suivi de projet,
- les différents outils,
- les rôles de chaque acteur du développement.
- la justification des choix techniques (tests de faisabilité),
- le dimensionnement du système (performance, volume),
- avoir lu l' offre et le contrat

A la fin de la phase, on sait "ce que doit faire le système" .

La phase de spécification débute par une réunion 0 , qui permet d'affiner les chiffres données par les C.T. lors de la phase d'avant vente, donnés dans la DCA. (Document de Chiffrage Affaire) Plusieurs réunions peuvent avoir lieu, jusqu'à l'acceptation par la DT. de la responsabilité de la réalisation du système, acceptation qui a lieu durant la réunion de validation. A ce moment, la DCA devient une DCA/R (Réalisation), et elle est décomposée en deux : la DCA qui comprend le matériel sera suivi par la DAC, et la DT. fournira le " Cost to Complete ", à savoir les ressources nécessaires pour terminer le budget. Ces deux informations servant à la D.F. (Direction Financière) de MC2.



Document de suivi de projet financier

CETTE PARTIE N'EST PAS ENCORE VALIDEE PAR LA D.F. ET LA DAC.

Les différentes production de la phase sont :

Production	- d'un *document de spécifications externes* , - du *planning général* , - du *plan Qualité* . - du *cahier de recette* , - des *spécifications du matériel* , - du document de *dimensionnement* , - d' *études de faisabilité* , validant les choix techniques.
Responsable	La phase est sous la responsabilité du chef de projet , aidé par le chargé Qualité et le consultant technique .
Condition de passage	La phase est achevée lorsque le document de spécifications est écrit, et, s'il est contractuel, accepté par le client., et quand le cahier de recette est cohérent avec les documents de spécifications (externe, matériel et dimensionnement).

3.2.2. Conseils

Il faut bien vérifier le modèle de données,
Pour chaque choix de la proposition technique, il faut vérifier leur faisabilité.

Il est préférable de faire rédiger le cahier de recette par le client : on vérifiera ainsi s'il a bien compris le fonctionnement de son futur système, et d'autre part cela permet de le faire participer à l'élaboration du système (il se sent plus concerné). Par contre, le chef de projet doit veiller à la cohérence de ce cahier de recette vis à vis des spécifications et les conditions générales de vente MC2, ainsi aux faits que les tests demandés soient “raisonnables”, en temps et coût.
Pour cela, il doit être absolument validé par MC2.
Il doit également vérifier la complétude du cahier de recette écrit par le client par rapport au plan type du cahier de recette tel que spécifié par MC2 (le plus simple étant de donner le plan type au client).

Dans le cas ou des fonctions du système sont exclusives entre elles, il faut l'indiquer très clairement. En générale, toutes les limitations doivent apparaître le plus tôt possible pour le client.

Si le système est appelé à remplacer un système existant (au sens large), il faut comprendre comment fonctionne le système précédent de façon à imaginer comment le nouveau système va le remplacer. Il convient également de décrire l’ancien système, pour vérifier que l’on à bien compris son but (vu que le document est lu par le client, on évitera ainsi les erreurs “je l’avais compris différemment").

Plus les spécifications seront complètes, plus on à de chance d’éviter les mauvaises surprises provenant de malentendus avec le client. De plus, les spécifications sont un rempart de protection si le client exige certaines opérations (du style : le système ne fonctionne pas sous mon ancien système d’exploitation...).

A CE NIVEAU, L'ERREUR LA PLUS COMMUNE EST LE MALENTENDU SUR LE PROBLÈME DU CLIENT

Ce type de problème peut entraîner la réalisation d'un logiciel mal ciblé : on va négliger les extension s de telles ou telles fonctions, alors que pour le client, c'était la deuxième étape obligée de son système.

3.3. Conception

3.3.1. Définition

La phase de conception permet de répondre à la question " comment réaliser le système ". Le but de la phase est d'effectuer le découpage du système en blocs élémentaires. Le niveau d'affinage de ces objets est laissée sous la responsabilité du chef de projet : ce peut être les processus du système, ou plus précis mais plus couteux en temps, la décomposition en modules, bibliothèques...

On doit vérifier la disponibilité des moyens de productions de la phase suivante, et faire un calcul de ressources (cas de ressources critiques). Ces différentes considérations doivent apparaître dans le planning de réalisation.

Les principaux points à définir sont :

- penser en terme de réutilisabilité,
- faire le découpage suivant le degré d'affinage choisi (processus, modules...), et justifier le choix,
- définir l'implémentation des données,
- prévoir les futures extension s demandées, et/ou potentielles,
- le rôle de chacun des blocs élémentaires (processus, modules, objets...), ses limites et fonctions,
- la méthode de travail entre les blocs dégagés (message inter-processus...).

Production	- d'un document de *spécifications internes* , - du *plan de test* , comprenant les tests unitaires et d'intégrations. - du *planning de réalisation* , - de la première version du *manuel de maintenance* , - de la première version du *cahier d'extension* .
Responsable	Le chef de projet et le chargé Qualité dans leurs domaines.
Condition de passage	Lorsque les documents de spécification et le plan de tests sont écrits et validés.

3.3.2. Conseils

Il faut effectuer la conception à plusieurs.

Il est préférable d'avoir une recommandation du support quant à l'architecture retenue.

Les outils de tests et les jeux de test doivent être prévus à ce moment, ainsi que les différents simulateurs de matériel (voir avec l'équipe support).

La lecture du chapitre I de." Object oriented software construction ", de Bertrand Meyer est fortement conseillée (confère bibliographie)

La liste des modules appartenant à MC2 doit être établie. Les sources de ces modules ne seront pas communiqués.

3.4. Réalisation

3.4.1. Définition

Le but de la phase est d'écrire le logiciel, et d'effectuer les tests unitaires et d'intégration. Les tests de validation (ou d'acceptation) étant effectués dans la phase suivante. C'est une phase critique quant à la qualité, car elle produit le logiciel, mais ce n'est pas la phase la plus importante de la vie du logiciel. En effet, dans le but d'améliorer la productivité, la tendance est d'utiliser le plus possible des outils (toolbox, générateur de code), qui mettent en fait en avant les erreurs de conception, beaucoup plus difficiles à corriger.

Les tests d'intégrations sont menés durant cette phase. Ils permettent de vérifier l'adéquation du logiciel par rapport à la conception, et non par rapport aux besoins du client. La phase se termine par l'élaboration des mesures du cahier de recette.

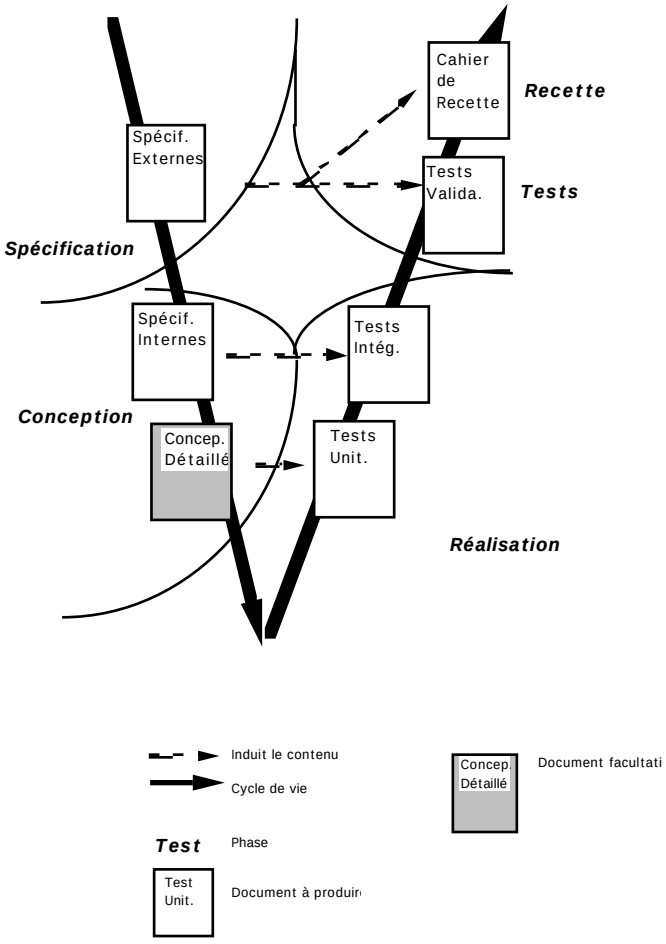
On discerne trois types de tests :

- les **tests unitaires** , sont les tests menés sur chaque module par le réalisateur.
- les **tests d'intégration** , lorsque un nouveau module est inséré dans le système,
- les **tests généraux** , qui testent le système complet en charge.

En général, seuls les tests généraux sont présents dans la recette, et les tests unitaires sont effectués dans la phase de réalisation; pour ceux-ci, seuls les principes peuvent être énoncés.

Cette phase doit aboutir à la validation du plan de test , dont il vaut mieux commencer la rédaction le plus tôt possible, lors de la phase d'analyse des besoins.

La rédaction du plan de test peut suivre le cycle en V :



Cycle en V

Dans ce modèle, on écrit chaque jeu de test pour chaque niveau lors de sa conception : ainsi, les tests généraux sont écrits dès que le document de spécification externe est validé, les tests d'intégration dès que les documents de conception sont écrits, etc...

Cette technique aboutit à un meilleur résultat pour deux raisons :

- on a en général plus de temps à ce moment d'écrire les jeux de test que lors de l'intégration,

- le programmeur est toujours influencé par la réalisation effectuée et a tendance à ne pas tester les bonnes choses : en effet, le réalisateur ne désire pas que son travail soit remis en cause implicitement, donc il désire que le test soit positif.

Il y a deux façons de vérifier un logiciel :

- de manière formelle, par preuve de programme,
- de manière empirique, par tests.

La première méthode n'étant guère réalisable de manière industrielle, on choisit en général une campagne de test. Pour tester complètement un logiciel, il faut pouvoir parcourir tous les chemins de ce logiciel, avec toutes les valeurs possibles des variables par rapport aux domaines de ces variables. Cet objectif étant impossible (car trop énorme), on se contente d'une série de tests que l'on choisit judicieusement. En tout état de cause, tant que le double objectif n'est pas atteint, un logiciel informatique ne pourra jamais être déclaré exempt de fautes.

Production	Le programme, le plan de tests mis à jour quand aux résultats de tests.
Responsable	Le chef de projet et le chargé Qualité dans leurs domaines.
Condition de passage	Lorsque l'ensemble du programme est écrit, les tests unitaires et d'intégration défini dans le plan de test effectués avec succès

3.4.2. Conseils

Il ne faut pas chercher à valider et à tester le logiciel en une fois, mais écrire de petits morceaux, les tester, et ensuite en écrire d'autres. Cette méthode favorise l'écriture des modules de gestion en premier (écriture ascendante).

Chaque développeur doit avoir son environnement personnel, et un dépôt des sources corrects. Ainsi, chacun est assuré d'avoir une version fonctionnant du logiciel, et cette technique favorise le versionnage et la validation. Pour le dépôt, il est préférable d'utiliser une base de programme.

Il est préférable d'avoir une étape d'intégration et de validation pour les sources devant être repositionnées dans le dépôt. Cette étape doit être menée par une personne de l'équipe qui n'a pas participé au développement de la partie, sur une base de test qu'il aura écrit et qui sera indiqué dans le plan de test.

Il faut écrire à ce moment, et en parallèle du développement, les différents manuels (exploitation, référence, utilisateur).

Il ne faut pas employer trop de niveaux dans l'arborescence disque de développement. 4 niveaux de profondeur semble être un maximum.

Lors des tests unitaires :
Tous les chemins doivent être testés
Cela signifie que le réalisateur du module choisit de bonnes valeurs pour les variables, et teste le résultat.

Les variables doivent être testées aux limites
Ainsi, dans le cas d'une variable caractère, on essayera avec le caractère du début de domaine, et le dernier caractère du domaine.

Les cas particuliers doivent être testés
Si on a un test "A>10", alors on doit essayer avec la valeur 10, et 11.
Bien sûr, dans le cas donné, d'une part ce test doit être réalisé pour parcourir tous les chemins, d'autre part il est très simple, le résultat est (aux défauts du compilateur prés) assuré. Par contre, il existe toujours de tels cas limites dans un logiciel.

Vérification et métriques
Des outils comme Lint en C devraient être systématiquement utilisés.

Lors des tests d'intégration :

La personne qui a écrit le module doit être présente, mais ce n'est pas elle qui mène la campagne de tests d'intégration de son module.

La personne qui mène la campagne de test doit être la même que celle qui a écrit le jeu de tests d'intégration, et elle doit être responsable du résultat produit.

Lorsque le langage ne peut pas le faire de manière automatique (langage C par exemple), les interfaces du module vers les autres modules devraient être contrôlés par rapport à la conception détaillée.

Une lecture croisée du code devrait être faite par la personne qui mène la campagne de test d'intégration, afin d'une part de trouver des défauts, d'autre part d'améliorer la connaissance du logiciel, enfin dans un but de confrontation des idées.

Tests généraux

Vu le métier de MC2, qui traite beaucoup de production, le système devrait être testé en charge.

Toutes les fonctionnalités doivent être testées.

Vu le caractère multi serveurs / multi machines, les cas d'exception, tel l'arrêt brutal d'une machine ou d'un serveur, devraient être testés.

De même, un périphérique qui se met à mal fonctionner devrait être découvert par le système.

Dans le cas d'un dysfonctionnement d'un matériel, le système devrait être toujours capable d'identifier le problème, de prouver son innocence, d'en avertir l'opérateur, et ensuite de proposer une solution.

3.5. Recette interne (IAT)

3.5.1. Définition

La recette interne est menée sans la présence du client, par l'équipe de développement, à laquelle il est préférable d'associer un Consultant technique , une personne du support technique et une personne du marketing .. Les buts de la recette interne sont :

- connaître les éventuelles erreurs et faire fonctionner le système en configuration réelle,
- permettre une répétition de la FAT, et ainsi assurer la qualité de la prestation des acteurs lors de cette FAT,
- vérifier la conformité aux spécifications,
- vérifier la maintenabilité du logiciel,
- connaître l'exploitabilité du système (qui est un facteur lourd à vérifier)

Le consultant technique du projet est invité.

Production	Le *cahier de recette* mis à jour. La liste des tâches et leur charge est confrontée avec le temps restant dans le *planning général* . du *manuel utilisateur* , - du *manuel de référence* , - du *Dossier d'exploitation* , Des *demandes d'évolutions* sont rédigées.
Responsable	Le chef de projet quant à sa réalisation, le chef de groupe quant aux résultats.
Condition de passage	Le cahier de recette a été parcouru, tous les tests relevés dans ce cahier effectués.

3.5.2. Conseils

Il est nécessaire de ne pas faire la recette trop tard (afin de se laisser une plage de correction), ni trop tôt (pas assez de parties écrites et testées). L'idéal semble être 15 jours avant la recette usine.

La recette interne devrait être menée par une personne extérieure à l'équipe de développement, afin de se mettre dans une situation de recette usine.

Le consultant technique devrait être présent.

Il faut repartir à zéro vis-à-vis du système : vider les fichiers temporaires, les bases de données...

Il faut débrancher la plate-forme des réseaux de MC2, afin de vérifier que tous les logiciels et ressources soient présents,

Si une installation est requise (sur une autre machine par exemple), elle doit être testée et c'est une personne qui n'a pas fait la procédure d'installation qui doit le faire,

Il faut essayer dans les tests de dépasser les dimensionnements : que se passe t-il s'il ne reste plus de place sur le disque ? Sur le disque optique ? Ces problèmes apparaîtront chez le client, il est utile de les tester avant à MC2.

Des tests d'endurance doivent être menés, mais, en ce qui concerne le matériel, l'autorisation du support Hardware doit d'abord être délivrée. En effet, c'est elle qui est le plus en mesure de vérifier l'usure du matériel.

Si on a prévu des modes dégradés, ils doivent être testés.

On doit produire des erreurs, afin d'étudier la réaction du système. Ces erreurs peuvent être de toutes natures (coupure de courant, réseau surchargé, disque en panne...). On essayera le plus possible de faire de la **simulation** (inutile de casser un disque dur : changer son nom ou sa localisation produit le même effet).

La recette interne peut être décomposée en plusieurs petites recettes. Il est préférable ainsi de tester le plus rapidement possible les sous-systèmes, dès qu'ils sont prêts (chevauchement avec la phase de réalisation).

3.6. Recette usine (FAT)

3.6.1. Définition

La recette usine est un constat des tests définis dans le document de recette. Elle précède la recette sur site. Normalement, tout ce qui est présenté dans la recette usine doit avoir été testé dans la recette interne.

Production	Résultats et taux de réussites des tests de la recette (mise à jour du document du *cahier de recette*) -- un procès verbal de recette (écrit), - une autorisation de livraison du système par le client
Responsable	Le maître d'ouvrage , guidé par le chef de projet.
Condition de passage	Après accord du client et l'autorisation de livraison du directeur général de MC2

3.6.2. Conseils

Il est essentiel de vérifier avant que le jeu de test donné dans la recette est cohérent avec les spécifications, et complet afin de tester le maximum de choses.

La recette doit être vue comme le dernier test avant l'envoi du système. Après, les corrections sont coûteuses (déplacements, arrêt du système...).

Il est préférable d'annoncer tout de suite les fonctions qui ne marchent pas.

Le chef de projet doit organiser le planning de la recette très précisément (au quart d'heure près), et doit le justifier et le présenter au client en début de recette. L'organisation a un rôle psychologique certain sur le client : un planning sérieux et suivi donnera une impression de rigueur qu'il transférera sur le système.

Les secrétariats (pour la réservation des salles, les pauses cafés) ainsi que le personnel de MC2 doivent être prévenus (afin d'éviter que quelqu'un choisisse ce jour là pour arriver déguisé en quoi que ce soit par exemple...).

3.7. Transport - Installation

3.7.1. Définition

La phase de transport consiste à transporter le système des locaux de MC2 chez le client. C'est une étape sensible. Le transport peut faire l'objet de deux contrôles de la part de MC2 :

- la configuration et l'environnement requis sont bien présents chez le client,
- le transport n'a pas endommagé les matériels.

Production	Un procès-verbal spécifiant si la configuration requise est bien présente chez le client, et si le transport s'est bien déroulé. Les *spécifications du matériel* sont remises à jour vis-à-vis de la recette matérielle
Responsable	Le maître d'ouvrage quant à la configuration requise, le chef de projet quant à la configuration envoyée, et les services spécialisés de MC2 dans le transport. Le procès-verbal est écrit par le chef de projet, co-signé par le maître d'ouvrage .
Condition de passage	Procès verbal de réception des matériels

3.7.2. Conseils

Définir les responsabilités à la réception du matériel chez le client.

Vérifier "qui emballe le matériel", "qui le met dans le camion", "qui le transporte", "qui le décharge", et "qui le déballe". Naturellement, ce sont les services spécialisés de MC2 qui effectuent la plupart de ces travaux, mais le chef de projet doit connaître ces détails, surtout vis-à-vis de la réception, afin d'éviter que les cartons passent une nuit en dehors des bâtiments par exemple.

La liste du matériel qui part doit être connue, carton par carton.

Le problème des douanes doit être vérifié dans le cas d'un passage de frontière.

Le plan de route du camion doit être connu (surtout sur la localisation des arrêts). Il faut éviter de faire partir le matériel le vendredi (ou reste le matériel le week end ?).

La plupart du temps, le matériel reste dans une pièce pour la recette usine. Cette pièce doit être localisée, et la procédure de transfert du matériel vers son lieu final identifié.

Lors de la réception, une recette matérielle, consistant à vérifier le bon fonctionnement du matériel hors système, doit être menée. C'est par exemple le test hard de l'imprimante, le boot de la machine...

Il faut prévoir des consommables pour la recette matérielle.

3.8. Recette site (SAT)

3.8.1. Définition

Une phase de recette site est à nouveau nécessaire, de manière à tester le système sur place, avec les matériels réels et dans l'environnement du client. Cette phase a pour but de proclamer le début de l'utilisation du système par les utilisateurs finaux. Elle réalise les Tests fonctionnels , en opposition avec les Tests d'aptitudes : on montre donc durant la recette site que les différentes fonctions sont bien présentes et fonctionnent correctement, ce qui ne signifie pas que le système répond au besoin exprimé par le client : seuls les tests d'aptitudes le permettront.

La procédure de recette est identique, et amène aux productions identiques à celles décrites lors de la recette usine.

Le chargé Qualité écrit le bilan Qualité du projet, inséré dans le Plan Qualité du projet.

Dans le cas des marchés d'états , cette phase s'appelle la VA (Validation d'Aptitude).

Production	Du *cahier de recette* et d'un procès verbal d'acceptation .
Responsable	Le maître d'ouvrage .
Condition de passage	Acceptation par le client du système.

3.8.2. Conseils

Souvent, l'acceptation définitive intervient après une période de validation qui commence le jour de l'acceptation de la recette site (cas des marchés d'état par exemple).

3.9. Mise en exploitation

3.9.1. Définition

Le but de cette phase est de donner le projet à l'équipe support de MC2, et de rester à l'écoute du client, qui commence l'exploitation de son système. Cette phase permet également au client de vérifier les Tests d'aptitude du système à répondre à ses besoins, en terme de performance de nombre de panne par exemple. Dans le cas des marché d'état français, cette phase est appelée VSR(Validation du Service Rendu).

Durant cette phase, l'équipe de développement reste à l'écoute du système, pour en déceler certaines erreurs, et les corriger, ou connaître les extension s possibles, et qui pourraient faire l'objet d'une offre .

Production	Un *cahier d'extension* est remis au consultant technique du projet.
Responsable	Le chef de projet et le responsable du support .
Condition de passage	Le service support accepte le projet.

3.9.2. Conseils

3.10. Maintenance

3.10.1. Définition

La phase de maintenance n'est pas effectuée par le chef de projet. Elle concerne d'autres services de MC2.

Elle commence le jour de l'acceptation de la recette site (SAT).

3.11. Extensions

3.11.1. Définition

Une extension est vue comme un projet. Elle peut donc suivre les mêmes phases qu'un développement. Les productions à fournir peuvent être par contre beaucoup plus allégées.

3.11.2. Conseils

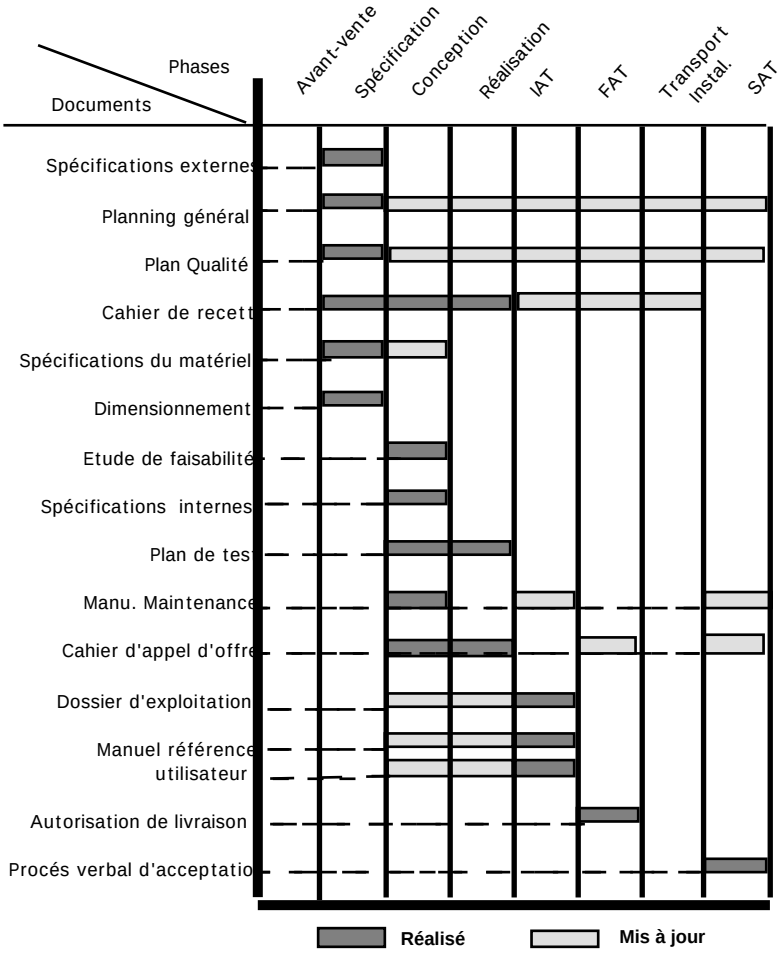
Il est préférable de baser le travail sur les demandes d'évolution, et de mener chacune d'elle comme un petit projet.

3.12. Résumé

Cette partie donne un résumé des productions sur un projet. Elle permet d'avoir une vue générale du projet.

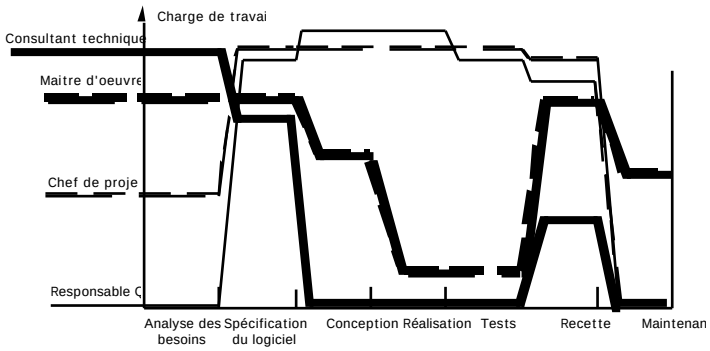
Suivant le niveau de Qualité choisi, et tel que donné dans le Manuel D'Assurance Qualité Système , certaines productions peuvent être délivrées dans des phases ultérieures. Le schéma donné ici montre le niveau de Qualité "Normal".

Productions par rapport aux phases :



Productions / cycle de vie

Le diagramme suivant montre la charge (théorique) des différents acteurs définis. Ce diagramme n'engage en rien la réalité.



Charge des participants

Annexe A - Fiche d'Anomalie

Cette annexe donne une fiche d'anomalie , qui doit être utilisée pour tout rapport d'anomalie ou d'erreur relevée au cours du développement. Elle est traitée en interne par MC2 (anomalie relevée par un membre de l'équipe ou par un participant Qualité, et suivie par l'équipe de développement).

Rappel :

Défaut : non conformité avec les spécifications de départ,
Erreur : spécification de départ fausse.

Une erreur exige une résolution avec le client, puisqu'elle provient de lui.

Fiche d'anomalie

Nature de l'anomalie : Défaut - Erreur *On indique la nature de l'anomalie*
Nom de l'inspecteur : *Nom de la personne ayant relevé l'anomalie*
Date du relevé :

Description

Localisation : *Dans un document, dans telle fonction... On indique OU et COMMENT reproduire l'anomalie. (description des données pour une fonction)*

Nature : *On indique l'anomalie ET PAS SA CORRECTION*

Correction envisagée

Gravité donnée : Grave - Gênant - Mineur *Le modérateur de l'inspection, le chargé Qualité... la personne responsable du relevé d'information indique la gravité de l'anomalie, pour prévoir sa correction. :*

Date : *On indique la date normale de correction, c'est à dire la date à partir de laquelle la correction sera effectuée*

Nom du correcteur : *La personne responsable de la correction est désignée*

Suivi de correction

On indique, avec une ligne par action, aussi bien les opérations de contrôle que de vérification de l'anomalie.

Date	Contrôlé par	Rapport

Dans un deuxième temps, on peut préciser l'anomalie plus en détail (données nécessaires, environnement...), voire de proposer une méthode de correction.

Annexe B - Rapport d'anomalie exploitation

Ce rapport est rédigé par l'utilisateur, ou la personne chargée par le client de l'exploitation, lorsqu'il découvre une anomalie en exploitation. Elle est ensuite envoyée à MC2, soit à l'équipe de développement, soit au support .

Rapport d'anomalie

(Remplir en CAPITALE)

Date : .....

Heure : .....

Machine : SUN 3/60 ☐ SUN 3/160 ☐ SUN 3/260 ☐
SUN 4/20 ☐ SUN 4/60 ☐ SUN 4/370 ☐
Autres .....

Lieu géographique (précis) : .....

Nom de la personne qui a constaté l'anomalie : ..... Téléphone : .....

Nom logique de la machine : ..... N° Internet : .....

Nom de la personne qui suit l'anomalie : ..... Téléphone : .....

Décrivez la marche à suivre qui fait apparaître l'anomalie :

.....

.....

.....

.....

.....

Décrivez l'anomalie :

.....

.....

.....

.....

.....

Cette anomalie est-elle : ☐ secondaire ☐ génante ☐ grave ☐ bloquante

Réservé MC2

Responsable : ..... Numéro : .....

Rapport d'anomalie

Glossaire

Autorisation de livraison	Autorisation remise à MC2 par le client après la recette usine (FAT)
C.T. :	Consultant Technique
Cahier d'extension :	Cahier remis à la partie commerciale de MC2, montrant les évolutions qui ont été vues par l'équipe de développement au cours de la réalisation du système.
Cahier de recette	Il détermine les différents tests que le client veut faire subir au système pour démontrer son bon fonctionnement par rapport à celui prévu dans les spécifications externes.
Chargé Qualité :	Le chargé Qualité est un membre de l'équipe de développement, chargé de l'application et du contrôle de la Qualité.
Consultant Technique :	Personne effectuant l'analyse des besoins du client. Son rôle principal intervient durant les phases d'avant-vente et de spécification.
DT.	Direction Technique
Défaut	Non conformité avec les spécifications de départ. Confère Erreur
Dimensionnement	Ce document démontre le dimensionnement du système, aussi bien en terme de performance que de capacité.
Direction Technique	Ensemble des personnes réalisant les systèmes
Erreur :	Spécification de départ fausse. Confère Défaut
Etude de faisabilité :	Etude permettant de vérifier que les choix techniques pris sont bien réalisables.
FAT :	Factory Acceptance Test (recette usine)
IAT :	Internal Acceptance Test (recette interne)

Maître d'ouvrage : Personne nommée par le client, et qui le représente. Il est chargé de la définition des besoins, des recettes, et d'assurer le suivi du projet côté client.

Dossier d'exploitation : (ou Dossier d'exploitation). Il décrit toutes les opérations utiles à l'exploitation du système.

Manuel d'utilisation : Il est remis à l' utilisateur final. Il peut y en avoir plusieurs (un par sous système par exemple). Il doit avoir un rôle pédagogique certain.

Manuel de Maintenance : Manuel remis au support, et qui indique ou sont les arborescences de développement, les exécutables, les fichiers de configuration...

Manuel de Référence : Manuel remis à l'administrateur du système, donnant différentes indications, le plus détaillées possibles, sur le système.

MC2 : Marketing Consultant Conseil

Outil : Vu au sens du client. Un outil peut être une étude, un système ou un service (maintenance).

Planning de réalisation : Planning découpant la phase de réalisation en tâche.

Plan de Test : Méthode prise pour mener la campagne de tests. Les différents jeux de tests sont également présents dans le document.

Planning Général : Il regroupe l'ensemble des phases du cycle de vie, avec les dates importantes.

Plan Qualité : Définit par rapport au Manuel D'Assurance Qualité Système , il donne les méthodes, assurances et contrôles effectués pour assurer la qualité de la réalisation finale.

Procès verbal d'acceptation : Acceptation par le client du système, après la recette site (SAT). Cette acceptation comporte souvent des réserves de temps.

Projet Opération effectuée par MC2 en vu de réaliser un outil

SAT: Site Acceptance Test (recette site)

Spécifications du Matériel : Elles contiennent la liste du matériel, mais également les fiches techniques de chaque matériel, la description de l'emplacement des matériels chez le client et la recette matérielle , montrant le bon arrivage.

Spécifications externes Description externe du système, vu par le client.

Spécifications internes : Document regroupant la décomposition du système en composant plus fin, permettant la réalisation informatique. Les liens entre les composants sont également décrits.

Support : Service de MC2 chargé de la maintenance hard et logiciel des systèmes réalisés.

Système :	Ensemble cohérent d'appareils matériels et logiciel formant un outil pour la gestion électronique de document.
Tests d'aptitudes :	Tests permettant de vérifier que le système répond bien aux aptitudes demandées, sur une période donnée.
Tests fonctionnels :	Tests mettants en évidence le bon fonctionnement des différentes fonctions du système.
VA :	Validation d'Aptitude - vocabulaire employé dans le cas des marchés d'états français. Cette validation est vérifiée durant la recette site.
VSR:	Validation de Service Rendu - vocabulaire employé dans le cas des marchés d'états français. Cette validation est vérifiée sur une période.

Index

anomalie 22
Autorisation de livraison 24
Bertrand Meyer 8; 29
blocs élémentaires. 8
C.T. 6; 24
cahier d’extension 8; 17; 24
cahier de recette 7; 13; 14; 16; 24
chargé Qualité 7; 24
chef de groupe 5; 13
chef de groupe. 4
chef de projet 7; 13; 17
commercial 4
conditions générales de vente 7
consultant technique 1; 4; 5; 7; 12; 13; 17; 24
contrat 4
Cost to Complete 6
cycle en V 10
D.C.A. 5
D.F. 6
DCA 6
DCA/R 6
Défaut 24
demandes d’évolutions 13
dépôt 11
DG/91035 5
dimensionnement 7; 24
directeur général 14
directeur technique 4
direction commerciale 5
direction financière 5
direction technique 5; 24
document de spécifications externes 7
Dossier d’exploitation 13; 25
DT. 6; 24
Erreur 24
étude complète 2
Etude de faisabilité 24
étude préliminaire 2
études de faisabilité 7
extension 2; 5; 7; 8; 16; 17
FAT 24
Fiche d’Anomalie 20
IAT 24
maintenance 2; 17
maître d'ouvrage 14; 15; 16; 24
Manuel D'Assurance Qualité Système ii; 18; 25
manuel de maintenance 8
manuel de référence 13
manuel utilisateur 13
marchés d’états 16
marketing 12
métriques 11
Object oriented software construction 8; 29
objets 29
offre 4; 5; 6; 16
outil 2
plan de test 8; 9
plan Qualité 4

plan Qualité. 7
planning de réalisation 8
planning général 7; 13
Plans Types, ii
procès verbal d’acceptation 16
projet 2
proposition 4
proposition financière 5
proposition technique 5
Rapport d’anomalie 22
recette interne 24
recette matérielle 15; 25
recette site 17; 25
recette usine 24
réunion 0 6
spécifications du matériel 7; 15
spécifications internes 8
support 1; 13; 16; 17; 22
support technique 12
système 2
Tests d’aptitude 16
Tests d’aptitudes 16; 25
tests d'intégration 9
Tests fonctionnels 16; 25
tests généraux 9
tests unitaires 9
utilisateur 25
VA 16; 26
VSR 16; 26

Fiche bibliographique

AFCIQ 1990
AFCIQ, "Guide pour la rédaction d'un Plan D'assurance Qualité" - AFCIQ/SL/90/005/-1

AFCIQ 1990
AFCIQ ,"Guide pour la rédaction d'un plan de développement logiciel" - AFCIQ/SL/90/007/ - 1

MEYER 1988
Bertrand Meyer , " Object oriented software construction " Prentice-Hall Internationnal (UK)Ltd,Hemel Hempstead 1988; traduction française : "Conception et programmation par objets " InterEdition Paris 1991

MEYER 1990
Bertrand Meyer "The new culture of software development", Journal of object oriented programming vol 3, no 4 nov/dec 1990

Formulaire Qualité

Ce formulaire est à utiliser pour toute remarque vis à vis de ce document. Il est à remettre au groupe chargé de la définition de la Qualité au sein de la DT.

Qualité MC2

Remarque

Demandeur :

Document Manuel Qualité PDS Plans types Autre :
Localisation :

Description / amélioration souhaitée

Fiche de suivi :

	Date	Remarques
Notification		Par
Discuté le		
Status		Accepté Refusé Déféré au
Modifié		Version document :

Remarques / modifications :

suite au verso si besoin

QUALITÉ

Plans types

Manuel de référence

Version 0.4

Référence	DT/8012/M.003
Révision	0.4 - 5 Mars 1992
Auteur	Pierre-Yves MONNET
Diffusion	MC2
Accès	MC2
Contrat	Qualité
Pages	
Résumé	Donnée des plans types de documents
Mots Clés	Qualité

Versions

Date	*Version*	*Auteur*	*Commentaires*
28 mai 1991	0.1	P.Y. Monnet	Version préliminaire
11 Déc. 1991	0.2	P.Y. Monnet	Validation des Spécifications externes, modification du plan
11 Jan 1992	0.3	P.Y. Monnet	Validation des Cahier de recette, Spécifications internes, Spécification du matériel
5 Mars 1992	0.4	P.Y. Monnet	Validation et éclatement du dossier d'exploitation en Manuel d'installation et Manuel d'exploitation, Validation du dossier de dimensionnement,

Note au lecteur

Ce document est en cours de validation par le groupe Qualité MC2. Certaines parties sont par contre déjà utilisables car validées. Les autres ne sont que des propositions. Dans ce cas, la mention "Ce document n'a fait l'objet d'aucune discussion" est apposée.

Les différentes phases d'un projet sont données dans le **Plan de Développement Système** . Ce document fait référence à un ensemble de documents, dont les plans types sont dans ce document.

Les différents contrôles sont explicités dans le **Manuel D'Assurance Qualité Système** décrivant les différents intervenants, les contrôles qualité.

Ces trois documents (Plan de développement Système, Plans types, Manuel D'Assurance Qualité Système) peuvent être lus de manière indépendante.

Table des matières

Versions i

Note au lecteur ii

Table des matières iii

Liste des figures ix

Liste des tables x

1. Introduction 1

2. Remarques générales 2

2.1. PRÉSENTATION DES DOCUMENTS 2

2.2. RÈGLES D'ÉCRITURES 3

3. Autorisation de livraison 5

3.1. PRÉSENTATION 5

3.2. PLAN TYPE 5

3.3. CONSEIL 5

4. Cahier d'extensions 6

4.1. PRÉSENTATION 6

4.2. PLAN TYPE 6

I. Introduction 6
II. Bilan du système 7
III. Extensions 7
IV. Documents de référence, Glossaire, Index, Annexes (facultatif) 7

4.3. CONSEILS 7

5. Cahier de recette 8

5.1. PRÉSENTATION 8

5.2. PLAN TYPE 8
I. Introduction 8
II. Généralités 9
III. Engagement du client 9
IV. Recette matérielle/Logiciels de base 9
V. Documentation 11
VI. Formation 11
VII. Tests logiciels 12
VIII. Liste des tâche 13
IX. Bilan général : procès verbal de recette 13
X. Documents de référence, Glossaire, Index, Annexes (facultatif) 13

5.3. CONSEILS 14

6. Conception détaillée 15

6.1. PRÉSENTATION 15

6.2. PLAN TYPE 15
I. Introduction 15
II. Contexte de développement 16
III. Spécification détaillée 16
IV. Documents de référence, Glossaire, Index, Annexes (facultatif) 17

6.3. CONSEILS 17

7. Dimensionnement 18

7.1. PRÉSENTATION 18

7.2. PLAN TYPE 20
I. Introduction 20
II. Principe général 20
III. Rappel des spécifications 20
IV. Données et hypothèses 21
V. Tests effectués 21
VI. Rappel de l'architecture 21
VII. Calculs détaillés 21
VIII. Points critiques généraux 22
IX. Améliorations 22
X. Conclusion 22
XI. Documents de référence, Glossaire, Index, Annexes (facultatif) 22

7.3. CONSEILS 23

8. Etudes annexes 24

8.1. PRÉSENTATION 24

8.2. PLAN TYPE 24
I. But 24
II. Terminologie 25
III. Concepts (ou principes) 25
IV. Utilisation 25
V. Performances 25
VI. Configuration 25

VII. Fiche technique 26
VIII. Problèmes reconnus 26
IX. Documents de référence, Glossaire, Index, Annexes (facultatif) 26

8.3. CONSEILS 26

9. Etude de faisabilité 27

9.1. PRÉSENTATION 27

9.2. PLAN TYPE 27

9.3. CONSEILS 27

10. Journal de bord 28

10.1. PRÉSENTATION 28

10.2. PLAN TYPE 28

10.3. CONSEILS 28

11. Manuel d'exploitation 29

11.1. PRÉSENTATION 29

11.2. PLAN TYPE 30
I. Introduction 30
II. Administration 30
III. Utilisation normale 32
IV. Maintenance 32
V. Documents de référence, Glossaire, Index, Annexes (facultatif) 33

11.3. CONSEILS 33

12. Manuel d'installation 34

12.1. PRÉSENTATION 34

12.2. PLAN TYPE 34
I. Introduction 34
II. Installation 34
III. Réinstallation 35
IV. Environnement de test 35
V. Documents de référence, Glossaire, Index, Annexes (facultatif) 36

12.3. CONSEILS 36

13. Manuel de gamme 37

13.1. PRÉSENTATION 37

13.2. PLAN TYPE 38
I. Introduction 38
II. Gammes matérielle 38
III. Gamme d'installation 38
IV. Gammes d'exploitation 38
V. Documents de référence, Glossaire, Index, Annexes (facultatif) 38

13.3. CONSEILS 38

14. Manuel de maintenance 39

14.1. PRÉSENTATION 39

14.2. PLAN TYPE 39
I. Introduction 39
II. Définition du site matériel 39
III. Définition du site logique 40
IV. Localisation des indicateurs 40
V. Outils et méthodes 40
VI. Documents de référence, Glossaire, Index, Annexes (facultatif) 40

14.3. CONSEILS 40

15. Manuel de référence 41

15.1. PRÉSENTATION 41

15.2. PLAN TYPE 41

15.3. CONSEILS 41

16. Manuel utilisateur 42

16.1. PRÉSENTATION 42

16.2. PLAN TYPE 42

16.3. CONSEILS 42

17. Planning de réalisation 43

17.1. PRÉSENTATION 43

17.2. PLAN TYPE 43
I. Introduction 43
II. Méthode de calcul 43
III. Présentation du planning 44
IV. Gestion des retards 44
V. Suivi de planning 44
VI. Bilan de réalisation 44
VII. Documents de référence, Glossaire, Index, Annexes (facultatif) 44

17.3. CONSEILS 44

18. Plan de tests 45

18.1. PRÉSENTATION 45

18.2. PLAN TYPE 45
I. Introduction 45
II. Démarche générale 45
III. Tests généraux 46
IV. Tests des serveurs 46
V. Tests d'intégration 47
VI. Tests unitaire 47
VII. Conclusion 47
VIII. Documents de référence, Glossaire, Index, Annexes (facultatif) 47

18.3. CONSEILS 47

19. Planning général 48

19.1. PRÉSENTATION 48

19.2. PLAN TYPE 49
I. Introduction 49
II. Référentiel du projet 49
III. Ressources 50
IV. Cycle de vie 50
V. Méthode de calcul 51
VI. Etapes 51
VII. Evaluation détaillée 52
VIII. Vue du client 52
IX. Vue des personnels 53
X. Définition des points de contrôle 53
XI. Bilan 53
XII. Documents de référence, Glossaire, Index, Annexes (facultatif) 53

19.3. CONSEILS 53

20. Plan Qualité 54

20.1. PRÉSENTATION 54

20.2. PLAN TYPE 54

20.3. CONSEILS 54

21. Spécifications du matériel 55

21.1. PRÉSENTATION 55

21.2. PLAN TYPE 55
I. Introduction 55
II. Présentation générale 56
III. Présentation des sous systèmes 56
IV. Inventaire 57
V. Historique des changements 57
VI. Validation de la pré visite 58
VII. Fiches matérielles 58
VIII. Documents de référence, Glossaire, Index, Annexes (facultatif) 58

21.3. CONSEILS 58

22. Spécifications externes 59

22.1. PRÉSENTATION 59

22.2. PLAN TYPE 60
I. Introduction 60
II. Contexte 60
III. Description générale 61
IV. Description des objets manipulés 63
V. Description détaillée 64
VI. Documents de référence, Glossaire, Index, Annexes (facultatif) 68

22.3. CONSEILS 68

23. Spécifications internes 69

23.1. PRÉSENTATION 69

23.2. PLAN TYPE 70
I. Introduction 70
II. Contexte de développement 70
III. Description générale 70
IV. Description détaillée 72
V. Données 74
VI. Description par fonction 74
VII. Documents de référence, Glossaire, Index, Annexes (facultatif) 75

23.3. CONSEILS 75

Glossaire 76

Index 79

Fiche bibliographique 81

Formulaire Qualité 82

Liste des figures

Exemple de calcul de dimensionnement 18
Résolution d’une erreur 32
Planning par phase 51
Planning détaillé 52
Exemple de système 56
Exemple de sous système 57
Exemple de "data flows" 65
Exemple de description générale 71
Exemple de description détaillée 72
Exemple de description par appel 75

Liste des tables

Recette des engagements des clients 9
Recette du matériel 10
Recette de la documentation 11
Recette de la formation 12
Recette des tests logiciels 12
Bilan des tests logiciels 13
Différents types de cycle de vie 50
Inventaire du matériel 57

1. Introduction

Le but de ce document est de présenter l'ensemble des plans types des documents à rédiger, tels qu'ils sont fixés par le Plan de développement système.

Pour chacun des documents, on commence par définir CE QU'IL EST et SON BUT. Pour cela, on le replace dans l'ensemble du projet. On insiste donc sur son contenu principal. Ensuite, on donne le plan type à suivre, en explicitant chaque paragraphe

2. Remarques générales

2.1. Présentation des documents

Cette partie présente les différents objets décrits dans ce document.

Le **Dossier technique** est écrit par le consultant technique. Il regroupe les différentes demandes du client, et constitue une première ébauche de la demande fonctionnelle du client.

Le Plan Qualité n'est pas décrit dans ce document. Il reprend le plan du manuel Qualité MC2, et doit être défini par rapport à lui.

Le document de **spécifications externes** décrit le plus précisément possible les différentes fonctionnalités attendues, sans donner les solutions de réalisation. Il donne le CE QU'IL FAUT FAIRE du produit. Il doit être validé par le client.

Le document de **Plan de tests** est la description de la méthode de tests à employer sur le projet, ainsi que le type de tests à effectuer.

Les **spécifications du matériel** donnent l'ensemble du matériel, ainsi que la stratégie de résolution du problème vis-à-vis du matériel (QUI FERA QUOI).

Le document de **spécifications internes** décrit l'architecture du système logiciel, en s'appuyant sur les spécifications matérielles. Il décrit quels logiciels doivent être développés, quelles sont les couches de bases, les objets manipulés, les systèmes de communication, etc...

Le dossier de **dimensionnement** permet de vérifier les temps de réponses et capacités exprimés par le client sont acceptés par le système. En s'appuyant sur la description des matériels et sur des méthodes statistiques, on doit démontrer que la charge du système est possible.

Le document de **conception détaillée** reprend le document de spécifications internes, mais avec une granularité plus grande. Il donne par exemple la structure des données dans le langage de programmation, et les différentes interfaces des modules logiciels et matériels.

Le **Document de maintenance** est donné au support.

Le **Cahier d'extension** est remis au consultant technique. Il permet à l'équipe de développement de décrire les évolutions du système, une fois celui-ci réalisé. Il peut contenir des aspects fonctionnels (rajout de fonctions qui se sont révélées nécessaires à l'utilisation), ou des aspects opérationnels (augmenter les performances de telles parties...).

Le **Cahier de recette** est un document contractuel. Il permet de définir les jeux de tests effectués lors de la recette, ainsi que les résultats de ces tests le jour de la recette.

Le **Planning général** est un document de suivi de projet et d'étude de coût. Il présente aussi le planning général, et permet le calcul des impacts (retards, défaut de ressource....)

Le **Planning de réalisation** est le document de travail lors de la phase de développement. Sa granularité est plus faible que le planning général, et il doit tenir compte du plan de tests.

2.2. Règles d'écritures

Chaque document comprend une introduction. Cette introduction doit contenir une présentation du document et de son objectif. Le lecteur doit savoir ce qu'il va trouver. On effectue également un bref rappel du projet.

Ensuite, en fin de document, on peut trouver les chapitres suivants :

- Documents de référence
- Glossaire
- Index
- Annexes

Le chapitre **Document de référence** liste tous les documents ayant trait au projet et pouvant aider le lecteur à comprendre les concepts introduits dans le document. Il fournit une liste définissant pour chaque document :

- le titre,
- l'auteur,
- les références du document si elles existent,
- le sujet du document,
- la source du document,
- un numéro permettant de le citer dans le présent dossier.

Le **Glossaire** contient la définition de tous les termes, concepts et sigles spécifiques au document ou au projet. Il est très important car il permet de préciser des termes qui nous paraissent évidents et qui peuvent avoir une toute autre signification dans l'esprit du client.

Les termes du clients doivent être définis : un plan, une page pour éviter le phénomène contraire, ou c'est le Chef de projet qui aurait mal compris le vocabulaire du client. Un certain nombre de termes peuvent être automatiquement définis, comme un scanner, un disque optique, etc... : ce sont les mots courants de notre métier. Ils sont définis dans le Manuel Qualité MC2.

Les glossaires ne doivent pas être dupliqués : le glossaire général doit se trouver dans les spécifications externes et surtout dans le Plan Qualité du projet. Les glossaires des autres documents permettent de redonner les mots locaux au document, pour éviter ainsi au lecteur un retour constant au plan Qualité.

L'**index** comprend, dans l'ordre alphabétique, l'ensemble des mots avec les pages où ils apparaissent. Ces mots doivent être significatifs. Il permet une utilisation rationnelle du document. Il est vivement conseillé d'avoir un outil automatique pour son élaboration.

Les **Annexes** permettent de ranger les éléments pouvant aider le lecteur en tant que support de lecture du document.

3. Autorisation de livraison

CE DOCUMENT N'A FAIT L'OBJET D'AUCUNE DISCUSSION ACTUELLEMENT

3.1. Présentation

3.2. Plan type

3.3. Conseil

4. Cahier d'extensions

CE DOCUMENT N'A FAIT L'OBJET D'AUCUNE DISCUSSION ACTUELLEMENT

4.1. Présentation

Ce document est remis au Consultant technique chargé du projet. Il doit décrire :

- la réalisation finale, en décrivant précisément les changements et les raisons de ces changements (matériels, logiciels),
- les différents coût associés, en reprenant les coûts donnés par le consultant dans la DCA, en précisant les changements,
- les fonctions supplémentaires que le client désire, ou que la mise en exploitation rendent évidentes, afin que le consultant puisse refaire une offre.

Ce document est donc d'une part le bilan du projet, d'autre part l'ouverture vers les évolutions possibles du système.

4.2. Plan type

I. Introduction

Confère remarques générales.

II. Bilan du système

On donne en quelques lignes le bilan du système tel qu'il a été réalisé, l'accueil des utilisateurs, les fonctions bien réalisées, et celles qui ont été mal ciblées.

III. Extensions

Les fonctionnalités sont classées par ordre de priorité, au sens de l'équipe de réalisation. Il peut s'agir soit d'évolution de fonction existante, soit de nouvelles fonctions. On décrit chaque fonction en donnant un coût approximatif de l'effort requis.

IV. Documents de référence, Glossaire, Index, Annexes (facultatif)

Confère remarques générales.

4.3. Conseils

5. Cahier de recette

5.1. Présentation

Cette partie décrit un plan type de cahier de recette. Il est fondamental, car la recette permet de terminer normalement un contrat. Il est réalisé par le maître d'oeuvre, en collaboration avec le chef de projet, et le consultant technique.

Il doit être complété en fin de recette par le procès-verbal de recette, validé par le client, qui constate l'existence et la conformité du produit aux spécifications. Il peut comporter des réserves sur des points précis.

Il y a deux recettes : une en usine (dans les locaux de MC2), et une sur site. Il est conseillé de remplir un cahier de recette par recette : il est bien sûr rédigé une fois pour toutes, mais les résultats des tests sont notés sur chaque cahier. On peut, de plus, faire un cahier de commentaire et un cahier de problème, et sur certain tests, faire référence à ces cahiers.

5.2. Plan type

I. Introduction

Confère remarques générales.

II. Généralités

II.1 Démarche de test

On donne la démarche générale de tests : ce peut être par exemple
- vérification de tout le matériel du système,
- tests de la partie scanning,
- tests de la partie stockage,
- tests de la partie consultation...

II.2 Planning

On donne le planning de la recette, jour par jour. Celui-ci doit être le plus précis possible.

II.3 Environnement

On précise l'environnement de tests (par exemple, la connexion avec telle machine sera effective, ou le réseau sera débranché entre la plate-forme et le reste de MC2...)

II.4 Protocole

Sont identifiés les horaires, les personnes qui participeront à la recette, ainsi que le protocole d'exécution (qui effectue les opérations), etc...

III. Engagement du client

On donne dans cette partie tous les engagements du client pour la recette : il doit fournir un jeux de données, etc...

On peux également donner ici une liste de tests menés sur l'environnement du client, sous la même forme que ceux plus loin, à savoir sous la forme d'un tableau (ou d'une fiche) :

Nom	Objectif	Situation de départ	Action	Résultat attendu	Date	Présents	Status	Commentaire

Recette des engagements des clients

Voir ci-après la description des champs

IV. Recette matérielle/Logiciels de base

La recette matérielle consiste :
- à vérifier la présence des matériels,
- à tester leur fonctionnement hors applications.

Le plus simple est d'extraire la liste des matériels des spécifications du matériel. Les logiciels de base sont tous les produits logiciels fournis avec le système (Operating System, Windows, ScriptWorks...).

De plus, on peut décomposer la liste en Composant/Module, suivant le degré de granularité visible par le client : ainsi, on peut parler du "composant Com Plotter", ou le décomposer en "Un Com Plotter Wicks & Wilson et un PC plus une carte Ethernet dans lePC".

En général, il vaut mieux décrire les composants jusqu'à l'arrivée sur des éléments comportant des numéros de série.

La liste se présente sous la forme de tableau :

Nom	Libellé	Numéro de série	Date	Présents	Status	Commentaire
IFF3	Carte IFF3	334332			Présent	
Com Plotter	**Composant**					
Wicks & Wilson	Com Plotter	33555BG			Fonctionne	
PC.		88765556H			Différé	Ecran en panne P25
P66	Carte Ethernet	887667			Différé	2 ème phase

Recette du matériel

Nom : nom de la carte, du matériel...

Libellé : description plus précise et non ambiguë

Numéro de série : numéro de série principal de matériel

Les autres champs sont rédigés lors de la recette:

Date : Date de la vérification,

Présents : personnes présentes (MC2 et client),

Status : Présent/Fonctionne/Différé/non conforme
Présent : on n'a pas pu tester le bon état de marche du système,
Fonctionne : le matériel fonctionne (hors application),
Différé : soit parce que le test ne fait pas partie de la phase, soit parce qu'il ne marche pas. Dans les deux cas, un commentaire doit être rédigé. Notez la référence à un problème (P25) dans la zone commentaire. La description de ce problème se retrouvera dans le cahier de problème.

Non conforme : Les spécifications annoncées par le constructeur (performances, connexion) ne sont pas assurées. Ce problème ne devrait jamais apparaître lors d'une recette, mais avoir fait l'objet d'Etude de faisabilité auparavant.

Commentaire : Rempli si le status est Différé ou non conforme.

V. Documentation

On donne la liste des documents fournis par MC2, ainsi que le support (papier, informatique).

En général, le contenu de la documentation n'est pas validé le jour même, mais après une période de lecture de la part du client.

Nom	Support	Nb	Date	Présents	Status	Commentaires
Manuel de réfé.	Papier	2				
Man	Info.	1				

Recette de la documentation

Nom : nom du document

Support : papier ou informatique

Nb : Nombre d'exemplaire fourni

Date : Date de la vérification

Présents : personnes présentes (MC2 et client)

Status : Présent/Différé/non conforme
Présent : La documentation est présente
Différé : Documentation non terminée...
Non conforme : les spécifications annoncées par MC2 n'ont pas été respectées (plan incomplet...).

Commentaire : Rempli si le status est Différé ou non conforme.

VI. Formation

On dresse ici le procès verbal des formations données au client. Deux bilans sont donnés par formation :
- elle a été effectuée,
- elle a été comprise.

Ce qui donne :

Nom	Participants	Date/durée	Date	Présents	Status	Commentaires
Administrateur		10/12/91 - 10 j.				
Opérateurs		15/12/91 - 2 j.				

Recette de la formation

Nom : nom de la formation

Date/durée : date et durée de la formation

Date : date de la vérification

Présents : personnes présentes pour la validation

Status : Effectuée/Comprise/Incomplète
Effectué : La formation à été effectuée
Comprise : Elle a été acquise par les personnes qui l'ont suivie
Incomplète : Il manquait des choses

Commentaire : Remplis si le status est Incomplet

VII. Tests logiciels

VII.1 Démarche

On donne ici la démarche de tests que l'on a choisie. On ne parle ici que de la démarche pour les tests logiciels.

VII.2 Tests fonctionnels

On donne pour chaque test :

Nom	Objectif	Situation de départ	Action	Résultat attendu	Date	Présents	Status	Commentaires

Recette des tests logiciels

Vu la complexité des tests, on peut rédiger, pour la partie gauche, une fiche par tests. Par contre, à la fin de cette partie, on doit trouver un tableau synthétique donnant :

Nom	Date	Présents	Status	Commentaires

Bilan des tests logiciels

Conseil : les problèmes doivent être rédigés en même temps, de manière à être co-signés par le client.

VII.3 Tests opérationnels

Ils correspondent aux spécifications opérationnelles des spécifications externes. Ils permettent de mettre en évidence :
- les performances
- les propriétés énoncées,
- la capacité,
- la sécurité,
- l'exploitation du logiciel,
- l'ergonomie de l'interface,
- les modes dégradés.

On décrit chaque test comme ci-dessus.

VIII. Liste des tâche

On donne une liste de tâches à effectuer après la recette, en donnant le descriptif de la tâche et une date de validation. Ce peut être la lecture de la documentation par le client, le lancement de certains tests qui n'ont pu être faits faute d'un composant (un fichier de données n'as pas été donné par le client), de nouveaux tests à écrire ou tout simplement des corrections à effectuer.

Pour chaque test on donne :
- le nom,
- une description,
- un responsable (MC2, client...),
- une date de vérification.

IX. Bilan général : procès verbal de recette

Un bilan général doit être rédigé. Il permet de spécifier le résultat de la recette, et synthétise tous les comptes-rendus des tests.

Remarque : si la recette n'est pas validée par le client, on l'indique sur le procès verbal, et une autre recette sera organisée.

X. Documents de référence, Glossaire, Index, Annexes (facultatif)

Confère remarques générales.

5.3. Conseils

6. Conception détaillée

CE DOCUMENT N'A FAIT L'OBJET D'AUCUNE DISCUSSION ACTUELLEMENT

6.1. Présentation

Ce document intervient après le document de spécifications internes. Il est facultatif. En effet, il est intéressant que si ce n'est pas la même équipe qui réalise et qui conçoit, ou si l'équipe est grosse. Vis-à-vis du document de spécification interne, ce document décrit comment les spécifications internes sont implémentées dans le langage choisi. On donnera ainsi le contenu précis des modules, avec les fonctions internes au module, les algorithmes utilisés et les données privées de chaque module.

Remarque : pour les projets maintenus en interne, ce document n'est à priori, pas requis. De plus, si les normes de programmation sont suffisantes, on pourrait imaginer de l'extraire directement du logiciel.

6.2. Plan type

I. Introduction

Confère remarques générales.

II. Contexte de développement

Description des éléments d'environnement de développement.

- la configuration informatique (matériel, système d'exploitation, ...)
- le(s) langage(s) utilisé(s)
- l'ensemble des utilitaires nécessaires au programme
- le système de gestion de base de données, (ORACLE, INFORMIX, ...)
- le réseau utilisé (Transpac, MAP, MSNET, ...)

III. Spécification détaillée

Pour chaque module on fournit la fiche suivante (le type de classement des fiches sera choisi de manière judicieuse.) :

DESCRIPTION DES DIFFERENTS Modules

Nom du module

On donne le module.

Philosophie d'utilisation

On donne la philosophie d'utilisation, les contraintes générales du module (telle fonction ne doit pas être utilisée avant telle autre, etc...). A ce niveau, il est intéressant de prévoir un mode de vérification à l'intérieur du module, pouvant être déconnecté (par une définition d'un symbole protection par exemple). Ce mécanisme est à rapprocher des pré-conditions d'Effel.

Ressources importées

On donne toutes les ressources (variables, types, fonctions) importées du module, en explicitant leur rôle.

Ressources exportées

On donne toutes les ressources exportées du module.

Réalisation du module

On donne le mode de réalisation du module, les algorithmes utilisés, etc... Normalement, en lisant uniquement les paragraphes précédants, on doit être capable de connaître le fonctionnement du module. Cette partie n'est consultée que dans le cas d'une modification nécessaire du module (extension, correction d'un défaut).

Pour chaque Fonction, interne au module ou exportée :

On décrit ici la fonction du module d'une manière succincte et non ambiguë, en indiquant son domaine de validité contextuelle (ainsi, avant un write, on doit avoir réalisé un open en mode écriture), de domaine (telle variable doit avoir une valeur comprise entre 10 et 20).

On explicite la fonction en expliquant le rôle de chacun des paramètres, les conditions émises dessus (par exemple, en C, lors du passage d'un pointeur, on déclare qui doit réserver la place mémoire). On donne également les variables globales impactées, comment et pourquoi.

IV. Documents de référence, Glossaire, Index, Annexes (facultatif)

Confère remarques générales.

6.3. Conseils

7. Dimensionnement

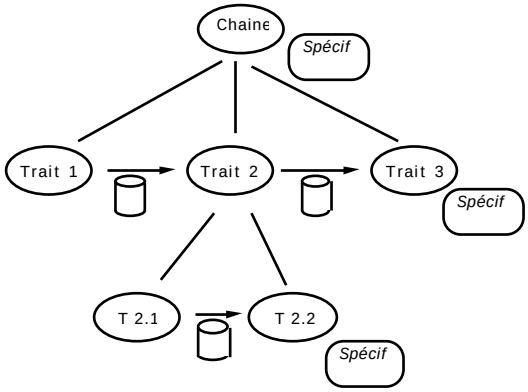
7.1. Présentation

Ce document s'appuie directement sur les parties Performance et Capacités du document de spécifications externes. Il doit montrer comment l'architecture proposée dans les spécifications matérielles et internes répond aux performances et capacités attendues.

L'étude des performances ne peut pas se faire toujours de manière exacte : elle peut donc s'appuyer sur des méthodes statistiques.

Lors de la vie d'un projet, plusieurs architectures peuvent être proposées. On ne retiendra dans ce document que la dernière, mais on rappellera dans la partie "Architecture" du sous système donné l'ancienne solution, ses performances et le pourquoi de son rejet, afin d'avoir une trace de l'évolution de l'architecture en terme de choix de performance.

La décomposition du système en sous système n'est pas un problème simple. Le mieux est de raisonner par chaîne de traitement. Ce qui donne par exemple :



Exemple de calcul de dimensionnement

Dans cet exemple, on décompose la chaîne en trois sous-traitements. Pour rendre l'exemple plus réaliste, imaginons qu'il s'agisse de l'impression d'une image sur une imprimante. Le traitement 1 consiste à rechercher l'image dans le système (sur disque optique par exemple), le traitement 2 à lire et décompresser l'image, tout en créant un fichier prêt à l'impression. Le traitement 3 va permettre alors l'impression réelle de ce fichier sur une imprimante. Le traitement 2 est décomposé en deux sous-traitements, qui consiste à lire l'image et à la décompresser.

On isole ensuite les traitements qui sont asynchrones de ceux qui ne le sont pas (par exemple, les trois traitements sont asynchrones, alors que les sous-traitements ne le sont pas), et ensuite les spécifications de chaque élément : les spécifications de la chaîne totale sont ceux demandés par le client. Il est important alors d'avoir :

- le nombre d'élément demandés (par exemple, 1200 demandes/jour),
- la durée du traitement permis (par exemple, 6 heures d'impression/jour sur l'imprimante),
- la bande passante donnée par le client (de 6h du matin à midi).

Certains de ces éléments vont être reportés seulement sur des sous-éléments : par exemple, on n'as droit qu'à 6 heures d'impression sur l'élément 3, qui gère l'imprimante. Cela permet donc de préparer des images à l'avance si le débit de fourniture de ces images par le traitement 2 est supérieur au débit de l'élément 3.

Ensuite, on recherche les éléments connus ; on peut ainsi savoir que le temps d'impression dans le traitement 3 est de 6 s/images, et qu'une image est un fichier raster de 2 Mg.

On connait aussi par exemple ou par test que le temps de décompression par image est de l'ordre de 4s (ce temps là variant suivant le type d'image, il faut soit s'appuyer sur des données statistiques, soit prendre le cas le pire), et que le temps de lecture est de l'ordre de 2 s. On peut ainsi déduire le temps de traitement numéro 2 de à 6 s (2 s + 4 s - traitement synchrone).

Par la suite, on pourra ainsi calculer toutes les différentes capacités intermédiaires à utiliser pour arriver dans les spécifications demandées (ou ne pas y arriver : dans ce cas, on isolera l'élément le plus faible).

Dans le cas d'un calcul d'occupation réseau et d'occupation machine, il est plus simple de raisonner par pourcentage d'occupation.

Il faut pour cela connaître :

- la capacité de la ressource critique (par exemple, 150 Ko/s de transfert pour un réseau),
- les différents éléments qui utilisent cette ressource,
- pour chaque élément, la taille des éléments, leur fréquence d'envoi et la bande passante d'utilisation.

Ainsi, si une chaîne de traitement utilise le réseau de 6h à 12 h, à raison d'un transfert de 50 Ko toutes les 10 s, l'occupation du réseau entre 6h et 12h sera de :

150 Ko/s <=> 1500 Ko/10s; 50 Ko/10 s occupe donc 50/1500 * 100 =3,33 % entre 6h et 12h.

Si, pour une bande passante donnée on dépasse 100 %, la ressource sera alors sur-employée, ce qui causera un ralentissement de toutes les chaînes qui utilisent ce réseau. Si sur une journée on dépasse les 100 %, le système ne sera alors pas utilisable !

Remarque : il est plus simple de raisonner en pourcentage, car comment prévoir que la chaîne 1 va produire de manière régulière toutes les 10 secondes, à savoir à 6h, 6h 10s, 6h20s...Il se peut toujours qu'à un instant immédiat T le réseau soit surchargé, et qu'une chaîne soit obligée d'attendre;, par contre, si le réseau n'est pas surchargé en moyenne, ce temps de retard ne devrait pas excéder la demande totale divisée par le débit : ainsi, si toutes les chaînes produisent 50 Ko + 100 Ko+...=X Ko, l'attente maximale, qui correspond au fait que toutes les chaînes produisent à l'instant T leurs requêtes, est égale à X / Debit s (soit X/150 Ko s dans le cas du réseau spécifié ci dessus).

7.2. Plan type

I. Introduction

Confère remarques générales.

II. Principe général

On donne le principe général du calcul : par statistiques, par étude de sous-système (dans ces cas, les préciser), l'environnement et les hypothèses générales (réseau ne comprenant que l'application, ou réseau extérieur branché et n'occupant que 10 % des ressources...).

Les méthodes de calculs doivent être données.

III. Rappel des spécifications

On donne un rappel des spécifications demandées par le client, ainsi qu'un **rappel des aptitudes Qualité** demandées (efficacité, extensibilité...).

Les taux généraux de temps d'accès sont donnés.

Il est préférable de faire un renvoi au document de spécifications externes, partie "Performances".

IV. Données et hypothèses

Cette partie présente toutes les données et hypothèses prises pour le calcul de performances. Elles forment la base des calculs.

IV.1 Données

Ce sont des constantes : délai de numérisation, débit d'un réseau, la taille d'une image A0 en raster, etc...

IV.2 Hypothèses

Ce sont les hypothèses que l'on formule. Elles doivent être justifiées. Ce sont par exemple la valeur d'une image A0 compressée TRIF, le taux de débit d'un réseau (en particulier s'il est utilisé par d'autres applications), etc...

V. Tests effectués

Ils sont la base des calculs. On indique pour chaque test son nom, son but (ce qu'il doit calculer), son environnement (combien de machines présentes sur le réseau en cas de calcul de charge...), ses résultats (accompagnés de graphique si possible), ainsi qu'une interprétation des résultats.

VI. Rappel de l'architecture

On donne l'architecture aussi bien matérielle que logicielle proposée, et sur laquelle portent les calculs. On pourra monter, dans les calculs détaillés, d'autres architectures et les raisons du refus de leur choix.. Il ne s'agit ici que d'un rappel général, présentant les sous-systèmes s'il y en a.

Il est préférable de faire une description simple, et d'effectuer un renvoi au document de spécifications matériels.

VII. Calculs détaillés

Pour **chaque calcul** "**x**" de performance (pour les jukebox, les imprimantes...), on donne les sous points suivants. Le but est de démontrer que les spécifications de performances demandées par le client sont acquises, et de proposer des améliorations de performances, soit en modifiant l'architecture générale (création d'un sous-réseau, mise en place d'une nouvelle machine), soit en modifiant l'architecture logicielle (déplacement d'un traitement sur une autre machine, nouvel algorithme...). :

VII.x.1 Spécifications

On rappelle les spécifications demandées en terme de performances, de capacité et de qualité.

VII.x.2 Architecture

On donne l'architecture retenue, et sur laquelle va porter le calcul. On présente également les autres architectures (s'il y a) des autres matériels.

VII."x3 Méthode de calcul

On donne la méthode de calcul.

VII.x.4 Hypothèses

On donne les hypothèses du calcul (occupation du réseau, place disque, hypothèses mathématiques...) . Cette partie doit être très synthétique, et on doit pouvoir retrouver les renseignements le plus vite possible. Il est donc préférable de bien soigner la présentation des valeurs utilisées.

VII.x.5 Calcul

On effectue le calcul pour l'architecture retenue, et pour les autres matériels en études. Comme les hypothèses varient souvent (changement de matériel, nouvelles annonces constructeurs, tests en charge effectués...), il est nécessaire de pouvoir modifier cette partie facilement.

VII.x.6 Conclusion

On donne une conclusion, présentant si l'objectif est atteint (avec si possible le taux d'erreur) et les améliorations possibles (matérielles et logicielles). Cette partie doit être aussi synthétique que possible (présentation par tableau).

Cette partie est extrêmement importante, car à l'usage, seule elle sera lue.

VIII. Points critiques généraux

Cette partie fait un premier bilan de l'architecture retenue en terme de performances. On isole ainsi les points critiques (réseau, machine...).

IX. Améliorations

En fonction des points critiques isolés ci-dessus, on présente les améliorations possibles, si possibles chiffrées.

X. Conclusion

On présente une conclusion de l'étude de performance.

XI. Documents de référence, Glossaire, Index, Annexes (facultatif)

Confère remarques générales.

7.3. Conseils

8. Etudes annexes

CE DOCUMENT N'A FAIT L'OBJET D'AUCUNE DISCUSSION ACTUELLEMENT

8.1. Présentation

Lors d'un projet, une ou plusieurs personnes peuvent acquérir des compétences sur un sujet, sur un produit. Il faut alors propager cette connaissance, soit pour permettre aux autres membres de l'équipe d'utiliser le produit, soit pour en faire une présentation qui sera utilisée par d'autres personnes (diffusion de la veille technologique).

Dans le cas d'un produit matériel, on rédige une fiche matérielle. Dans le cas d'un produit logiciel ou d'un concept, on rédige une fiche produit.

8.2. Plan type

I. But

Ce premier paragraphe doit nous donner une première idée générale sur la fonction du produit employé.

On pourra éventuellement faire une comparaison rapide avec d'autres produits existants afin de le situer par rapport à la concurrence.

II. Terminologie

A chaque produit sophistiqué est associé un vocabulaire spécifique. On donnera donc une définition précise de chaque mot du vocabulaire employé en donnant sa traduction en anglais (puisque, en général, la documentation qui accompagne ce produit est en anglais). Tout au long du tutorial on n'hésitera donc pas à donner la traduction anglaise du mot employé en le mettant entre parenthèses par exemple. Si un mot représentant un concept important pour ce produit est représenté sous forme d'icône, on dessinera l'icône représentant ce concept.

III. Concepts (ou principes)

Dans ce paragraphe on s'efforcera de modéliser le produit afin que l'utilisateur en comprenne le principe de fonctionnement . On citera les normes respectées par ce produit

Pour chaque nouveau concept, on introduira les opérations qui s'y rapportent.Par exemple, pour le concept "composant" de NSE on donnera toutes les opérations qui se rapportent à ce concept : création du composant" ("acquiers").

"mise à jour du composant avec son père" ("réconcilier")

"création d'une révision de composant" ("préserve")

"résolution des conflits avec le fils" ("resync")

IV. Utilisation

En fonction de l'utilisation que l'on veut faire du produit (modification ou simple mise en œuvre) on expliquera la façon de procéder.

V. Performances

On donne les performances du produit. Cette partie peut être omise.

VI. Configuration

On précisera dans quel contexte il faut installer le produit et les implications éventuelles sur cet environnement. Exemple : avant d'installer le produit NSE, il faut d'abord installer le service NIS de Sun...etc

S'il s'agit d'un système matériel, on fera un schéma de ce système.

VII. Fiche technique

On donne la fiche technique du produit : fabricant, personne à contacter, dernier numéro de version... ainsi que les personnes compétentes sur le sujet à MC2.

VIII. Problèmes reconnus

On donne tous les problèmes rencontrés, et les solutions (si elles existent !).

IX. Documents de référence, Glossaire, Index, Annexes (facultatif)

Confère remarques générales.

8.3. Conseils

9. Etude de faisabilité

CE DOCUMENT N'A FAIT L'OBJET D'AUCUNE DISCUSSION ACTUELLEMENT

9.1. Présentation

9.2. Plan type

9.3. Conseils

10. Journal de bord

CE DOCUMENT N'A FAIT L'OBJET D'AUCUNE DISCUSSION ACTUELLEMENT

10.1. Présentation

10.2. Plan type

10.3. Conseils

11. Manuel d'exploitation

CE DOCUMENT EST EN COURS DE VALIDATION

11.1. Présentation

Le manuel d'exploitation est destiné à être remis au client dans le cadre du dossier d'exploitation.

Le but de ce manuel est de permettre l'exploitation du système. Ce cadre signifie qu'il doit permettre à l'exploitant de trouver la manière de faire les sauvegardes et les restaurations, les différentes configurations ainsi que les solutions a apportées en cas de problème. Ainsi, il fait référence au **Manuel de Gamme** , autre document fourni à l'exploitant dans le cadre du dossier d'exploitation.

Ce document est celui réellement utilisé par l'exploitant. Il n'est en général jamais lu en entier de manière séquentielle (ou une fois), mais l'exploitant y fait constamment appel pour connaître les solutions à apporter à un problème. Sa structure est donc très horizontale, et il est conseillé de soigner la réalisation de l'index, toujours très utilisé dans ce genre de document.

Enfin, il est conseillé de débuter sa réalisation le plus tôt possible, afin de permettre aux différents développeurs de le compléter au fur et à mesure de la réalisation du système (ce qui est particulièrement vrai pour la partie consacrée aux erreurs).

Dans certains systèmes, les différentes fonctions d'administration font l'objet d'une aide en ligne (man page d'unix). De plus, certaines procédures d'intervention sur le matériel par exemple font l'objet de gamme de la part des constructeurs du matériel donné . Il est bien sûr hors de question de recopier ces différents documents ; par contre, et c'est là un des aspects du métier d'intégrateur de MC2, toutes ces différentes interventions, si elles peuvent être faites par l'exploitant, doivent être référencées dans ce document. Ainsi l'administrateur dispose-t-il d'un document fédérateur de son exploitation, qui lui permet de régler tous ses éventuels problèmes, et qui est pour lui sa seule référence.

Donc, dans le cas de l'existence d'une explication plus fournie ailleurs, on aiguillera le lecteur vers cette documentation, en lui donnant les références complètes ainsi que le moyen d'y accéder.

11.2. Plan type

I. Introduction

Confère remarques générales.

II. Administration

Cette partie comprend toute la partie administration proprement dite; à savoir la configuration, les sauvegardes, ainsi que tout ce qui touche à la manière de fonctionnement du système (mode dégradé, réglage, release note).

II.1 Configuration

II.1.1 Description

Il ne s'agit pas de rappeler la configuration donnée dans les spécifications matérielles, mais plutôt la configuration logicielle-matérielle : quelle fonction tourne sur quelle machine, qui utilise le disque, etc... Cette description est faite par fonction Il est également inutile de rentrer trop dans les détails : une phrase du genre "la fonction d'impression utilise la machine 1, ainsi que le disque externe de cette machine afin de stocker les images en attente d'impression par l'imprimante". Des schémas sont utiles.

Les différents fichiers de configuration doivent être explicités, et les valeurs initiales de ces fichiers (fixées par MC2) doivent être données.

II.1.2 Evolution possible

Cette partie décrit les évolutions possibles de la configuration. Ces évolutions sont de deux ordres :
- celles réalisables par l'administrateur (déplacement d'une fonction de machine par exemple),
- celles qui sont des extensions du système.

Le deuxième cas n'as pas à être développé : il n'intéresse pas l'exploitant ! On donnera donc le principe d'évolution ainsi que les gammes associées à ces évolutions.

II.1.3 Impossibilité d'évolution

On donne ici les différentes impossibilités d'évolution, afin de permettre à l'exploitant de gagner du temps s'il recherche une solution non prévue (recherche d'une optimisation de configuration, ou d'une solution face à un mode dégradé non prévu). On peut ainsi indiquer qu'étant donné qu'il est impossible de placer plus de 3 cartes dans un Sparc 2, il est impossible de placer sur le même serveur 2 imprimeurs laser et 1 imprimeur Com Plotter.

II.2 Sauvegarde / restauration

Deux types de sauvegardes sont possibles : la sauvegarde du système, et la sauvegarde des données manipulées par le système. Pour chaque sauvegarde possible (les données pouvant être stockées sous plusieurs formes, donc nécessiter plusieurs commandes séparées), on donnera le moyen de sauvegarde (référence à une gamme) ainsi que le moyen de restauration, ainsi que les conséquences de cette restauration (perte des informations depuis la dernière sauvegarde), ainsi que des moyens de connaître ces conséquences (liste des images enregistrées depuis la dernière sauvegarde, permettant de reprogrammer le travail).

II.3 Gestion des données

La gestion des données doit permettre l'accès de l'administrateur aux données auxquelles il peut avoir accès. On donnera par exemple ici les procédures de création des rapports, ou les procédures permettant la destruction des bases de données locales. Comme toujours, la description des différentes fonctions énoncées ici ne peut être qu'une référence à une gamme.

II.4 Démarrage / surveillance / arrêt

Toute la procédure de démarrage du système, le moyen de le surveiller ainsi que le moyen de l'arrêter. Si le système peut se démarrer par sous système, alors on indiquera le moyen ainsi que le dépendance entre sous système.

II.5 Mode dégradé

Cette partie est très importante. Elle n'indique pas les différents modes dégradés possibles, qui sont déjà donnés dans les spécifications externes, mais le moyen de basculer le système dans un mode dégradé. Cette partie est donc organisée par ressource.

Pour chaque ressource, on indique les conséquences de la disparition de cette ressource, le principe de basculement du système, ainsi que la gamme permettant de le faire.

Par exemple, dans le cas d'une imprimante qui venait à s'arrêter, les conséquences sont par exemple le non respect de certaines performances demandées (quelles sont les nouvelles ?), ainsi que les possibles surchargements de certaines parties (telle le réseau).

Ensuite, on indique la manière de basculer toutes les impressions demandées sur cette imprimante vers une autre imprimante. On oublie naturellement pas également de décrire la manipulation inverse (sortir du mode dégradé).

II.6 Défauts répertoriés, realeases

Les différents défauts relevés dans le système et reconnus par MC2 sont notés, ainsi que les différences amenées par la livraison d'une nouvelle version.

II.7 Environnement de test

Dans certains cas, MC2 livre un environnement de test.On décrit cette environnement de test : ce qu'il permet de faire, comment peut-on le faire, quelles commandes du système principal fonctionnent sous cet environnement, etc..

III. Utilisation normale

III.1 Travail journalier / hebdomadaire

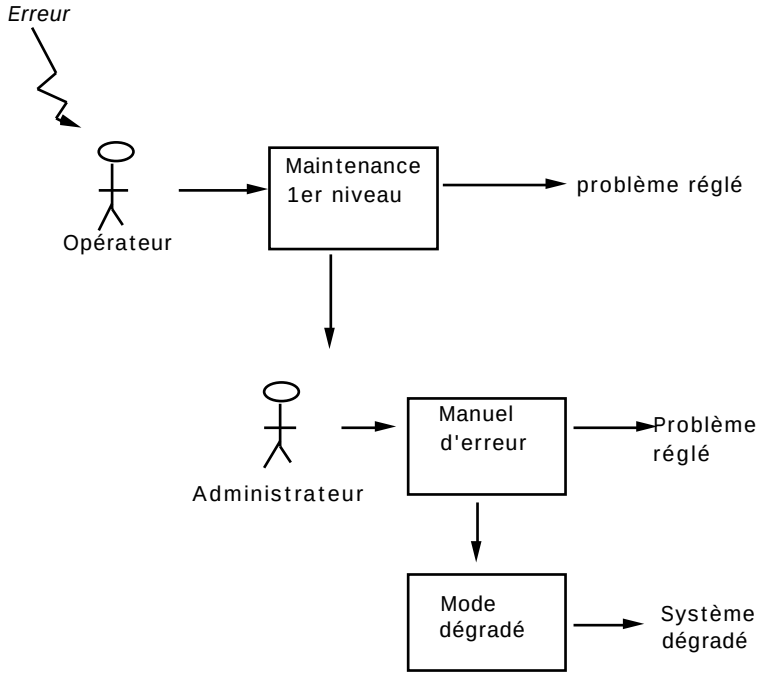
On décrit dans cette partie le travail journalier ou hebdomadaire attendu de la part de l'administrateur. Ainsi, il devra chaque jour à 6 heures récupérer une information depuis un autre service, et lancer une commande sur ce fichier, ou faire une sauvegarde tous les jours, etc...

III.2 Fonctions

On décrit chaque fonction possible de l'exploitant, à laquelle il à accès. C'est par exemple l'arrêt de la chaîne d'impression, ou le lancement d'une fonction sur des données importées, etc... On décrit la fonction ou on fait référence à une gamme.

IV. Maintenance

Cette partie permet de résoudre une erreur lorsqu'elle survient dans le système. Dans le cas général, le problème peut être réglé par l'opérateur (plus de papier dans l'imprimante, carte à fenêtre mal positionnée dans le scanner...). Si l'opérateur ne peut pas résoudre le problème, il consulte l'administrateur du système, qui fait référence au manuel d'erreur. Celui-ci lui indique alors comment résoudre l'erreur, ou lui demande de basculer en mode dégradé, en lui donnant les références du mode dégradé à mettre en oeuvre.



Résolution d'une erreur

IV.1 maintenance 1er niveau

La partie sur la maintenance 1er niveau s'adresse donc plus particulièrement aux opérateurs. Elle peut faire référence aux documentations constructeurs.

IV.2 Manuel d'erreur

Le manuel d'erreur est organisé de manière très classique : pour chaque erreur, on trouve un libellé de l'erreur, la cause de l'erreur et l'action à mettre en place pour la résoudre ou la contourner. Il est préférable que chaque erreur ait un numéro de référence (unique bien sûr).

IV.3 Approvisionnement en consommable

Cette partie ne permet pas de résoudre une erreur, mais fait partie de la maintenance préventive. On donne pour chaque composant qui en utilise la quantité de consommable nécessaire et la fréquence de réapprovisionnement en fonction des spécifications du système.

V. Documents de référence, Glossaire, Index, Annexes (facultatif)

Confère remarques générales.

11.3. Conseils

Il est préférable de commencer la rédaction de ce document dès la phase de conception (quant aux modes dégradés possibles), puis de laisser chaque développeur remplir les parties qui les concernent (manuel d'erreur).

12. Manuel d'installation

CE DOCUMENT EST EN COURS DE VALIDATION

12.1. Présentation

Le manuel d'installation fait partie du dossier d'exploitation. Il permet de connaître le moyen d'installer et de réinstaller le système. Il est surtout utile à l'équipe support, mais permet à l'exploitant de connaître les différentes procédures d'installation, nécessaires s'il possède un environnement de test et qu'il est responsable du passage des nouvelles versions de l'environnement de test vers on environnement de production.

12.2. Plan type

I. Introduction

Confère remarques générales.

II. Installation

Cette partie donne les différentes gammes d'installation pour l'installation matériel (et configuration logicielle de ce matériel), logiciels de base (i.e. produits logiciels), logicielle (i.e. logiciel développé par MC2), et des données.

Soit on peut donner la gamme dans ce document, soit on peut faire référence au manuel de gamme.

II.1 Matériel

On donne ici toutes les procédures pour installer le matériel ainsi que la configuration de ce matériel (configuration du noyau dans le cas d'Unix).

Si plusieurs installations existent suivant le type de matériel ou le type de la station (station dédiée à l'impression), on explicite alors toutes les installations différentes.

II.2 Logiciel de base

On donne tous les logiciels de base ou produits logiciels à installer, les endroits où ils doivent être installés ainsi que la gamme d'installation.

II.3 Logicielle

Il s'agit ici des logiciels développés par MC2. On donne la philosophie d'installation, ainsi que la gamme d'installation, ou les gammes si l'installation est différente suivant le type de machine.

II.4 Données

Dans le cas où des données doivent être initialisées, les gammes d'initialisations doivent être données. Les fichiers de configuration ne sont pas détaillés ici, mais dans le manuel d'exploitation.

Par contre, si de nouvelles données peuvent être installées, la gamme d'installation est donnée.

III. Réinstallation

On donne dans cette partie toutes les différences entre une installation et une réinstallation. Si la différence est importante, on reprendra alors la structure matériel/logiciel de base/logicielle/donnée du paragraphe précédent.

IV. Environnement de test

Cette partie est utile dans le cas où un environnement de test existe. On donne ici l'installation de l'environnement de test, et non son exploitation, qui est détaillée dans le manuel d'exploitation. On donne les gammes d'installation de cet environnement, puis dans le cas où l'environnement de test sert au test des nouvelles versions, le passage des logiciels de l'environnement de test vers l'environnement de production.

IV.1 Installation

On donne ici la gamme d'installation de l'environnement de test.

IV.1.1 Passage vers l'environnement de production

Le passage des logiciels de l'environnement de test vers l'environnement de production est décrit.

V. Documents de référence, Glossaire, Index, Annexes (facultatif)

Confère remarques générales.

12.3. Conseils

13. Manuel de gamme

CE DOCUMENT EST EN COURS DE VALIDATION

13.1. Présentation

Ce manuel est le regroupement de l'ensemble des gammes séquence d'actions) utilisé par l'exploitant. Il fait partie du dossier d'exploitation donné au client.

Ce manuel donne des gammes aussi bien logicielles que d'intervention sur le matériel. A priori, les gammes d'intervention sont données par le constructeur, mais dans le cas où elles sont incomplètes, inexistantes ou fausses, MC2 peut être amenée à les redéfinir. Elles figurent dans ce cas dans ce manuel.

Ce manuel doit être considéré comme une base, et non comme un manuel que l'on lirait de manière séquentielle. Néanmoins, une certaine classification des gammes est nécessaire; celles-ci sont donc classées suivant 3 grandes classes :

- matérielle, il s'agit de toutes les interventions de maintenance sur le matériel,
- installation, il est alors question de toutes les gammes d'installation du système, aussi bien pour le matériel, les logiciels de base ou le système,
- exploitation, elle regroupe toutes les gammes utiles à l'exploitant.

13.2. Plan type

I. Introduction

Confère remarques générales.

II. Gammes matérielle

On donne toutes les gammes relatives au matériel, telles par exemple la maintenance 1er niveau

III. Gamme d'installation

Les gammes relatives à l'installation (du matériel au système) sont regroupées ici.

IV. Gammes d'exploitation

Les gammes utiles à l'exploitation sont données.

V. Documents de référence, Glossaire, Index, Annexes (facultatif)

Confère remarques générales.

13.3. Conseils

14. Manuel de maintenance

CE DOCUMENT N'A FAIT L'OBJET D'AUCUNE DISCUSSION ACTUELLEMENT

14.1. Présentation

3 niveaux :

1- vérifications de base
2- télémaintenance
3- approfondie

1er niveau : tel bouton s'allume, pourquoi ? Comment ouvrir l'imprimante ? Comment relancer le spooler ?

14.2. Plan type

I. Introduction

Confère remarques générales.

II. Définition du site matériel

Réseau, machine, login

III. Définition du site logique

Quel login, quels outils...

IV. Localisation des indicateurs

Ou sont les traces ? Comment les obtenir ? Ou sont les erreurs ?

V. Outils et méthodes

Comment discuter avec un composant (comme le jukebox) ? Comment avoir accès aux sauvegardes ?

VI. Documents de référence, Glossaire, Index, Annexes (facultatif)

Confère remarques générales.

14.3. Conseils

15. Manuel de référence

CE DOCUMENT N'A FAIT L'OBJET D'AUCUNE DISCUSSION ACTUELLEMENT

15.1. Présentation

15.2. Plan type

15.3. Conseils

16. Manuel utilisateur

CE DOCUMENT N'A FAIT L'OBJET D'AUCUNE DISCUSSION ACTUELLEMENT

16.1. Présentation

16.2. Plan type

16.3. Conseils

17. Planning de réalisation

CE DOCUMENT EST EN COURS DE VALIDATION

17.1. Présentation

Ce document peut être écrit pour la phase de réalisation, et concerne le planning de développement, qui doit donc tenir compte de la stratégie de test élaborée dans le plan de test.

17.2. Plan type

Confère remarques générales.

I. Introduction

Confère remarques générales

II. Méthode de calcul

On donne la méthode de calcul du planning de réalisation. En général, la méthode découle directement de la méthode de tests donnée dans le plan de test.

III. Présentation du planning

On présente le planning de réalisation, avec ses étapes marquantes et ses points de contrôles. On justifie l'allocation des ressources sur chacun des postes.

IV. Gestion des retards

On précise dans le paragraphe la méthode de gestion des retards : soit des ressources supplémentaires sont disponibles, et dans quelles mesures, soit des fonctionnalités seront réduites. On donne les différents points de contrôles des retards, et les mesures à prendre sur chacun des points.

V. Suivi de planning

Pour chacun des points de contrôle défini, on effectue un bilan présenté ici (une partie par point de contrôle). Le bilan doit comporter, pour chacune des tâches en cours, une évaluation, puis une évaluation générale (si certaines parties sont en retard, d'autres sont peut-être en avance), afin de déterminer l'état d'avancement du projet.

On donne ensuite la nouvelle répartition des ressources si besoin est.

VI. Bilan de réalisation

A la fin de la réalisation, l'équipe dresse un bilan donnant l'écart entre les ressources prévues et celles utilisées, ainsi qu'une interprétation, le but de cette partie étant de parfaire le savoir-faire de MC2 dans la gestion de projet.

VII. Documents de référence, Glossaire, Index, Annexes (facultatif)

Confère remarques générales.

17.3. Conseils

18. Plan de tests

CE DOCUMENT N'A FAIT L'OBJET D'AUCUNE DISCUSSION ACTUELLEMENT

18.1. Présentation

Le plan de test donne la démarche de test. Il influence directement la réalisation : ainsi, si l'on choisit une méthode de test par le bas (tests des modules de plus bas niveau d'abord), la réalisation des modules de bas niveau sera prioritaire.

Il regroupe tous les principes et règles de tests, ainsi que la démarche de contrôle. Il doit être nominatif et responsabilisant pour les membres de l'équipe de développement.

18.2. Plan type

I. Introduction

Confère remarques générales.

II. Démarche générale

On donne la démarche générale de tests, les différents niveaux (si l'on exécute des tests unitaires par exemple), scénarios (qui test, qui rédige le plan de test, qui le remet à jour, que se passe t-il si le test est négatif, etc...).

Ces différents principes, qui peuvent être différents pour chaque niveau de test, seront détaillés par la suite

La granularité des objets à tester est aussi définie : Ainsi, dans le cas d'un système multi serveurs, on peut se donner comme objet les modules et les serveurs : on menera d'abord une campagne de tests unitaire/intégration/généraux par serveur (test des objets module), puis dans un deuxième temps, une campagne d'intégration/généraux pour le système complet (test des objets serveurs).

III. Tests généraux

III.1. Démarche

On donne dans cette partie les principes et conseil de tests

III.2. Jeux de test

Dans le cas des tests généraux, le jeu de tests recouvre au moins la recette. Il est inutile alors de redonner les différents tests, par contre les résultats de ces tests peuvent être indiqués ici.

III.3. Traitement d'exception

Les différentes possibilités d'exception sont données et testées (arrêt d'une machine, d'un serveur...).

IV. Tests des serveurs

On peut se donner ce niveau de détail dans le cas d'identification des objets Serveurs à tester.

IV.1. Démarche

Idem

IV.2. Jeux de test

Idem. On traitera entre autre les dépendances des serveurs entre eux.

IV.3. Traitement d'exception

Arrêt brutal du serveur, d'un autre serveur, test en mode dégradé : le serveur A utilise les ressources de B, qui n'est pas présent. Comment réagit A ? Dans le cas où B est soudainement lancé ? soudainement arrêté ?

V. Tests d'intégration

V.1. Démarche

On déterminera ici les personnes responsables de quels modules, et celles responsables des tests d'intégration, ainsi que le niveau d'erreur accepté pour que le module soit validé (aucune erreur grave, moins de 10 % d'erreurs gênantes).

V.2. Jeux de tests

Les différents jeux de test sont donnés, avec leur niveau de gravité (grave, gênant, mineur). Un contrôle de la conformité aux spécifications en matière d'interface devrait être obligatoire.

V.3. Traitement d'exception

Appel d'une procédure avec un paramètre hors domaine sémantique (par exemple, si on définit un niveau de priorité compris entre 1 et 10, et que le module est appelé avec 11).

A ce niveau, le test des traitements d'exception permet de tester la solidité du logiciel.

VI. Tests unitaire

Il est impossible et fastidieux de donner tous les tests unitaires effectués. On se bornera dans ce chapitre à donner les principes de tests.

VII. Conclusion

Cette conclusion est a rédiger à la fin de la campagne de test. Elle doit identifier les défauts mis en évidence par la campagne de test (points positifs), et ceux que la campagne n'a pas permis de tester (points négatifs).

VIII. Documents de référence, Glossaire, Index, Annexes (facultatif)

Confère remarques générales.

18.3. Conseils

19. Planning général

CE DOCUMENT EST EN COURS DE VALIDATION

19.1. Présentation

Le planning définit l'ensemble des actions qui vont être menées pour optimiser et gérer les délais, les coûts et les ressources pour la réalisation du produit final.. Il doit permettre quatre visions :

- la vision du client, intéressé par les dates de livraison, et le contenu de ces livraisons,
- le contrôle du financier, pour lequel le coût et l'occupation des ressources sont primordiaux,
- permettre au directeur technique, qui alloue des ressources au projet, d'optimiser le partage des ressources entre projets,
- le contrôle et la répartition des ressources à l'intérieur du projet par le chef de projet.

De plus, le planning général doit permettre de calculer les impacts dus à un retard de livraison extérieur, ou à un retard interne sur les ressources et dates du projet.

Il doit également permettre la rédaction d'un bilan du projet, car un autre aspect de la gestion de projet est l'apprentissage des coûts. Pour cela, la rédaction du bilan, comparant les coûts estimés et les coûts réels, est nécessaire. Cela suppose bien évidemment un outil de suivi des coûts réels, suivant la granularité de suivi choisie (si la granularité est la tâche, alors il faut rédiger des comptes rendus par tâche, et aller plus loin que le sigle REA de MC2). Le bilan intéresse le directeur technique, le consultant technique, le chef de groupe (responsable du chiffrage des affaires) et le financier.

Le planning proprement dit peut, pour des raisons de confidentialité MC2<->client, ou Direction<->équipe de développement, être séparé du plan de développement qui devient confidentiel. Les documents de planification sont importants pour la communication, aussi bien pour l'information de l'équipe de développement que pour la communication vers l'extérieur de cette équipe (client, direction financière et générale). Le planning général peut être décomposé en plusieurs document : un planning prévisionnel, un document de suivi de projet et un bilan.

Le suivi de projet se fait en réactualisant ce document. Une période d'un mois sur un projet d'un an semble bonne.

	Matériel			Humain			
	Archi.	Planning	Coût	Tâche	Coût	Planning	Allocation
Financier			X				
Chef de projet	X	X	X	X	X	X	X
Client	X	X	X				
Achat	X	X	X				
Direc. Techn.			X		X	X	X
Commercial	X	X	X				

X : est intéressé par.

19.2. Plan type

I. Introduction

Confère remarques générales.

II. Référentiel du projet

Cette partie présente le référentiel, les grandes lignes sur lesquelles MC2 et le client se sont entendus.

II.1. Echéancier

L'échéancier peut être phasé donné sous forme de phase. Si c'est le cas, on donne les fonctions à livrer, phase par phase.

II.2. Planning de facturation

Il s'agit ici du planning de facturation. On donne également les moteurs de la délivrance des paiements (par exemple, l'acceptation du système par le client en SAT permet une délivrance de 20 % de la somme totale)

II.3. Planning d'approvisionnement

Planning d'approvisionnement en matériel.

III. Ressources

On donne ici les ressources prévues, matériel, logiciel et humaines.

III.1. Hypothèses

Les hypothèses de délivrance de ressource (un spécialiste X Windows sera disponible du 10 janvier au 10 Février, quelqu'un suivra une formation Oracle avant février, le logiciel ScriptWorks sera délivré en mars).

III.2. Calendrier

Le calendrier des ressources est donné, aussi bien humain que matériel et logiciel. Ce calendrier sera la base du lissage du Pert par rapport aux ressources.

IV. Cycle de vie

Le cycle de vie retenu est présenté : cela implique d'abord un découpage du projet en élément plus fin. Il est recommandé à ce niveau le découpage en phase, tel qu'il est présenté dans le Plan de Développement Système. A partir de là, on peut choisir un cycle de vie **séquentiel pure** : une phase doit se terminer et être validée pour passer à la suivante, ou un cycle de vie avec **chevauchement** : à un moment, deux phases peuvent se chevaucher (certains travaux de la phase suivantes commencent alors que la phase n'est pas validée).

De plus, en cas d'erreur, on peut décider d'autoriser le **retour arrière** : une fois la correction effectuée, on corrige les impacts éventuels sur les produits générés après, tout en continuant le développement de la phase en cours. **Sans retour arrière** , il faut reprendre entièrement le développement à la phase mise en défaut : toutes les productions des autres phases sont alors détruites, et les différentes étapes de validation sont à repasser entièrement.

	Séquentiel	Chevauchement
Retour		
Sans		

Différents types de cycle de vie

Le cycle de vie choisi doit être justifié. Pour MC2, le meilleur choix semble être le cycle de vie avec chevauchement et retour arrière, mais c'est le plus dure à gérer.

Remarque : le projet peut être géré en étape, chaque étape regroupant les phases classiques de spécifications, de conception, de réalisation... Dans ce cas, les différentes étapes, et leur ordonnancement (chevauchement, séquentiel...) doivent être présentés et justifiés.

V. Méthode de calcul

On donne ici la méthode de calcul (Pert par exemple) ainsi que les outils de calcul et/ou de suivi (Pert avec Mac Projet, Excel...).

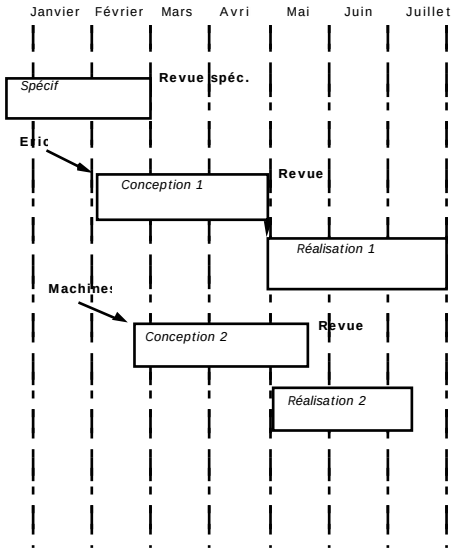
VI. Etapes

On présente le planning prévisionnel, calculé à partir de la méthode et des ressources. On peut faire également un Gant, permettant de voir ou sont les problèmes de ressources critiques.

Si on raisonne par phase et par Pert, il faut isoler tous les enchaînements, et toutes les pré-conditions aux phases (autres phases ou prestations extérieures, comme par exemple la fourniture de produit). On isole alors le chemin critique. Le suivi de projet se fera par étude du réseau.

Cette partie intéresse l'équipe de développement et la direction générale, pour l'allocation de ressource. Une période de contrôle (15 jours, 3 semaines, 1 mois) est choisie pour le suivi de projet.

Cette partie permet de présenter un planning sous la forme :



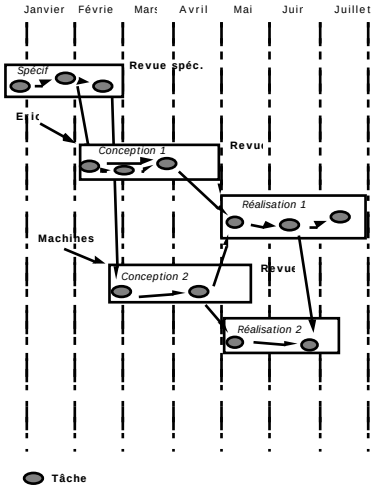
Planning par phase

A ce niveau, les liens avec les phases ne sont pas dévoilés. Ils le seront à l'étape suivante. On ne donne que les liens par rapport aux ressources externes.

En fait, ce planning doit permettre déjà de travailler, mais ne favorise pas le suivi de projet.

VII. Evaluation détaillée

Le schéma précédent est repris, et on positionne dessus l'ensemble des tâches.



Planning détaillé

A ce niveau, l'utilisation d'un outil comme Mac Project est utile. Dans ce planning, on doit voir apparaître de façon claire et simple :

- la liste des tâches,
- les liens entre ces tâches,
- les hypothèses de début des tâche,
- les ressources allouées aux tâches,
- les différents points de contrôle associés aux phases,
- les étapes du projet.

VIII. Vue du client

La vue du client est le planning vu par lui. Il comporte donc les livraisons en matériel, en fonctions, les recettes, etc...

Il peut être vu sous la forme d'un dessin.

IX. Vue des personnels

Il s'agit ici des personnels du projets. Pour eux, ils veulent connaître quand ils vont travailler sur telle ou telle tâche. On peut faire un Gant pour chaque personne.

Ce planning doit permettre également aux personnes de gérer leurs temps, en connaissant les différentes livraisons qu'elles doivent produire.

X. Définition des points de contrôle

On donne un certain nombre de points de contrôle, avec une estimation des retards. Ce sont soit des points de contrôle de livraison, soit de fourniture que l'équipe doit produire (a-t-on fini de spécifier ce module ?).

On essaye de calculer les impacts d'éventuels retards.

XI. Bilan

Cette partie peut être rédigée au fur et à mesure, ou à la fin. Elle présente le bilan de la gestion de projet : débordement (avec la raison) et impact, tâche bien ou mal maîtrisée, etc...

XII. Documents de référence, Glossaire, Index, Annexes (facultatif)

Confère remarques générales.

19.3. Conseils

20. Plan Qualité

20.1. Présentation

20.2. Plan type

20.3. Conseils

21. Spécifications du matériel

CE DOCUMENT EST EN COURS DE VALIDATION

21.1. Présentation

Le document de spécification matérielle est ébauché par le consultant technique, qui défini, avec le client les grandes fonctionnalités du système et donc également les machines qui seront dans le système.

Ce document à donc pour but de présenter la configuration matérielle, ainsi que les différents postes de travail qui en découlent. On peut pousser la granularité du document en spécifiant les flots de données au travers du système, mais on risque alors d'être redondant avec le document de spécification externes.

21.2. Plan type

I. Introduction

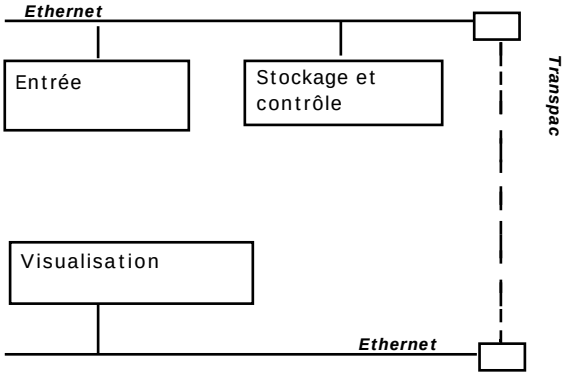
Confère remarques générales.

II. Présentation générale

On présente l'ensemble du matériel par groupe de sous-systèmes. Un sous système peut être isolé soit par une fonctionnalité, soit par une localisation géographique.

Exemple : On peut ainsi isoler le sous système de contrôle qualité, de visualisation, de stockage (par fonctionnalité) soit le sous système basé à Paris et le sous système basé à Grenoble, chacun des sous-système pouvant effectuer les fonctions de visualisation et de contrôle qualité. En plus, le sous système de Grenoble est responsable du stockage.

On décrit les liens (réseau, transpac, bande) entre les différents sous-systèmes. On nomme chaque sous-système, et on présente sa fonctionnalité. Un schéma est obligatoire, comme :

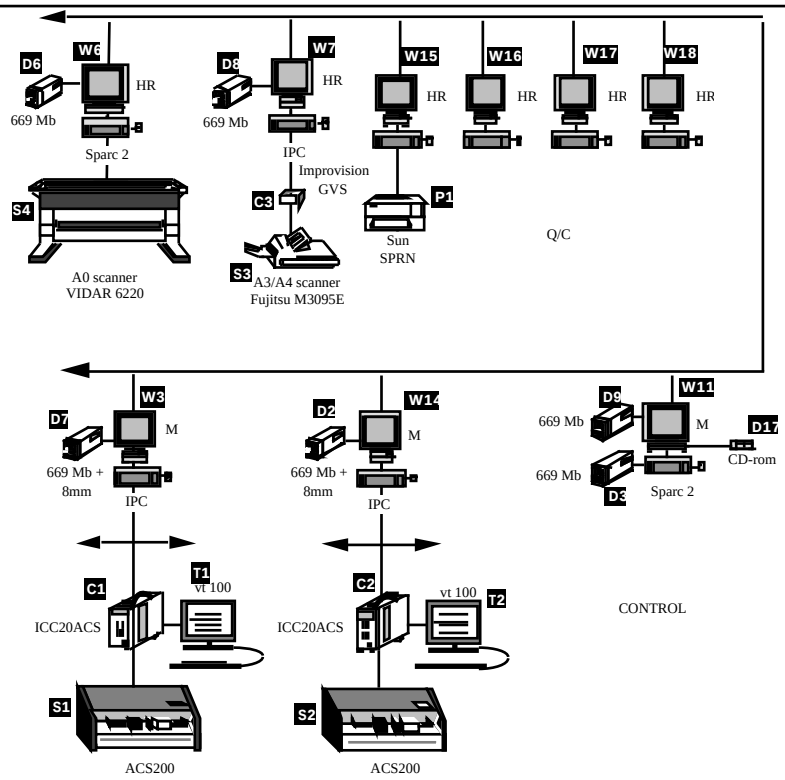


Exemple de système

III. Présentation des sous systèmes

Chaque sous système est détaillé, en reprenant les icônes de la bibliothèque MC2 définissant les différents matériels.

On détaille chaque élément, et on donne sa référence ainsi que son **rôle dans l'architecture générale** . On nomme ainsi chaque élément, pour permettre ensuite d'y faire référence tout au long du projet (par exemple, le serveur imprimeur pour un SUN frontal de plusieurs imprimantes).



Exemple de sous système

IV. Inventaire

On donne la liste du matériel.

Nom	Libellé	Date

Inventaire du matériel

V. Historique des changements

On donne l'historique et l'évolution des spécifications matérielles (pourquoi tel matériel à été rajouté...)

VI. Validation de la pré visite

Chaque installation est préparée par une pré-visite des services matériels de MC2. On donne le compte rendu dans cette partie.

VII. Fiches matérielles

On donne, pour chaque matériel présenté, la fiche matériel :
- l'icône MC2,
- les différentes performances,
- les conditions d'utilisations (températures...) et une description rapide.

Ces fiches ne sont pas à réécrire, mais a chercher au secrétariat.

VIII. Documents de référence, Glossaire, Index, Annexes (facultatif)

Confère remarques générales.

21.3. Conseils

22. Spécifications externes

22.1. Présentation

L'objectif du dossier de spécifications externes est de formaliser les besoins exprimés par le client. Il décrit donc les fonctions et les données à gérer, sans décrire la solution informatique finale. Il peut comporter des détails techniques (algorithmes, bases de données...) si c'est contractuel.

Ce document est une vue du système par le client.

Ce travail doit être suivi et validé par le client, et également par la personne qui va être responsable du système, et a donc une idée précise de son futur fonctionnement (ce n'est pas forcément la même personne !).

Plusieurs vues (pour le client) sont possibles :
- le management ne s'intéressera qu'à une description générale du système,
- les techniciens auront plutôt une approche verticale (des points précis : la description de la base de données, ou la sécurité du système...),
- le maître d'ouvrage lira le document complètement.

Le plan type de ce document permet ces approches.

22.2. Plan type

I. Introduction

Confère remarques générales.

II. Contexte

Ce chapitre définit en quelques mots les objectifs et les hypothèses du produit à réaliser.

Cette partie est très importante, et permet d'éviter certains malentendus lors de la recette !

II.1. Objectifs

Ce paragraphe reprend la partie du cahier des charges sur le sujet. Il doit contenir les objectifs (ce que le produits devra faire), les limites (ce qu'il ne devra pas faire) ainsi qu'une description des clients.

Si plusieurs phases apparaissent dans le projet, la description et l'objectif de chaque phase doit apparaître.

II.2. Hypothèses

Ce paragraphe contient l'inventaire des conditions à satisfaire pour le bon déroulement du projet vis-à-vis des clients, des concepteurs, des fournisseurs et des partenaires intervenant dans le projet.

Il permet ainsi de préciser :

deux fonctions ne pouvant pas marcher l'une sans l'autre),
- les aides éventuelles de partenaires extérieurs,
- les **matériels, logiciels et documentations devant être présents** pour l'accomplissement de tâches spécifiques, les dates de disponibilités et les taux d'utilisation..

Bref, tous les objets attendus ni du client, ni de MC2.

Cette partie est la protection de MC2. Il ne faut surtout pas la négliger.

Attention : il s'agit ici des hypothèses du contexte du projet, donc générale. Une partie spéciale regroupe les contraintes de développement, et les contraintes d'exploitation. Le but de ce paragraphe est de fixer les hypothèses, du style "SUN est responsable du développement de tel fonctionnalité, et doit être réalisé avant telle date". Seules les choses importantes doivent apparaître ici.

III. Description générale

Ce chapitre fournit la description générale du projet afin de faciliter la compréhension de la description détaillée des fonctions.

C'est le premier niveau de lecture du client. En lisant cette partie, il doit pouvoir vérifier le niveau de compréhension de MC2 et l'orientation de la solution. Il y a de fortes chance que le client ne lise que cette partie, alors que le responsable du système (le maître d'ouvrage tel qu'il est défini dans le plan Qualité) lise, lui, l'ensemble du document.

III.1. Présentation du problème

Cette partie décrit de la manière la plus complète qui soit, le problème ayant amené la demande de réalisation du système.Il ne faut pas faire de "philosophie" (décrire un problème qui n'en est pas forcement un pour le client), et rester cohérent avec la proposition commerciale. Cette partie doit refléter le problème du client., et ne dois jamais dépasser sa vision. De plus, il faut éviter de faire le lien avec d'autres objets généraux du client, et ne parler que des objets à traiter.

La lecture d'au moins cette partie par le Consultant Technique est nécessaire.

Ce paragraphe permet d'éviter beaucoup de malentendu, il doit donc être le plus complet possible.

III.2. Présentation des objets manipulés

On donnera l'ensemble des données gérées par le client, ainsi que leurs liens, leurs types, leurs tailles. On donnera également une fréquence de création par jour, ou par année (par exemple, 1100 cartes à fenêtres sont produites par jour...). De plus, la vie des données doit être décrite (le brevet A devient, après la phase de validation, un brevet B qui doit être diffusé à tous les abonnés).

Il s'agit d'un résumé du chapitre sur les données. Il s'agit d'être cohérent avec la proposition commerciale (vocabulaire...).

III.3. Intégration dans les systèmes existants

Ce paragraphe donne la situation du projet par rapport à l'existant et aux autres produits ou projets dont il dépend. Il décrit ainsi les interactions avec l'environnement dans lequel s'exécutera le logiciel. Il faut rester générale, et parler d'environnement au sens large.

Au moins un schéma est obligatoire.

III.4. Fonctions du système

Cette partie décrit les fonctions du système, de manière générale. Les différents postes de traitement peuvent apparaître, ainsi que les données traitées. Il s'agit ici que d'une description générale.

Attention : cette partie doit être cohérente avec la description détaillée.

III.5. Caractéristiques des utilisateurs

Ce chapitre décrit le contexte utilisateur (population concernée, exemple : secrétaire, ingénieur, commercial, ouvrier, ...) et en particulier les connaissances de chacun, son expérience, son niveau informatique et technique etc...

Ces informations auront une influence considérable sur la qualité requise de l'interface homme-machine. Par exemple, une population d'utilisateurs occasionnels amènera à élaborer un système d'aide très important.

Naturellement, c'est au client d'identifier ses utilisateurs.

III.6. Engagement du client

Tous les engagements du client doivent apparaître :
- le niveau de formation des utilisateurs,
- les contraintes physiques d'environnement (sur les matériels livrés),
- l'environnement d'exploitation (débit réseau, temps d'exploitation...),
- les jeux de tests et les plannings qu'il doit fournir,
- les validations éventuelles,

En général, ce qu'il fournit et ce qu'il assure.

Cette partie est la protection de MC2 par rapport au client. Il convient de la préciser soigneusement.

III.7. Contraintes générales

Ce paragraphe décrit les contraintes pouvant restreindre le développement et l'exploitation du logiciel.. MC2 aura tendance à restreindre les contraintes de développement, et à bien préciser les contraintes d'exploitation (version de système...). En fait, cette partie regroupe tout ce que le client impose.

1. Contraintes de développement

Logiciels utilisés

Outils logiciels utilisés pour le développement (exemple : langage de programmation, utilitaires spéciaux, compilateurs, traitements de textes, etc...).

Indiquer leurs disponibilités, leurs références (sans oublier les numéros de versions) et la documentation associée.

Matériels utilisés

Configuration informatique utilisée pour le développement. Composantes et disponibilité du matériel (exemple : type de calculateur, taille mémoire, périphériques d'impression et de sauvegarde).

Méthodes et contraintes diverses

Méthodes utilisées et organisation retenue en fonction de contraintes diverses pour le développement :

- gestion des équipes de développement,
- disponibilité et gestion de la documentation technique,
- disponibilité des sous-ensembles logiciels (versions, sauvegardes, tests, etc,..)
- gestion des sous-ensembles matériels et de l'environnement sur sites de développement,
- options techniques fondamentales discutées avec le client pour le développement du logiciel (emploi d'experts, de méthodes, etc...).

2. Contraintes d'exploitation

Logiciels utilisée

Description des logiciels utilisés pour l'exploitation. (Exemple : système d'exploitation, langages, progiciels, etc... sans oublier les numéros de versions).

Matériels utilisés

Détail de la configuration matérielle nécessaire à l'exploitation du logiciel. (Exemple : type de calculateur, taille mémoire, périphériques, etc ...).

Environnement utilisé

Contraintes sur l'environnement (salles sans poussière, sol stable pour les jukebox...).

Méthodes et contraintes diverses

Méthodes utilisées et organisation retenue pour l'exploitation du logiciel.

- gestion des équipes d'exploitation (affectation des responsabilités, planning d'exploitation, etc...),
- disponibilité et gestion de la documentation technique,
- disponibilité et gestion des sous-ensembles logiciels (versions, sauvegardes, tests, résultats d'exploitation, etc...),
- disponibilité et gestion des sous-ensembles matériels et de l'environnement sur sites d'exploitations,
- options techniques fondamentales discutées avec le client pour le développement du logiciel (emploi d'experts, de méthodes, etc,...).

IV. Description des objets manipulés

IV.1. Externe

Les objets sont toutes les informations qui seront amenées à être manipulées par le systèmes, au sens des données vues par le client (et non des données internes qui servent à la gestion propre du système !).

On donnera :
- la terminologie,
- le cycle de vie (et ses étapes),
- la structure,
- la vue qu'en a l'utilisateur,
- les règles d'exploitation,
- le modèle conceptuel des données.

Le volume des informations peut être évoqués ici, mais il doit se trouver dans la partie Capacité.

Des schémas peuvent présenter de manière plus légère et plus précise les définitions.

IV.2. Vision du système

Parmi toutes les données présentées, toutes ne seront peut être pas manipulées par le système, ou sous des aspects plus simple. Cette partie montre les données traitées par le système (sous-ensemble des données ci dessous).

Attention : on reste ici aux données du client !

V. Description détaillée

Ce chapitre contient toutes les informations nécessaires à la conception. C'est le Quoi du problème. Il fournit une description détaillée de chaque unité fonctionnelle identifiée dans la description générale, mais sans préjuger de la manière dont elle sera réalisée.

V.1. Spécifications fonctionnelles

Ce paragraphe décrit les traitements effectués pour chaque fonction. Ces traitements sont classés dans un ordre déterminé (exemple : chronologique, alphabétique, hiérarchique, etc...).

Un schéma général du système doit être présent, donnant tous les flots de données.

Si des familles de fonctions sont dégagées, elles doivent apparaître, avec leurs restrictions (restrictions d'accès, entre elles...).

1. Fonctions

Pour chaque fonction N

Les informations à spécifier pour chaque fonction sont détaillées ci-dessous :

Description de la fonction

La rédaction d'un texte concis et clair permet de donner, du point de vue de l'utilisateur, la description des traitements effectués pour cette fonction et les changements qui en résultent (tables de décisions, graphes d'états, etc...).

Appel et condition d'emploi

Ce paragraphe décrit les procédures de mise en oeuvre et d'arrêt. Il définit la façon dont la fonction est activée et éventuellement sur quel poste, quel machine.

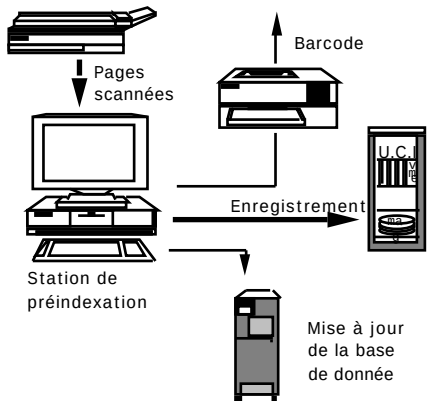
Si une fonction ne s'exécute qu'à un endroit du système (sur une seule machine par exemple), ou pendant un temps déterminé, cette information doit apparaître ici..

Données en entrée et en sortie

On donne la description des objets manipulés (défini dans le chapitre précédant) qu'il faut fournir à la fonction. Cela inclut le type et la structure, le format, la taille, le volume et la fréquence, la plage de validité et la cohérence ...

On procède de la même manière pour les objets en sortie de la fonction. La sortie des données peut intervenir sur le terminal,sur imprimante, dans un des groupes logiques de données, dans les bases de données ou dans une autre fonction. Les contraintes doivent être spécifiées.

On précise la manière dont la fonction va s'exécuter, quels transferts seront effectués (et quelles données seront transférer), et les postes de travail et machine mises en oeuvre, comme par exemple :



Exemple de "data flows"

Un dessin est obligatoire pour chaque dataflow, indiquant tous les objets impactés.

Contrôles et conditions d'erreur

On donne une description des causes de fonctionnement anormal, des messages d'erreurs générés et des mécanismes de reprise associés, une description des contrôles de forme et des contrôles de fond associés : erreur externes (informations erronées, telle une carte mal perforée), erreur du matériel, ou erreur interne).

Les traitement d'exceptions, logistique de traitement des anomalies, mode dégradé de la fonction sont précisés.

2. contrainte d'exécution

En général, un certain nombre de fonctions peuvent être exécutée en même temps,n et d'autres sont en exclusion. Il faut alors les lister pour éviter que le client demande l'exécution de plusieurs tâches en même temps.

3. Procédures exceptionnelles

On décrit toutes les procédures exceptionnelles, telles les procédures d'urgences qui sont destinés à arrêter proprement et rapidement le système alors qu'il se trouve par exemple dans l'exécution d'une fonction longue. Le comportement du système est décrit (existe-t-il ce genre de procédure, laissent-elle le système dans un état cohérent...).

4. Classes d'erreurs

Les différents types d'erreurs sont isolés. On peut trouver par exemple les erreurs causées par les calculs internes (division par 0), les erreurs utilisateurs (mauvaise formulation des commandes), ou des erreurs d'environnement (pas assez de mémoire, erreurs disques...).

Pour chaque type, on donne la **politique de traitement** du système, les méthodes de résolution et d'archivage de l'erreur (s'il y en a une).

Dans les cas d'erreur, il ne faut pas oublier les états d'incohérence, dus par exemple à une coupure de courant (le problème viendra peut être de la perte des valeurs des variables dans la machine au moment de la coupure), ou, plus grave, les pertes statiques comme la perte d'un disque optique. Les moyens de récupération sont à donner dans la partie "Sûreté" des spécifications opérationnelles.

Si un outil d'erreur est utilisé, le donner.

5. Contrôlabilité

Description des procédures d'arrêt et de reprise, des outils permettant de suivre le fonctionnement des traitements (traces et mesures de fonctionnement).

V.2. Spécifications d'interface

Les spécifications d'interfaces apparaissent parfois dans le contrat. Il s'agit d'être cohérent..

Attention, on décrit ici toutes les interfaces entre le système et l'extérieur : cela peut être du matériel, des logiciels existant, ou les opérateurs.

1. Interfaces matériel/système

Ce paragraphe décrit les relations entre le matériel et le logiciel par l'intermédiaire des données, commandes, comptes-rendus, documentation existante, etc...

Il s'agit ici du matériel externe au système, donc fournit par le client.

2. Interfaces logiciel/système

Ce paragraphe décrit la structure des échanges avec d'autres logiciels ou systèmes informatiques.

3. Interfaces homme/système

Ce paragraphe définit sous quelle forme ont lieu les échanges entre l'utilisateur et le logiciel.

Il décrit les principes et les moyens retenus pour la saisie des actions et des données de l'utilisateur (exemples de principes : menus déroulants ou arborescents, presses boutons, etc... Exemple de moyens : souris, scanner, tablette à digitaliser, etc...) et pour la présentation des informations sur l'écran et sur les états. C'est la partie visible du système, qui va déterminer l'à priori de l'utilisateur.

On peut faire un renvoi aux contraintes imposée par le client si celui-ci à exprimer de fortes demandes à ce niveau. On indique également l'existence d'une maquette s'il à été prévue d'en faire une, et les systèmes d'interface utilisés tel MOTIF ou Windows.

V.3. Spécifications opérationnelles

1. Propriété

Description des facteurs de qualité prépondérants pour le projet. Ces facteurs sont listés, définis séparément, puis regroupés en classes de besoins. Ces facteurs pourront être par exemple : l'indépendance vis à vis de l'environnement matériel et logiciel (portabilité), la facilité de traduction des écrans, textes et messages du logiciel (internationalisation), la modularité, la précision des calculs, etc... En annexe du Manuel Qualité, on trouvera les aptitudes et propriétés définis par L'AFCIQ.

La liste des standards à respecter est données ici.

On décrit également les exigences à satisfaire en matière de convivialité du logiciel (dialogue homme/machine). Il s'agit ici des qualités ergonomiques du logiciel : facilité d'apprentissage, système d'aide en ligne, mode novice/expert, multi-langue, cohérence des termes employés, homogénéité des commandes.

2. Sécurité, intégrité et sûreté

Description des exigences à satisfaire par le logiciel en matière de sécurité, de confidentialité (contrôles d'accès, droits de lecture et d'écriture, etc...), sur les traitements dégradés, etc...

Cette partie décrit les opérations à faire pour d'une part assurer la cohérence de la base en cas de problème (sûreté), et les procédures (automatique ? manuelle mais forcée par le système ?) qui doivent être exécutée pour retrouver un état de cohérence (sécurité).

Rappel :

Sécurité : tout élément perdu pourra être récupéré. La sécurité intervient APRES un incident.
Sûreté : on prévoit les erreurs possibles et on limite au maximum la possibilité qu'elles se produisent. La sûreté consiste à faire en sorte que l'incident ne se produise pas.
Intégrité : l'intégrité consiste à vérifier, à assurer qu'après un incident, le système reparte sur des bases (données, fonctions) cohérentes.

V.4. Dimensionnement

On donne dans ce chapitre toutes les demandes énoncés par le client en matière de dimensionnement. La justification et l'étude du système résultant sera dans le document Dimensionnement. Mais il vaut mieux déjà préparer la justification du système.

1. Performance

On donne les performances attendues et demandées, en tant que vitesse d'exécution, temps de réponse, fiabilité en milieux particuliers (exemple : bruits, interférences, etc...), occupation mémoire, nombre de sortie attendue, etc...

Description des performances requises pour le logiciel en termes quantitatifs. (Il vaut mieux dire "60 %" que "la majorité").

On donnera les contraintes sur :
- les temps de réponse,
- la fréquence d'utilisation,
- le temps d'indisponibilité.

Pour cela, on se donnera des documents types, que l'on justifiera. par exemple, on peut dire qu'un document A0 occupe 9 MO à 200 dpi, et occupe dans 90 % des cas 300 Ko en compressé Groupe IV, et entre 300 Ko et 2 MO dans les 10 **%** restant., qu'un réseau Ethernet débite à 80 Ko/s dans 100 % des cas. Ces contraintes sont très importantes, car elles induisent la configuration matérielle et logiciel, donc le coût du système. Normalement, tous ces calculs sont effectués par le consultant technique.

On peut écrire un diagramme des temps sur la journée (perte) pour visionner les marges et le chemin critique pour remplir l'objectif de production. Lors de la réalisation, on verra ainsi les tâches critiques en temps.

Plusieurs diagramme peuvent être écrit (diagramme pour une consultation de document, pour une indexation...)

2. Capacité

Cette partie regroupe toutes les capacités attendues :
- Volumes des objets a stocker,
- mais aussi le taux de requête attendues,
- le nombre de stations qui doivent pouvoir se connecter à un serveur,
- les débits attendus...

V.5. Documentation

Description de la documentation désirée par le client, avec son format, ses règles, et son mode de rédaction (traitement de texte unique).

Cette documentation contiendra bien sûr le dossier de spécifications du logiciel, les dossiers de conception, etc...

On donne ici la liste des documents à produire, avec le plan type (soit celui de MC2, soit celui choisi par le client). Pour chaque document, on indique le nombre d'exemplaire remis au client : soit 0 (le client ne recoit pas le document car il est sensible pour MC2, par contre il sait qu'il existe, ce qui est une preuve de Qualité pour lui), soit 1 ou plus.

VI. Documents de référence, Glossaire, Index, Annexes (facultatif)

Confère remarques générales.

22.3. Conseils

23. Spécifications internes

23.1. Présentation

Les objectifs du document de spécification interne sont :

- permettre le développement du système,
- expliquer comment a été réalisé le système (justification),
- permettre la recherche des modules unitaires, pour permette la factorisation et la réutilisabilité,
- faciliter la maintenance.

Il permet le développement car d'une part il présente l'architecture globale retenue, architecture qui peut donc être contrôlée par d'autres personnes, d'autre part parce qu'il sert de référence à l'équipe de développement. Il offre en plus l'avantage à l'équipe de développement d'améliorer sa connaissance du système.

Vis à vis du client, celui-ci est assuré qu'une étape réelle de conception à été menée, puis validée par plusieurs personnes. Cette démarche augmente donc la Qualité du développement.

Il doit de plus faciliter la maintenance, car il présente les différents composants du logiciel : une erreur peut donc être plus facilement située en lisant la description générale du système.

Toute décision prise dans ce document doit être justifiée.

Ce document est donc particulièrement stratégique, et il subira certainement de nombreuses variations de mise à jour. Le plus simple pour sa gestion est non pas d'effectuer la mise à jour APRES la modification du logiciel, mais AVANT, lorsque les différentes parties désirant une évolution sont présentes. Le document est alors modifié, et devient la référence : il ne reste plus pour chaque partie qu'à s'aligner dessus. Cette méthode présente en outre l'avantage de se protéger contre les "changeurs d'interface", race d'informaticien qui ont tendance à changer souvent les interfaces de leurs modules "pour des raisons de simplicité", et en omettant de vous prévenir. Avec ce système, une référence existe.

Remarque : ce document et le document de conception détaillée peuvent être fondu en un seul. En effet, il ne s'agit que d'un niveau de granularité entre les deux. Pour de petites équipes (moins de 4 personnes), il est préférable de ne pas rédiger le deuxième document, et de ne pas descendre trop bas la granularité du document. En effet, l'économie de temps est inférieur dans ce cas à l'effort de rédaction.

23.2. Plan type

I. Introduction

Confère remarques générales.

II. Contexte de développement

Ce paragraphe contient la description des éléments autour desquels s'effectuera le développement. Cela comprend :

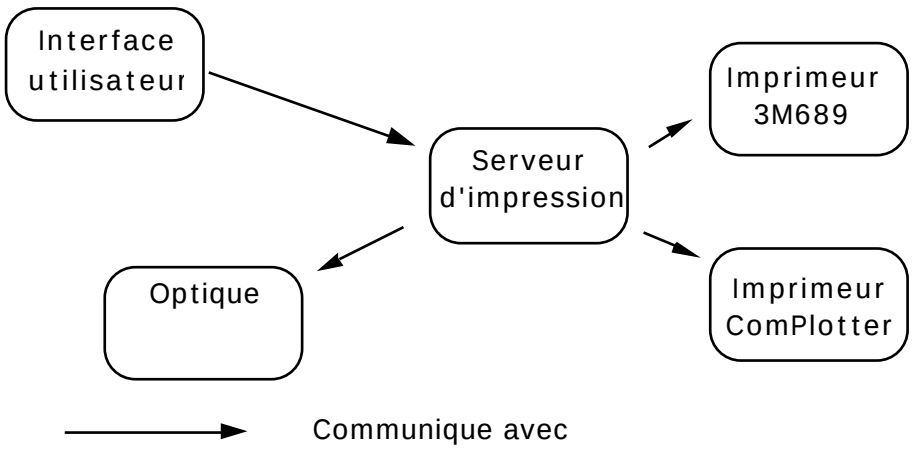
- la configuration informatique de développement (matériel système d'exploitation, ...),
- le(s) langage(s) informatique(s),
- l'ensemble des outils logiciels nécessaires au programme (compilateur, éditeur de lien, déboggeur, "make",...)
- le système de gestion de base de données,
- les algorithmes fondamentaux,
- les bibliothèques (briques de base MC2) réutilisé.

III. Description générale

III.1. Découpage en sous système

Ce paragraphe contient la description générale et la justification de la solution informatique retenue. Le système est décomposé en sous système, et chacun est décrit. Le but de cette partie est de rester générale, pour avoir une vue d'ensemble du système.

Un schéma est obligatoire comme par exemple :



Exemple de description générale

Dans cet exemple, les différents sous-ensembles sont donnés(Serveur d'impression, Optique...), ainsi que les différents liens (qui communique avec qui).

III.2. Description de chaque sous-ensemble

Pour chacun des sous-ensembles, on donne le rôle, les hypothèses de fonctionnement, ses actions avec les autres sous-ensemble. Ainsi, on donnera pour le Serveur d'impression son rôle (gérer les files d'attente, permettre du re-routage vers un autre serveur d'impression si tous les imprimeurs sont en panne, etc...

III.3. Données communes

On donne les données communes à l'ensemble du système, s'il y en a.

III.4. Protocole de communication

Le protocole de communication est défini. On en donnera le principe, quitte à donner la bibliothèque utilisée.

III.5. Workflow

On décrit le workflow (méthode de travail du système) général. On sera amené à dire : la référence est demandée par l'utilisateur via l'interface utilisateur. Cet ordre est donné au serveur d'impression. Ensuite, celui-ci recherche la référence sur la partie optique, et récupère un fichier à imprimer. et gère les différents ordres dans une file d'attente. Dès qu'un imprimeur du bon type se libère, le serveur d'impression lui donne un nouveau fichier à imprimer. Des schémas sont obligatoires.

III.6. DataFlow

L'éclairage du système se fait cette fois du point de vue des données. On utilise ici les données définies dans les spécifications externes, afin de faire un lien. Des schémas sont obligatoires, s'ils ne sont pas redondants avec le workflow.

III.7. Politiques générales

Les différentes politiques générales sont isolées. Ce peut être par exemple la manière de gérer les traces, ou de donner des informations à l'utilisateur. Ou le moyen d'accéder au protocole de communication. Ce peut être aussi les algorithmes généraux à utiliser.

III.8. Consignes de tests

Les consignes de tests sont rappelées (elles doivent être dans le document Plan de Tests).

III.9. Liste des sous-ensembles livrés

Il s'agit ici des sous-ensemble dont le code source est livré. En effet, certains sous-ensemble peuvent être des briques de base de MC2, dont les sources ne sont pas livrées au client. Dans le cas où un sous-ensemble est partiellement livré, on ne l'indique pas ici, mais on donnera la liste des sous parties livrées dans la description détaillée du sous-ensemble.

III.10. Implémentation par machine

Dans le cas où un sous-ensemble est particulier à une machine (pour des raisons d'implémentation technique de performance ou de capacité, voire de demande client), on l'indique ici.

Ce peut être par exemple : les imprimeurs doivent se trouver sur les machines physiquement reliées aux imprimantes.

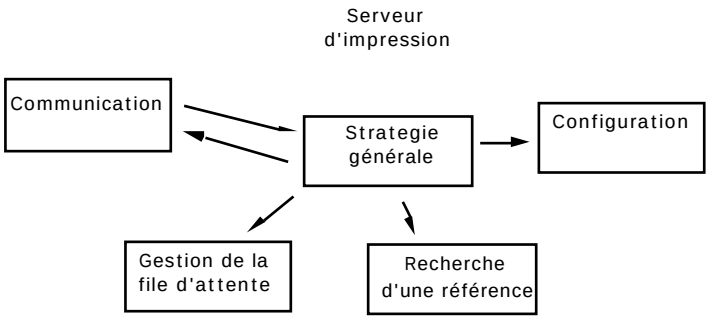
IV. Description détaillée

Pour chaque sous-ensemble, on donne les mêmes renseignements que pour le cas général, mais cette fois localement au sous-ensemble. Le niveau de granularité finale de la description est choisi par le rédacteur du document. Ce peut être par exemple le module ou la fonction.

Quelque soit le niveau, le document de conception détaillée permet de reprendre et d'approfondir la description du système.

IV.1. Découpage en sous partie

Un schéma est obligatoire.



Exemple de description détaillée

IV.2. Description de chaque sous partie

Pour chacun des sous-ensembles, on donne le rôle, les hypothèses de fonctionnement, ses actions avec les autres parties. On isolera les parties qui sont partagées avec d'autres sous-ensemble (la partie communication par exemple), et on pourra donner les algorithmes généraux de chaque partie.

IV.3. Données communes

On donne les données communes au sous-ensemble, s'il y en a.

IV.4. Protocole de communication

Le protocole de communication est défini. On en donnera le principe. Dans le cas le plus simple, il s'agit d'un appel de procédure.

IV.5. Workflow

On décrit le workflow (méthode de travail). Des schémas sont obligatoires.

IV.6. DataFlow

L'éclairage se fait cette fois du point de vue des données. On utilise ici les données définies dans les spécifications externes, mais aussi d'autres éléments que l'on peut définir. Des schémas sont obligatoires, s'ils ne sont pas redondants avec le workflow.

IV.7. Politiques

Les différentes politiques du sous-ensemble sont isolées, si elles ne font pas partie des politiques générales au système. Ce peut être par exemple la manière de gérer les traces, ou de donner des informations à l'utilisateur. Ou le moyen d'accéder au protocole de communication. Ce peut être aussi les algorithmes généraux à utiliser.

IV.8. Consignes de tests

Les consignes de tests sont rappelées (elles doivent être dans le document Plan de Tests).

IV.9. Liste des sous-ensembles livrés

Il s'agit ici des sous-ensemble dont le code source est livré. En effet, certains sous-ensemble peuvent être des briques de base de MC2, dont les sources ne sont pas livrées au client. Dans le cas où un sous-ensemble est partiellement livré, on l'indique ici.

IV.10. Implémentation par machine

Si certaines parties du sous système sont dépendantes d'une machine, l'indiquer ici.

IV.11. Bibliothèque

On termine la description par la liste de l'ensemble des bibliothèques, avec leurs rôles, utilisé dans le système.

V. Données

Rappel : on appelle objet manipulé les données manipulées par le système, et décrites dans les spécifications externes.

Les données dont on parle ici doivent bien sûr être au moins les objets manipuler, mais peuvent être enrichies par des données internes au système, servant à le faire fonctionner (statut des objets manipulés, état du traitement...), ou par exemple la gestion des files d'attentes d'une impression.

V.1. Structures logiques

On donne les structures et liens logiques sur les données : telle donnée possède un lien vers un autre ensemble de données, etc...

V.2. Implémentation physique

Dans cette partie, on décrit l'implémentation physique de chacune des données. Ce peut être sous la forme de fichier, de table interne, ou de base Oracle/Ingres.

V.3. Liste des rubriques

Pour chaque ensemble de données, on donne la liste des rubriques (nom, type, condition d'existence si elle en possède une).

V.4. Diagramme des états

Certaines données changent au cours de leur vie dans le système (par exemple, une requête d'impression est crée, en cours de composition, en cours d'impression, en erreur, finie...). Pour chacun de ces cycles de vie, on donne un diagramme d'état, avec un schéma. La méthode ainsi que le formalisme du schéma est laissée au rédacteur.

V.5. Vues utilisateurs

On donne les vues que les utilisateurs/administrateur ou tout autre utilisateur au sens large du système, peut voir des données.

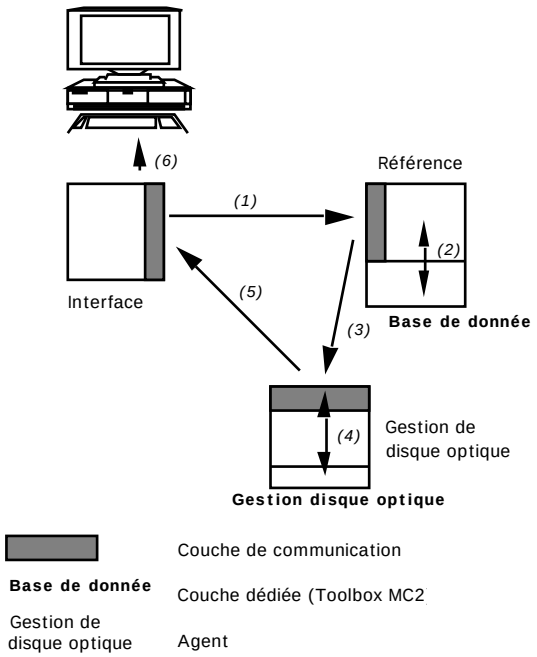
V.6. Erreurs

Les différents cas d'erreur, données sous forme de liste, est recensé.

VI. Description par fonction

Le but de cette partie est de valider l'architecture retenue en présentant le cheminement des fonctions données dans les spécifications externes dans les différents sous-ensembles .

Ce qui donnerait par exemple, pour une consultation de donnée :



Exemple de description par appel

Lors d'une consultation, l'agent "Interface" demande à l'agent "référence" où se trouve l'image demandée (1). Celui-ci communique avec la base de donnée pour rechercher la référence (2). S'il le trouve, il identifie le disque optique, donc la machine où se trouve l'image. Il demande alors à l'agent de gestion de disque optique d'envoyer l'image à l'agent interface (3). Celui-ci lit sur le disque, au travers de la couche de gestion du disque, l'image (4) et la renvoie à l'agent interface (5). Il peut alors l'afficher sur la console (6).

VII. Documents de référence, Glossaire, Index, Annexes (facultatif)

Confère remarques générales.

23.3. Conseils

Glossaire

Autorisation de livraison	Autorisation remise à MC2 par le client après la recette usine (FAT)
C.T. :	Consultant Technique
Cahier d'extension :	Cahier remis à la partie commerciale de MC2, montrant les évolutions qui ont été vues par l'équipe de développement au cours de la réalisation du système.
Cahier de recette :	Il détermine les différents tests que le client veut faire subir au système pour démontrer son bon fonctionnement par rapport à celui prévu dans les spécifications externes.
Chargé Qualité :	Le chargé Qualité est un membre de l'équipe de développement, chargé de l'application et du contrôle de la Qualité.
Consultant Technique :	Personne effectuant l'analyse des besoins du client. Son rôle principal intervient durant les phases d'avant vente et de spécification.
D.T.	Direction Technique
Défaut	Non conformité avec les spécifications de départ. Confère Erreur
Dimensionnement	Ce document démontre le dimensionnement du système, aussi bien en terme de performance que de capacité.
Direction Technique	Ensemble des personnes réalisant les systèmes.
Dossier d'exploitation :	Ensemble des documents remis à l'exploitant pour lui permettre d'assurer l'exploitation de son système. Le dossier d'exploitation comprend le manuel d'installation, le manuel d'exploitation, le manuel de gamme ainsi que tous les manuels fournis par les constructeurs.
Erreur :	Spécification de départ fausse. Confère Défaut

Etude de faisabilité :	Etude permettant de vérifier que les choix techniques pris sont bien réalisables.
FAT :	Factory Acceptance Test (recette usine)
Gamme :	Séquence d'action. Les gammes sont référencées dans le manuel de gamme.
IAT :	Internal Acceptance Test (recette interne)
Maître d'ouvrage:	Personne nommée par le client, et qui le représente. Il est chargé de la définition des besoins, des recettes, et d'assurer le suivi du projet côté client.
Dossier d'exploitation :	(ou Dossier d'exploitation). Il décrit toutes les opérations utiles à l'exploitation du système.
Manuel d'utilisation :	Il est remis à l'utilisateur final. Il peut y en avoir plusieurs (un par sous système par exemple). Il doit avoir un rôle pédagogique certain.
Manuel de Maintenance :	Manuel remis à la maintenance, et qui indique ou sont les arborescences de développement, les exécutables, les fichiers de configuration...
Manuel de Référence :	Manuel remis à l'administrateur du système, donnant différentes indications, le plus détaillées possible, sur le système.
MC2 :	Marketing Consultant Conseil
Outil :	Vu au sens du client. Un outil peut être une étude, un système ou un service (maintenance).
Planning de réalisation :	Planning découpant la phase de réalisation en tâche.
Plan de Tests :	Méthode prise pour mener la campagne de tests. Les différents jeux de tests sont également présents dans le document.
Planning Général :	Il regroupe l'ensemble des phases du cycle de vie, avec les dates importantes.
Plan Qualité :	Définit par rapport au Manuel d'Assurance Qualité Système, il donne les méthodes, assurances et contrôles effectués pour assurer la qualité de la réalisation finale.
Projet	Opération effectuée par MC2 en vu de réaliser un outil
SAT:	Site Acceptance Test (recette site)
Spécifications externes	Description externe du système, vu par le client.
Spécifications internes :	Document regroupant la décomposition du système en composant plus fin, permettant la réalisation informatique. Les liens entre les composants sont également décrits.

Spécifications du Matériel : Elles contiennent la liste du matériel, mais également les fiches techniques de chaque matériel, la description de l'emplacement des matériels chez le client et la recette matérielle, montrant le bon arrivage.

Système : Ensemble cohérent d'appareils matériels et logiciel formant un outil pour la gestion électronique de document.

Procès verbal d'acceptation : Acceptation par le client du système, après la recette site (SAT). Cette acceptation comporte souvent des réserves de temps.

Index

AFCIQ 67
approvisionnement 50
aptitude 20; 67
Bertrand MEYER 81
briques de base 70; 72; 73
C.T. 76
Cahier d'extension 76
Cahier de recette 76
Capacités 18
Chargé Qualité 76
Com Plotter 10
Com Plotter 10; 30
communication 73
consommable 33
Consultant Technique 61; 76
contrat 66
D.T. 76
Défaut 76
Dimensionnement 76
Direction Technique 76
donnée 31
dossier d'exploitation 29; 34; 37; 76; 77
échéancier 49
Effel 16
Environnement de test 31; 35
erreur 33; 74; 76
Etude de faisabilité 11; 76
Excel 51
exploitant 76
exploitation 76
facturation 50
FAT 77
fiche matériel 58
fonction 74
gamme 29; 30; 31; 32; 34; 37; 77
IAT 77
icônes 56
IFF3 10
Ingres 74
Mac Projet 51
maître d'ouvrage 59; 61; 77
man page 29
Manuel D'Assurance Qualité Système ii; 77
manuel d'erreur 33
manuel d'exploitation 76
manuel d'installation 76
Manuel d'utilisation 77
Manuel de Gamme 29; 35; 76; 77
Manuel de Maintenance 77
Manuel de Référence 77
Manuel Qualité 67
maquette 66
MC2 77
Mode dégradé 31
MOTIF 66
Object oriented software construction 81

objet manipulé 74
objets 81
Oracle 74
Outil 77
PC 10
Performance 18; 58
Pert 50; 51
phase 60
Plan de Développement Système ii; 1; 50
Plan de Tests 72; 73; 77
Plan Qualité 77
Planning de réalisation 77
Planning Général 77
pré visite 58
Procès verbal d’acceptation 78
Projet 77
proposition commerciale 61
propriété 67
recette interne 77
recette matérielle 77
recette site 77; 78
recette usine 76; 77
réinstallation 35
Ressources 50
SAT 77
spécifications du matériel 10; 77
spécifications externes 31; 74; 77
Spécifications internes 77
statistique 20
Système 78
test 73
unix 29
utilisateur 77
Wicks & Wilson 10
Windows 66
Workflow 73

Fiche bibliographique

AFCIQ 1990
AFCIQ, "Guide pour la rédaction d'un Plan D'assurance Qualité" - AFCIQ/SL/90/005/-1

AFCIQ 1990
AFCIQ ,"Guide pour la rédaction d'un plan de développement logiciel" - AFCIQ/SL/90/007/-1

MEYER 1988
Bertrand MEYER, "Object oriented software construction" Prentice-Hall Internationnal (UK)Ltd,Hemel Hempstead 1988; traduction française : "Conception et programmation par objets" InterEdition Paris 1991

MEYER 1990
Bertrand MEYER "The new culture of software development", Journal of object oriented programming vol 3, no 4 nov/dec 1990

Formulaire Qualité

Ce formulaire est à utiliser pour toute remarque vis à vis de ce document. Il est à remettre au groupe chargé de la définition de la Qualité au sein de la D.T.

Qualité MC2

Remarque

Demandeur :
Document Manuel Qualité PDS Plans types Autre :
Localisation :

Description / amélioration souhaitée

Fiche de suivi :

	Date	Remarques
Notification		Par
Discuté le		
Status		Accepté Refusé Déféré au
Modifié		Version document :

Remarques / modifications :

suite au verso si besoin