

THESIS / THÈSE

MASTER EN SCIENCES INFORMATIQUES

Étude et génération d'une documentation minimale pour accompagner la méthode d'analyse SDM, étude de cas

Detaille, Muriel

Award date:
1992

Awarding institution:
Universite de Namur

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Facultés Universitaires Notre-Dame de la Paix, Namur
Institut d'Informatique - année académique 1991-1992

Etude et génération d'une documentation
minimale pour accompagner la méthode
d'analyse SDM, étude de cas

Muriel DETAILLE

Mémoire présenté en vue de l'obtention du grade de Licencié et Maître en Informatique

REMERCIEMENTS

Mes remerciements s'adressent tout d'abord à Messieurs Roland LESUISSE et Christian GRAAS, tous deux promoteurs de ce mémoire.

Je tiens à les remercier pour leur disponibilité et leurs conseils fructueux.

Ce travail d'enquête n'aurait pu être mené à bien sans la complicité de tous les informaticiens de la CGER. Qu'ils en soient tous et toutes remerciés.

Je tiens également à remercier tous les membres du groupe de Support au Développement pour toutes les informations qu'ils m'ont fournies.

Finalement, un merci tout particulier à Monsieur Michaël BODAR pour sa disponibilité et son soutien moral.

INTRODUCTION

1. <u>PRESENTATION DE LA CGER</u>	1
1.1 LA CGER	1
1.2 LE DEPARTEMENT INFORMATIQUE	2
a. PRINCIPALES ETAPES DE L'INFORMATISATION DE LA CGER	2
b. STRUCTURE DU DEPARTEMENT INFORMATIQUE	3
2. <u>OBJECTIF DU MEMOIRE ET INTERET DU SUJET</u>	6

CHAPITRE 1 : LA METHODE SDM, LES MODELES, METHODES ET OUTILS D'AIDE AU DEVELOPPEMENT

1.1 SDM : UNE METHODE DE DEVELOPPEMENT DE LOGICIELS	8
1.1.1 CARACTERISTIQUES D'UN PROJET DE DEVELOPPEMENT	9
1.1.2 SDM A LA CGER	10
1.1.3 PRESENTATION DE LA METHODE SDM	12
1.2 LES MODELES D'AIDE AU DEVELOPPEMENT	15
1.2.1. LA MODELISATION CONCEPTUELLE DES DONNEES	15
1.2.2. LE MODELE DYNAMIQUE DES DONNEES	16
1.2.3. LE MODELE LOGIQUE DES DONNEES	16
1.2.4. LE DIAGRAMME DE FLUX DE DONNEES	16
1.2.5. LA STRUCTURE HIERARCHIQUE DES ACTIVITES ET DES TACHES	17
1.2.6. LE DIAGRAMME "HIERARCHICAL INPUT-PROCESS-OUTPUT"	18
1.3 LES METHODES D'AIDE AU DEVELOPPEMENT	18
1.3.1. LA METHODE "JACKSON STRUCTURED PROGRAMMING"	18
1.3.2. LA METHODE "INTERPROGRAM FUNCTION POINT ANALYSIS"	19
1.4 LES OUTILS D'AIDE AU DEVELOPPEMENT	20
1.4.1. LE DICTIONNAIRE DE DONNEES "DATAMANAGER"	20
1.4.2. PALOMA	21

1.4.3. SUPERPROJECT EXPERT	22
1.4.4. EXCELERATOR	22

CHAPITRE 2 : CRITIQUE THEORIQUE DE LA DOCUMENTATION PROPOSEE PAR SDM

2.1 DEMARCHE POURSUIVIE ET CRITERES DE SELEC- TION	26
2.1.1 QU'EST-CE-QUE LA DOCUMENTATION ?	27
2.1.2 CRITERES DE SELECTION	30
2.2 CRITIQUE A PRIORI DE LA DOCUMENTATION	31
2.2.1 PHASE 1 : ETUDE DE L'EXISTANT ET ANALYSE CON- CEPTUELLE	31
2.2.2 PHASE 2 : ANALYSE FONCTIONNELLE DETAILLEE	44
2.2.3 PHASE 3 : ANALYSE TECHNIQUE	52
2.2.4 PROGRAMMATION ET TESTS ISOLES	60
2.2.5 REALISATION DES TESTS	65
2.2.6 CONVERSION ET MISE EN PRODUCTION	72
2.3 TABLEAU RECAPITULATIF DE LA DOCUMENTATION MINIMALE D'UN PROJET	76

CHAPITRE 3 : RESULTATS DE L'ENQUETE, DES INTERVIEWS ET INTERPRETATION

3.1 PRESENTATION DE L'ENQUÊTE	79
3.2 QUESTIONS GENERALES	80
3.3 LA REALISATION DE LA DOCUMENTATION	81
3.3.1 LA DOCUMENTATION JUGEE INDISPENSABLE	82
3.3.2 LA DOCUMENTATION JUGEE UTILE	93
3.3.3 LA DOCUMENTATION JUGEE SUPERFLUE	99
3.3.4. RESUME DE LA SITUATION ACTUELLE DE LA DOCU- MENTATION	101
3.4 L'AUTOMATISATION DE LA DOCUMENTATION	105
3.4.1 SOUHAITS D'AUTOMATISATION	105
3.4.2 DEMANDES RÉALISABLES AVEC LES OUTILS EXIS- TANTS	107
3.4.3 OBJECTIFS PRIORITAIRES DU GROUPE DE SUPPORT . . .	111

CHAPITRE 4 : LES CASE TOOLS COMME GENERATEURS DE DOCUMENTATION

4.1 L'AD/CYCLE: L'INFORMATISATION DE L'INFORMATIQUE	115
4.1.1 DEFINITION DE L'AD/CYCLE	115
4.1.2 IMPACTS AU NIVEAU DE LA DOCUMENTATION	117
4.2 AD/CYCLE A LA CGER	117
4.2.1 SITUATION ACTUELLE	117
4.2.2 PERSPECTIVES D'AVENIR	119
<u>CONCLUSIONS</u>	121
<u>BIBLIOGRAPHIE</u>	123

TABLEAUX

1	: Les catégories de projet de Mac Farlan	10
2	: Récapitulatif de la documentation de l'analyse conceptuelle	43
3	: Récapitulatif de la documentation de l'analyse fonctionnelle	51
4	: Récapitulatif de la documentation de l'analyse technique	59
5	: Récapitulatif de la documentation de programmation de tests isolés	65
6	: Récapitulatif de la documentation de tests du système	71
7	: Récapitulatif de la documentation conversion et mise en production	75
8	: Récapitulatif de la documentation minimale d'un projet	76
9	: La documentation jugée indispensable	82
10	: La documentation jugée utile	93
11	: La documentation jugée superflue	99
12	: Souhaits d'automatisation	105
13	: Demandes réalisables avec les outils existants	107

FIGURES

1	: Structure du département informatique	5
2	: Le modèle "Waterfall" du cycle de vie d'un logiciel	13
3	: Diagramme de flux d'une activité : exemple (conventions de la CGER)	17
4	: Diagramme de structuration d'un traitement : exemple	17
5	: Diagramme HIPO d'une application : exemple	18
6	: Diagramme JSP d'un programme : exemple	19
7	: Structure du dictionnaire "Datamanager"	21
8	: La méthode de Benson	39
9	: Schéma général de l'AD/Cycle	116

INTRODUCTION

Le domaine de l'informatique subit actuellement une vague de rationalisation qui se traduit par une standardisation et une automatisation accrue du processus de développement de logiciels.

C'est dans cette lignée que s'inscrivent les méthodes de développement de logiciels, les modèles d'aide au développeur ainsi que les outils tels les outils d'analyse, les générateurs de code, les langages de 4ème génération ou les dictionnaires de données.

De plus en plus, les gens de l'informatique prennent conscience du fait que la réalisation de la documentation d'un système informatique occupe une place très importante dans les processus de développement du système : si les programmes quoique bien conçus ne s'accompagnent pas d'une documentation suffisante à leur bonne compréhension et à leur utilisation, leur efficacité sera nettement amoindrie.

Il faut bien admettre que dans bon nombre de départements informatiques, la documentation est trop souvent négligée, incomplète, difficile à comprendre et trop rarement et partiellement mise à jour.

La recherche d'une solution à ce problème s'effectue dans le sens d'une standardisation accrue du contenu et du mode de présentation de la documentation ainsi que vers une gestion centralisée de celle-ci.

Cependant, les méthodes de développement proposées actuellement ne fournissent que très peu d'indications quant à la réalisation d'une documentation de qualité.

Ce mémoire s'intéresse à une méthode de développement particulière (la System Development Method) instaurée à la Caisse Générale d'Epargne et de Retraite (CGER) depuis quelques années. Il va tenter de fournir une synthèse critique de cette méthode concernant le processus de création de documents qu'elle propose tout au long du développement d'un système informatique.

Il examinera ensuite les possibilités d'automatisation de la documentation minimale d'un projet développé à la CGER, compte-tenu des besoins des développeurs et des outils disponibles.

Ce mémoire a été structuré de la manière suivante :

- Pour fixer le cadre de ce mémoire, nous présenterons tout d'abord la Caisse Générale d'Epargne et de Retraite et son département informatique ainsi que les objectifs et intérêts de ce travail.
- Le premier chapitre s'intéressera ensuite à la méthode SDM, à son application à la CGER ainsi qu'aux principaux modèles, méthodes et outils d'aide au développement présents. Cet ensemble permettra de mieux connaître l'environnement informatique de la CGER et facilitera la suite de la lecture.

- Le deuxième chapitre, après avoir défini la notion de documentation, proposera une critique à priori de la documentation proposée par SDM et permettra de définir les composants de la documentation de SDM ainsi que leur utilité (indépendamment du contexte d'application de la méthode).
- Le troisième chapitre présentera ensuite les résultats des enquêtes et interviews menées au sein du département de la CGER.

Un premier point présentera le cadre de l'enquête.

Un second abordera les réponses fournies à propos de la réalisation de la documentation et de son utilité.

Ces résultats seront ensuite interprétés.

Un troisième point abordera les réponses concernant les possibilités et souhaits d'automatisation de la documentation. Ces résultats seront interprétés compte-tenu des outils informatiques présents à la CGER.

Un dernier point tentera de définir des objectifs prioritaires pour le groupe de support au développement de la CGER à partir des souhaits d'automatisation exprimés par les développeurs au point 3.

- Le dernier chapitre présentera les objectifs à long terme du département informatique concernant l'automatisation du processus de développement de systèmes d'information et plus particulièrement, l'automatisation de la documentation des systèmes développés.

1. PRESENTATION DE LA CGER ⁽¹⁾

1.1 LA CGER

La Caisse Générale d'Epargne et de Retraite a été créée en 1865 afin de collecter l'épargne populaire à l'échelle nationale. Au début, elle ne dispose pas de propre réseau commercial, les bureaux de poste lui servant d'intermédiaires habilités à recueillir et à rembourser l'épargne. Ce choix initial va influencer toute la structure ultérieure de la banque : il entraîne la centralisation et la centralisation administrative et il isole l'entreprise de son marché. Au cours des années, le réseau des bureaux de poste devient un obstacle à l'expansion après avoir été un des facteurs de réussite. Pour faire face à la concurrence des banques privées, la CGER va constituer son propre réseau d'agences à partir de 1959. La politique d'ouverture d'agences atteint les sommets entre 1975 et 1980 en créant jusqu'à 125 agences par an. Cette politique coïncide avec l'informatisation des opérations de guichet. Le développement du réseau d'agences crée la nécessité d'une décentralisation régionale. A partir des années 1976-1977, la CGER établit six sièges régionaux regroupés au sein de trois directions (Flandres, Bruxelles et Wallonie) et qui viennent chapeauter le réseau d'agences. Le poids des économies d'échelle et des rapports de puissance hérités du passé contribua toutefois à limiter les attributions accordées aux sièges régionaux. Les équipements et les développements informatiques sont presque totalement sous contrôle du siège central. La CGER ne porte aujourd'hui plus bien son nom : l'épargne traditionnelle lui échappe peu à peu et son rôle dans les retraites s'amenuise. En effet, elle est devenue durant ces dernières années une banque semblable aux grandes banques privées du pays avec ses filiales à l'étranger, son secteur d'assurances, ... Figurant parmi les quelques plus grandes banques belges, avec ses 1100 agences environ, ses quelques 10000 employés, la CGER reste une banque publique. En 1980, le législateur l'a autorisée à se déspecialiser au point de pouvoir fournir les mêmes services que les banques privées. Le législateur a imposé dans le même mouvement une division comptable ainsi que la création de deux conseils d'administration et de deux directions distinctes pour les activités de banque et pour les activités d'assurance. Cette division se retrouve, comme nous le verrons plus loin, au sein de la structure du département informatique. La CGER a donc élargi ses services: création de sociétés de leasing et de factoring, l'ES-finance, les ACE-factors, l'assurance-vie, l'assurance auto, ... Cet élargissement des activités a eu un effet considérable sur l'informatique puisque celle-ci était obligée de suivre le mouvement. Les besoins des utilisateurs du département informatique sont devenus plus complexes et plus pressants. De plus, le département est comme la plupart des départements informatiques des grandes entreprises, immergé dans la maintenance : il doit soutenir et adapter. Les délais de livraison des nouvelles applications paraissent démesurés aux yeux des responsables utilisateurs qui les

⁽¹⁾ Pour éviter toute redondance inutile, cette présentation est inspirée du mémoire de D. MARCELIS (1)

réclament. Bien que la bibliothèque des programmes contient environ 10000 logiciels, de nombreuses demandes économiquement justifiées doivent parfois attendre des années avant de se voir octroyer un projet informatique. C'est dans ce cadre qu'est venue s'instaurer la méthode de développement SDM.

1.2 LE DEPARTEMENT INFORMATIQUE

Le centre de traitement de l'information de la CGER occupant quelques 550 personnes, constitue un service particulier de support au sein de la CGER-banque et de la CGER-assurances. De ce fait, la stratégie informatique de la CGER est élaborée à partir de la stratégie de la banque et de l'assurance.

La CGER a une politique orientée produits qui se visualise au niveau de la structure du CTI. Avant de proposer cette structure, il serait bon de rappeler les principales étapes du processus d'informatisation de la CGER qui permettront également de mieux comprendre les méthodes et outils de développement présents actuellement à la CGER.

a. PRINCIPALES ETAPES DE L'INFORMATISATION DE LA CGER

Comme nous l'avons dit plus haut, la CGER décida dès 1959 de créer son propre réseau d'agences reliées chacune au siège central. On fit alors grand appel aux télécommunications. En 1976, les bases de données apparurent et, conscient des avantages qu'elles pourraient apporter, le département opta pour l'implémentation d'IMS (Information Management System) d'IBM. Cela ne se fit pas sans peine car les employés se considérant propriétaires des informations qu'ils manipulaient ne souhaitaient pas partager leurs fichiers. D'autre part, l'absence d'uniformisation des données ainsi que des définitions des termes utilisés imposait un travail de synthèse et de modélisation considérable car il était impossible de définir des fichiers IMS à partir d'une base aussi disparate. Les travaux de modélisation des bases de données et du dictionnaire des données débutèrent en 1978. Vers la même époque, les directions des entreprises d'une certaine taille commencèrent à s'inquiéter de l'ampleur et de l'évolution spectaculaire des charges informatiques. La CGER ne fit pas exception à la règle et, dans le souci d'améliorer la productivité des programmeurs, elle commença à s'intéresser aux méthodes de programmation de l'école européenne qui, opposée à l'école américaine, prenait en compte le niveau conceptuel d'analyse (quelles sont les informations véhiculées dans l'entreprise et les relations entre elles ?) où toute quantification et tout chemin d'accès étaient exclus. Les résultats de l'analyse conceptuelle étaient ensuite transformés en un schéma logique (sans considération physique) qui à son tour était transformé en schéma physique. L'approche américaine ne reprenait que les niveaux logiques et physiques. Après de nombreuses années, l'apport du niveau conceptuel fut reconnu par tout le monde et est devenu aujourd'hui un stade de développement obligatoire; les développeurs et les chefs de projet s'étaient rendu compte qu'une conception préalable permet une meilleure maîtrise de leurs projets.

Au niveau des bases de données, le premier essai d'implantation d'IMS fut réalisé pour certains traitements des comptes universels (comptes courants de la CGER). Un premier stade de cette implémentation revint à définir toutes les informations que manipulaient les traitements et à définir les règles à appliquer à ces informations.

Le relevé des informations utilisées engendra la création d'un dictionnaire de données manuel qu'il devint très vite impossible d'utiliser vu son ampleur. Un dictionnaire informatisé fut alors choisi : DATAMANAGER qui est toujours utilisé actuellement. D'un simple support de documentation, il a évolué vers un véritable gestionnaire de données.

IMS est un système orienté vers le traitement d'un grand nombre de transactions relativement simples avec un temps de réponse négligeable. Ces dernières années, le besoin d'un second système davantage réservé à un traitement plus lourd et plus sophistiqué de l'information s'est fait sentir et le système DB2 (Data Base2) d'IBM a été choisi et sa mise en service date de 1990.

b. STRUCTURE DU DEPARTEMENT INFORMATIQUE

La CGER est divisée en deux entités : les activités bancaires et les activités d'assurances. Le département informatique quoique situé dans l'entité des assurances travaille sans distinction pour les deux entités. Il comprend deux grandes parties : la partie production et la partie développement.

La production concerne principalement le fonctionnement du centre de calcul et les modes d'utilisation des ordinateurs. A cela sont rattachés divers services tels que l'imprimerie, la publication électronique, ...

La partie développement s'occupe de la conception et de la réalisation de l'ensemble des applications nécessaires à tous les départements de la CGER. Elle comprend également les deux groupes de recherche et de mise en oeuvre de méthodes et d'outils de développement de logiciels : le groupe "Méthodes" et le groupe "Outils". Le groupe "Méthodes" est chargé des méthodes, des outils de niveau conceptuel ainsi que de la formation. Le groupe "Outils" est chargé du dictionnaire de données, des outils de développement ainsi que de la qualité des logiciels.

La partie développement est elle-même décomposée en trois cellules de Développement de Systèmes d'Information (DSI).

Le DSI1 qui développe des applications pour l'entité bancaire, le DSI2 qui développe des applications pour l'entité assurances et le DSIM qui développe les applications communes aux deux entités et qui en plus, comprend les groupes "Outils" et "Méthodes".

Chaque cellule est composée de plusieurs groupes de développement, chacun ayant la charge d'un ou plusieurs produits de la CGER. Chaque groupe de développement constitue une équipe de travail composée d'un éventail complet des grades existant dans le département informatique.

Il s'agit :

- des informaticiens dont il existe quatre classes :
 - informaticiens D (chefs de départements)
 - informaticiens C (chefs de services)
 - informaticiens B (chefs de projet ou responsables d'un groupe de développement)
 - informaticiens A (chefs de projet)
- des analystes.
- des programmeurs.

Les groupes de développement travaillent dans un environnement très dynamique et relativement imprévisible défini par les politiques des produits d'assurances et des produits bancaires dont la détermination leur échappe. De ce fait, il ne peuvent prévoir même à court terme les demandes de développement qui leur seront soumises.

La figure ci-dessous vous présente la structure du département informatique :

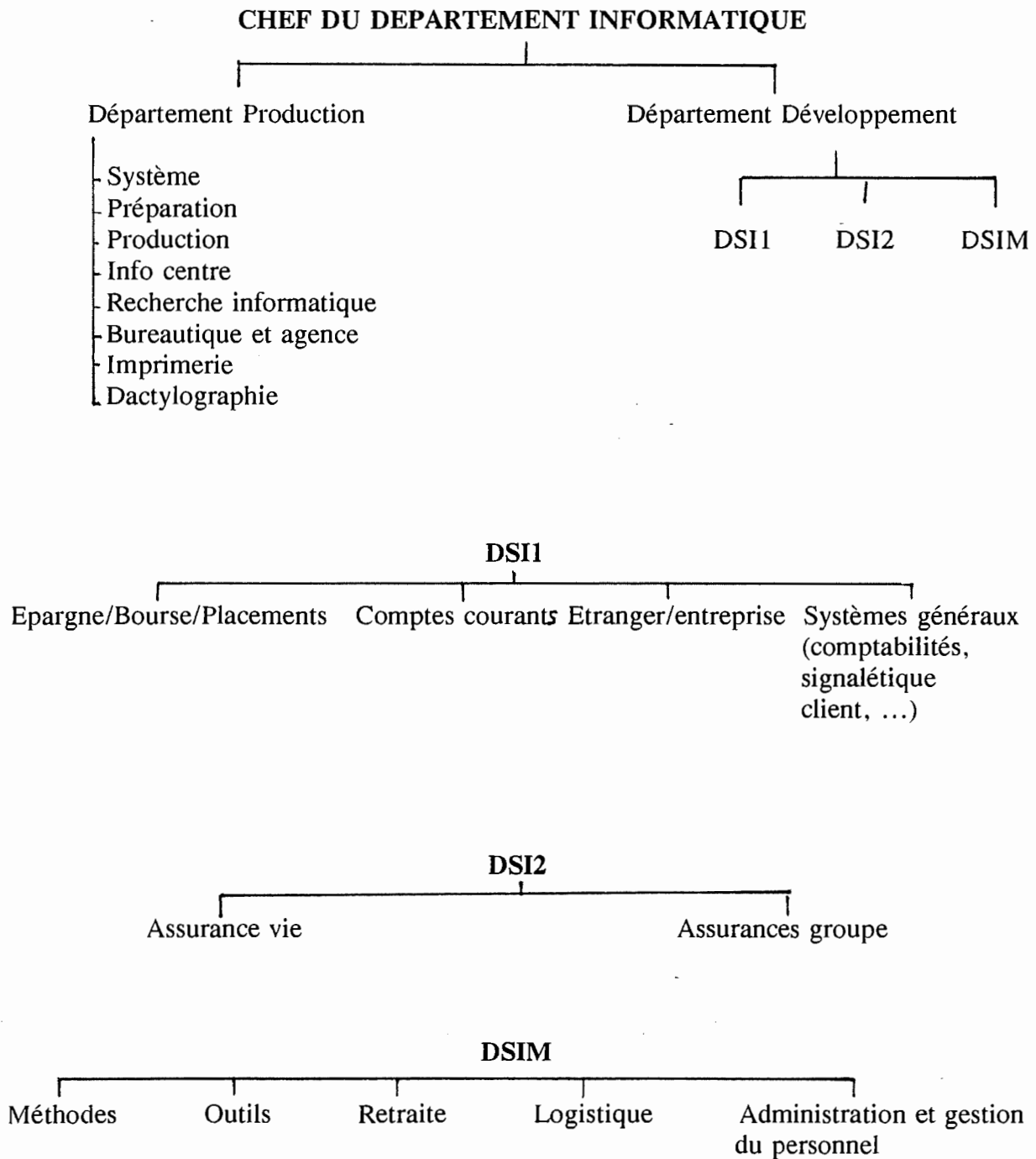


Figure 1: structure du département informatique

2. OBJECTIF DU MEMOIRE ET INTERET DU SUJET

Nous avons vu que fin des années 70, les directions des grandes entreprises se sont inquiétées de l'ampleur et de l'évolution spectaculaire des charges informatiques et ont donc décidé d'exercer un contrôle de ces charges. Un des principaux soucis était et reste toujours l'amélioration de la productivité des développeurs. C'est dans ce but que la CGER a créé deux groupes de support au développement; le groupe "Méthodes" et le groupe "Outils".

Le groupe "Méthodes" a eu la charge de choisir et de promouvoir une méthode de développement de logiciel.

Leur choix s'est porté sur la méthode SDM dont la grande souplesse lui permettait de s'instaurer progressivement en s'intégrant le mieux possible au passé.

Ensuite, ils se sont rendus compte qu'une méthode seule ne servait pas à grand chose et ils se sont donc intéressés aux outils conceptuels de haut niveau capables de supporter la méthode. Ils ont opté pour EXCELERATOR.

Ce mémoire réalisé à l'intention du groupe "Méthodes" s'intéresse à un aspect tout particulier de la productivité des développeurs qui est la qualité et la documentation réalisée tout au long du développement.

Pour améliorer la qualité de la documentation, on va mettre l'accent sur ce qui est automatisable, c'est-à-dire déterminer les parties de la documentation pour lesquelles il est possible de disposer d'un outil qui en faciliterait la réalisation.

L'objectif de ce mémoire est triple :

- En premier lieu, il s'agit de mieux connaître ce qui se fait actuellement au sein des équipes de développement en matière de documentation.
- Ensuite, déterminer les types de documentation régulièrement réalisés et pour lesquels les développeurs souhaitent une automatisation.
- Finalement, il recherche parmi les outils présents à la CGER, ceux qui sont susceptibles de répondre aux besoins.

Le résultat final consistera en un certain nombre d'objectifs prioritaires pour les groupes de support au développement. Ces objectifs tiendront compte de deux priorités fondamentales qui sont les souhaits des développeurs d'une part et la productivité des développeurs d'autre part.

Durant le stage, les développeurs ont suscité un intérêt tout particulier à ce mémoire : pour récolter les informations, nous avons rédigé et envoyé un questionnaire à tous les développeurs. Dans ce document était présenté un tableau récapitulatif de SDM qui, pour chaque activité de chacune des six phases de développement, propose le type de documentation que SDM suggère de réaliser.

Ce tableau condensé sur deux pages s'est avéré une "check-list" de documentation très pratique puisqu'elle résumait sur deux pages ce que SDM présente tout au long du livre de référence.

La critique théorique de la documentation présentée au chapitre 2 devrait permettre aux personnes intéressées de justifier ce tableau.

* * *

CHAPITRE 1

**La méthode SDM
Les modèles, méthodes et
outils d'aide au développement**

1.1 SDM : UNE METHODE DE DEVELOPPEMENT DE LOGICIELS

Les systèmes d'information sont de plus en plus nombreux et complexes au sein des organisations de taille respectable. De ce fait, leur conception constitue une tâche de plus en plus difficile et coûteuse. Pour faire face à ces problèmes, les organisations se sont intéressées aux méthodologies de développement de systèmes d'information. Actuellement, les entreprises reconnaissent la nécessité de recourir à une méthode et selon C. ROLLAND (2) les raisons sont les suivantes : "Une méthode est nécessaire :

- pour maîtriser la complexité du problème informationnel à résoudre;
- pour sortir la construction des S.I. de l'empirisme individuel et la fonder sur une coopération efficace entre informaticiens et gestionnaires;
- pour permettre la communication entre individus de l'équipe de conception;
- pour construire des systèmes pertinents, fiables, flexibles et adaptatifs;
- pour permettre d'évaluer le système à tout moment de son cycle tant sur le plan de son efficacité technique que sur celui de sa pertinence par rapport aux besoins des gestionnaires;
- pour améliorer les coûts, les délais et la productivité des activités de développement."

Les méthodes de développement sont nombreuses et variées et nous allons nous intéresser à l'une d'entre elles développée par PANDATA, une importante maison de software néerlandaise : la méthode SDM ou System Development Methodology.

1.1.1 CARACTERISTIQUES D'UN PROJET DE DEVELOPPEMENT

SDM est une méthode qui peut s'appliquer à tout type de projet. C'est pourquoi nous présentons d'abord ci-dessous les caractéristiques (citées dans la méthode SDM) permettant de différencier les types de projet :

- *le degré de dépendance du nouveau système par rapport au système existant* : système d'information entièrement nouveau, refonte d'un ancien, maintenance partielle ou encore élargissement d'un système existant.
- *le degré de priorité et la position stratégique du projet* : projet relevant de l'infrastructure opérationnelle générale ou application locale et non stratégique.
- *le délai de réalisation dont on dispose.*
- *la formation et l'expérience des développeurs.*
- *le support méthodologique en outils, systèmes et hardware dont disposent les développeurs.*
- *le lieu de développement du système* : développement interne ou par une firme de développement extérieure.
- *la taille de l'équipe de travail et la mobilité du personnel.*
- *le risque lié au projet* et défini comme la probabilité que le projet n'atteigne pas ou partiellement ses objectifs.

Cette dernière caractéristique que constitue le risque lié au projet influence fortement le développement et la gestion du projet et nous y ferons allusion à plusieurs reprises au chapitre suivant. MAC FARLAN (3) s'est particulièrement intéressé à cette notion et distingue trois classes principales de facteurs qui influencent le risque lié à l'implémentation d'un projet : la taille du projet, l'expérience des développeurs avec la technologie employée, la structure du projet. En combinant ces facteurs, il identifie huit catégories de projets présentées dans le tableau ci-dessous et auxquelles il associe les degrés de risque suivants : risque très faible, risque faible, risque modéré, risque élevé, risque très élevé.

STRUCTURE DU PROJET

		STRUCTURE DU PROJET	
		ELEVEE	FAIBLE
NIVEAU DE LA TECHNO- LOGIE UTILISEE PAR RAPPORT A	FAIBLE	TAILLE : IMPORTANTE RISQUE : FAIBLE	TAILLE : IMPORTANTE RISQUE : FAIBLE- MODERE
		TAILLE : MODESTE RISQUE : TRES FAIBLE	TAILLE : MODESTE RISQUE : FAIBLE- TRES FAIBLE
		TAILLE : IMPORTANTE RISQUE : MODERE- ELEVE	TAILLE : IMPORTANTE RISQUE : TRES ELEVE
L'EX- PERIENCE DES DEVELOP- PEURS	ELEVEE	TAILLE : MODESTE RISQUE : FAIBLE- MODERE	TAILLE : MODESTE RISQUE : ELEVE

- Tableau 1 : Les catégories de projet de MAC FARLAN -

1.1.2 SDM A LA CGER

Confrontée à des problèmes de délais et de réduction des coûts, la CGER a choisi la méthode SDM pour standardiser ses développements et améliorer ainsi la productivité des équipes de développement.

Dans son ouvrage intitulé "Systeemontwikkeling volgens SDM", H.B. HEILERS (4) présente SDM comme une méthode théorique globale qui doit être vue comme une **ossature** de la méthode à appliquer, indépendamment de la situation donnée. SDM propose une démarche à suivre et reste générale quant aux techniques et outils à employer. C'est pourquoi elle doit être adaptée à l'environnement dans lequel elle devra s'insérer ainsi qu'aux caractéristiques de chaque projet à développer. C'est dans ce but que différents manuels présentés ci-dessous ont été réalisés à la CGER.

- Tout d'abord, un résumé de la méthode a été élaboré. Il reprend pour chaque étape de développement, son nom, le responsable à contacter, le numéro de téléphone de ce responsable ainsi que les techniques applicables. Cependant on peut constater que l'identification du responsable par son nom est un mauvais choix car les personnes ne restent pas indéfiniment au même poste et, par conséquent, l'absence de mise à jour rend une partie de ce résumé inutilisable. Ce document est repris en annexe 1.
- D'autre part, un guide pratique de l'utilisation de la méthode SDM à la CGER a été élaboré par le groupe de support méthodologique en 1989. Sous le titre de "vade mecum du développeur", il a pour objectif d'adapter SDM à la CGER en déterminant pour chaque étape son but, les techniques et outils à utiliser, la manière de procéder, la documentation à réaliser et à consulter et l'application de l'étape à la CGER (techniques possibles, cours existants, informations disponibles, personnes à contacter, ...). Ce document est relativement neuf et volumineux, ce qui explique que peu de développeurs déclarent l'exploiter.
- Finalement, on peut également trouver une synthèse de la méthode SDM réalisée par deux mémorants à la CGER. Elle est très utile car elle permet une approche simple et complète de SDM. Ce document se trouve en annexe 2.
- En ce qui concerne l'implantation de la méthode SDM à la CGER, un mémoire d'étudiant y a été entièrement consacré. Sous le titre "Implantation de méthodes de conception et de développement de SI, étude de cas" (D. MARCELIS, 1), ce travail permet de constater le degré d'application de la méthode à la CGER, les principaux points forts et faibles de cette implantation ainsi que les changements que SDM a apportés dans la façon de travailler des développeurs.

1.1.3 PRESENTATION DE LA METHODE SDM

C. ROLLAND (5) distingue deux grands types de méthodes de conception des systèmes d'information : les méthodes cartésiennes et les méthodes systémiques. Les **méthodes systémiques** sont centrées sur la modélisation des données et le système d'information est perçu comme **une structure qui fournit une représentation des faits présents et passés de la vie de l'organisation**. Les méthodes orientées objet font partie de cette catégorie.

Les **méthodes cartésiennes**, les plus anciennes et les plus utilisées dans le monde, proposent une analyse et une conception du système d'information axée sur la modélisation des processus où le système d'information est identifié à **une fonction globale de gestion**. Ces méthodes préconisent une démarche descendante top-down qui part de la fonction globale et aboutit par raffinements successifs à la définition de processus élémentaires. Ces méthodes sont donc fondées sur la décomposition hiérarchique des processus. SDM fait partie de ces méthodes.

La méthode SDM repose sur le modèle de développement de logiciel "Waterfall"; il s'agit d'un modèle orienté 'activités' et qui présente le cycle de vie d'un projet comme une succession de huit phases avec des possibilités de rétroaction limitées à un niveau afin de minimiser le travail à recommencer. Le critère d'accomplissement des quatre premières phases est la réalisation de documents complets du travail réalisé. On reproche généralement à ce modèle son manque de flexibilité. Pour plus de détail, on consultera l'ouvrage de Barry W. BOEHM (6).

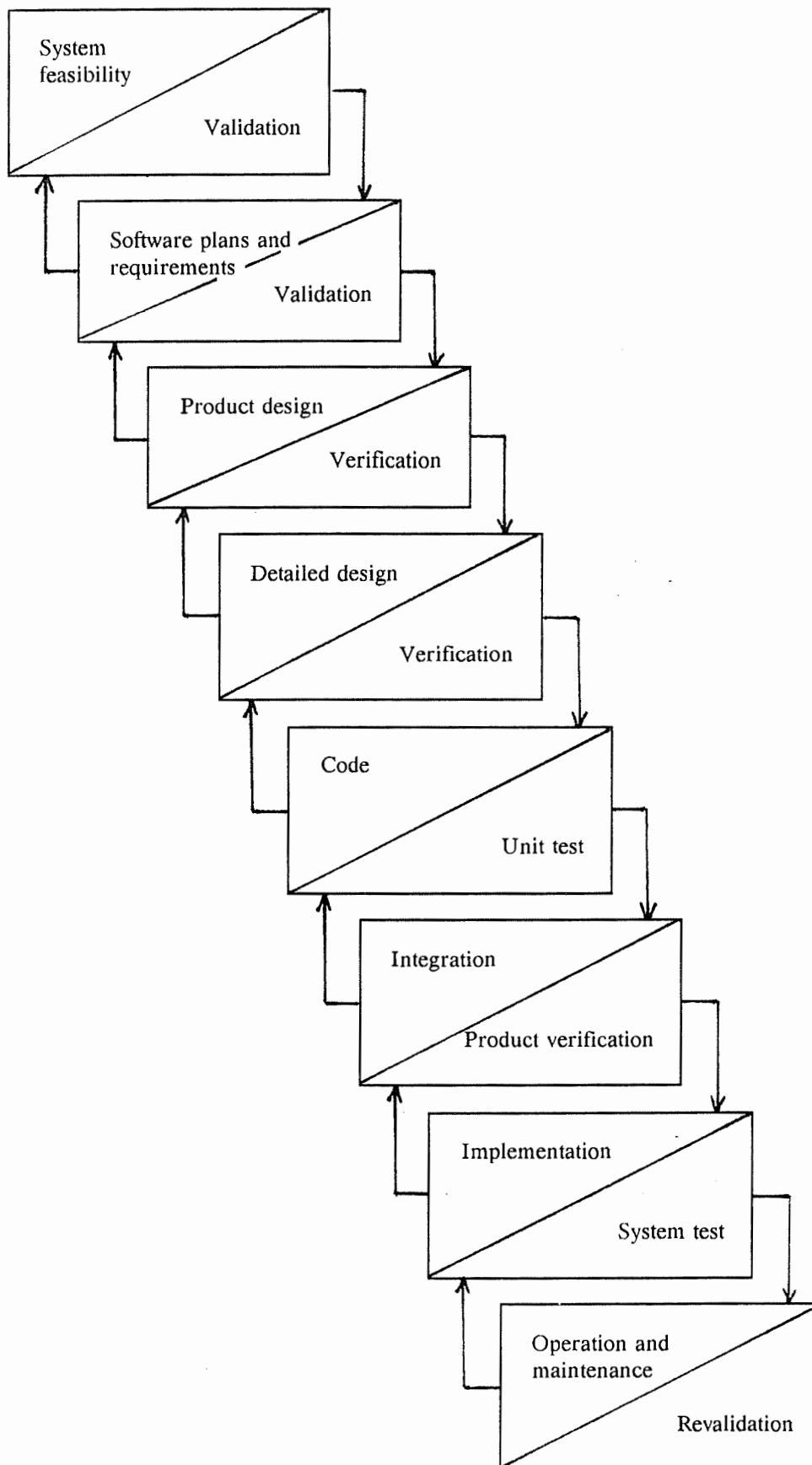


Figure 2 : Le modèle "Waterfall" du cycle de vie d'un logiciel

SDM découpe le développement de logiciel en **sept phases** d'égale importance, chacune permettant de passer d'un niveau de description à un niveau plus détaillé, en enrichissant la solution d'une composante supplémentaire. Ces sept phases sont proposées ci-dessous et mises en correspondance avec les huit phases du modèle Waterfall :

SDM	WATERFALL
- l'analyse conceptuelle	↔ 1. Faisabilité du système 2. Exigences et plan du logiciel
- l'analyse fonctionnelle	↔ 3. Conception du produit
- l'analyse technique	↔ 4. Conception détaillée
- la programmation et les tests isolés	↔ 5. Code
- les tests	↔ 6. Intégration
- la conversion et la mise en production (conversion des données et préparation de l'organisation)	↔ 7. Implémentation
- la gestion et la maintenance du système	↔ 8. Opération et maintenance

SDM décrit donc étape par étape tout le cycle de vie d'un projet. Ces étapes, exécutées selon un ordre précis, sont elles-mêmes divisées en un certain nombre d'activités. Dans son ouvrage, H.B. HEILERS (4) présente SDM de la manière suivante : chaque phase est introduite par une description d'un planning possible de la phase ainsi qu'une présentation des différents résultats à obtenir. Ensuite, chaque phase est décomposée en un certain nombre d'activités et chacune d'elle est décrite de la manière suivante :

- une introduction qui donne un aperçu du but et des bases de l'activité et notamment **la documentation qui doit être réalisée**; elle indique également les techniques qui peuvent être appliquées.
- la façon de faire; celle-ci explique comment réaliser l'activité.
- une indication des informations de base à utiliser.
- un exemple de résultat de l'activité.

Certaines activités sont obligatoires, d'autres sont optionnelles et leur exécution dépend des circonstances :

- la nature du projet peut rendre certaines étapes inutiles :
Par exemple, pour un système entièrement nouveau, l'étude de l'existant sera réduite et il n'y aura pas de problème de conversion des données.
- certaines activités peuvent être sous-traitées par d'autres départements ou par des tierces personnes.
- lorsque les exigences ou les buts d'un projet changent en cours de développement, certaines étapes doivent être réeffectuées.

La méthode SDM peut être utilisée à des fins diverses dont les plus importantes sont selon H.B. HEILERS (4) :

- une "**check-list**" pour le développement et la gestion d'un système d'information.
- une base pour la **planification**, le suivi de l'avancement et le contrôle de la qualité des projets.
- un **moyen d'uniformisation** en ce sens qu'elle offre une approche identique du développement à tous les développeurs.
- un outil de **communication** entre les utilisateurs, les clients et les spécialistes informatiques.
- une base de **documentation** d'un projet.
- une base de **concertation**.

Cette méthode n'a pas pour rôle de réduire l'activité de création des développeurs mais permet de les guider dans leur réflexion en leur offrant une liste des points auxquels ils doivent penser. C'est une raison pour laquelle les développeurs sont plutôt déçus. Ils espèrent disposer d'une méthode qui leur permettrait de résoudre tous les problèmes sans difficulté, c'est-à-dire pareille à un toboggan sur lequel ils n'auraient qu'à se laisser glisser pour arriver sans effort à la solution. Or, l'activité de développement de système est un processus de réflexion, de création et de prise de décisions, qui se base sur l'utilisation d'acquis (connaissances, méthodes et expériences).

1.2 LES MODELES D'AIDE AU DEVELOPPEMENT

Afin de faciliter la lecture des prochains chapitres, il est utile de présenter les principaux modèles, méthodes et outils d'aide au développement de systèmes d'information existant à la CGER et qui s'accordent avec la méthode SDM.

1.2.1. LA MODELISATION CONCEPTUELLE DES DONNEES

La modélisation conceptuelle des données s'effectue à l'aide du modèle entité-association. Elle permet de définir la sémantique des données en définissant les données ainsi que les relations entre celles-ci. Pour plus de détail, il suffit de consulter l'ouvrage de F. BODART et Y. PIGNEUR (7). Le schéma conceptuel s'obtient en suivant les étapes suivantes :

- énumération des informations qui seront manipulées par le futur S.I.
- élimination des redondances et des synonymes qui fournira alors une liste des propriétés des différents objets.
- déduction des types d'entité.
- rattachement des différentes propriétés aux entités.
- définition des types d'association entre types d'entité.
- détermination des cardinalités des types d'association.
- définition des contraintes d'intégrité.
- quantification du modèle conceptuel.

1.2.2. LE MODELE DYNAMIQUE DES DONNEES

Le modèle dynamique des données constitue une étape essentielle dans l'architecture des bases de données et à la CGER, il est contrôlé automatiquement grâce au logiciel PALOMA de l'équipe de support "méthodes" qui a rendu ce point de contrôle obligatoire pour tout projet. Ce logiciel sera présenté lorsque nous aborderons les outils. Le modèle dynamique des données indique les points d'entrée dans le modèle conceptuel, les accès dont les données peuvent faire l'objet et estime le trafic sur les arcs reliant les objets. Une fois terminé, il aidera l'administrateur des bases de données à transformer le modèle conceptuel en modèle logique des données. Pour plus de détail, on consultera l'ouvrage de Jean-Luc HAINAUT (8) dans lequel il décrit la réalisation du modèle dynamique des données qu'il nomme encore "schéma des accès possibles".

1.2.3. LE MODELE LOGIQUE DES DONNEES

Le modèle logique des données décrit les structures de données non seulement du point de vue de la sémantique des données mais également au niveau des accès que l'on peut faire à ces données. Ce modèle est réalisé à partir du modèle conceptuel des données et utilisera le modèle dynamique des données ainsi que les résultats des calculs d'estimation de trafic et d'accès aux données qui auront éventuellement été réalisés et qui contribueront à obtenir un modèle logique performant. Pour plus de détail, nous vous renvoyons à l'ouvrage de Jean-Luc HAINAUT (8) dans lequel il décrit la réalisation du modèle logique des données.

1.2.4. LE DIAGRAMME DE FLUX DE DONNEES

Un diagramme de flux de données (DFD)" permet de représenter la production, la circulation et la destination des informations échangées entre les différents composants de l'organisation" (F. BODART, 7).

A la CGER les concepts de base de ce diagramme sont : les flux représentant la circulation des informations, les traitements correspondant aux transformations opérées sur les informations, les stockages d'information, les acteurs externes permettant de modéliser les sources et les destinations externes des flux. Le diagramme de flux de données n'est pas unique. Un premier diagramme global est réalisé et est ensuite décomposé en diagrammes successifs de plus en plus détaillés et jusqu'à l'obtention d'un diagramme présentant des traitements

élémentaires.

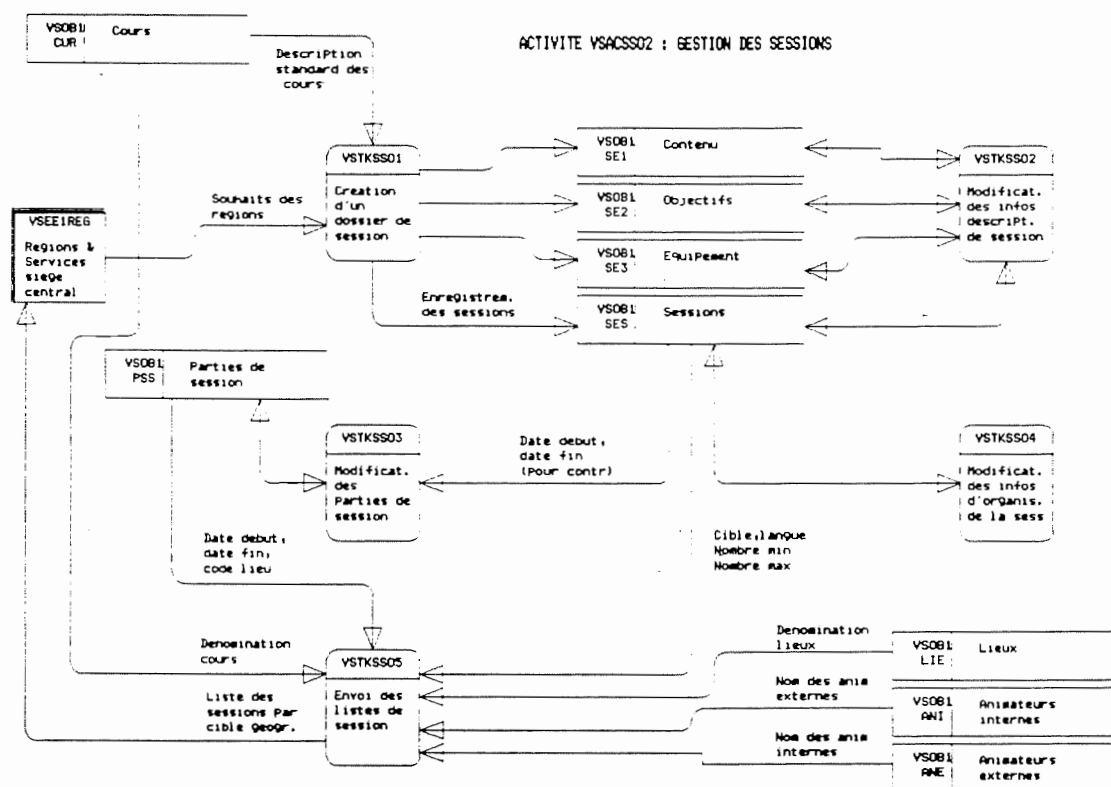


Figure 3 : Diagramme de flux de l'activité "gestion des sessions de formation"
Conventions de la CGER

1.2.5. LA STRUCTURE HIERARCHIQUE DES ACTIVITES ET DES TACHES

La structure hiérarchique des activités et des tâches (SHAT) est une technique de structuration des traitements qui permet de présenter l'arborescence du système en le décomposant en sous-systèmes, ces derniers en activités et les activités en tâches. Chaque niveau hiérarchique est défini par son niveau de détail. Ce modèle permet de structurer le projet par raffinements successifs sur base des processus selon la démarche proposée par les méthodes de développement cartésiennes.

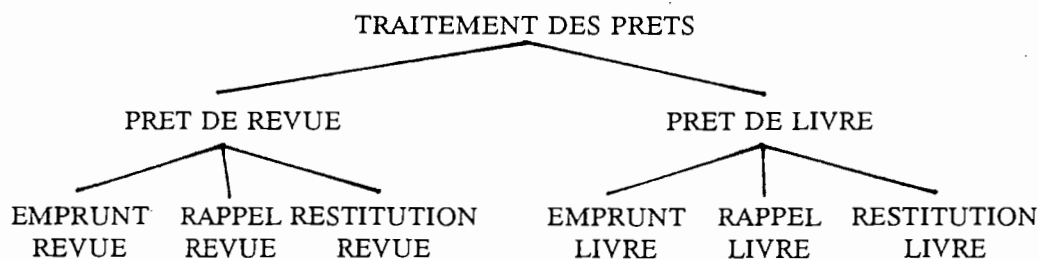


Figure 4: Diagramme de structuration du traitement "prêts dans une bibliothèque"
(F. BODART, 7).

1.2.6. LE DIAGRAMME "HIERARCHICAL INPUT-PROCESS-OUTPUT"

Le diagramme "hierarchical input-process-output" (HIPO) est un modèle proposé par IBM et qui permet de décrire en détail chaque application du système. Ce diagramme prend la forme d'un tableau de trois colonnes : la première décrit l'input à fournir pour exécuter la tâche, la deuxième décrit le traitement à réaliser pour obtenir l'output désiré à partir de l'input, la troisième décrit l'output de la tâche.

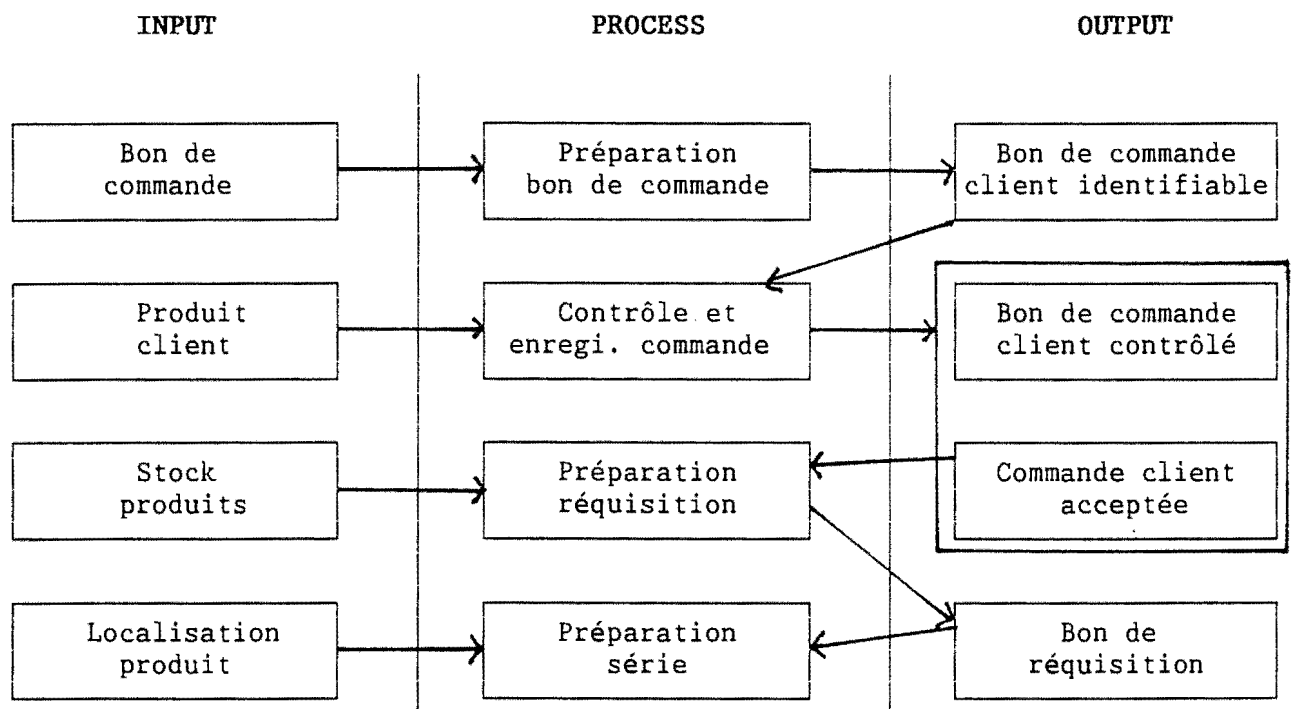


Figure 5 : Diagramme HIPO de l'application "traitement des commandes d'un client"

1.3 LES METHODES D'AIDE AU DEVELOPPEMENT

Nous vous présentons ci-dessous les deux principales méthodes d'aide au développement introduites et soutenues à la CGER par le groupe de support "Méthodes".

1.3.1. LA METHODE "JACKSON STRUCTURED PROGRAMMING"

La méthode "Jackson Structured Programming" ou JSP est une méthode de description des programmes que le groupe de support méthodologique de la CGER tente de généraliser. Son principe est de présenter sous forme d'arborescence les input et les output du traitement et de mettre ces deux structures en correspondance en mettant en évidence les parallélismes existant entre elles. Cette description des programmes permet de donner une structure de données générale sur laquelle le programme sera construit.

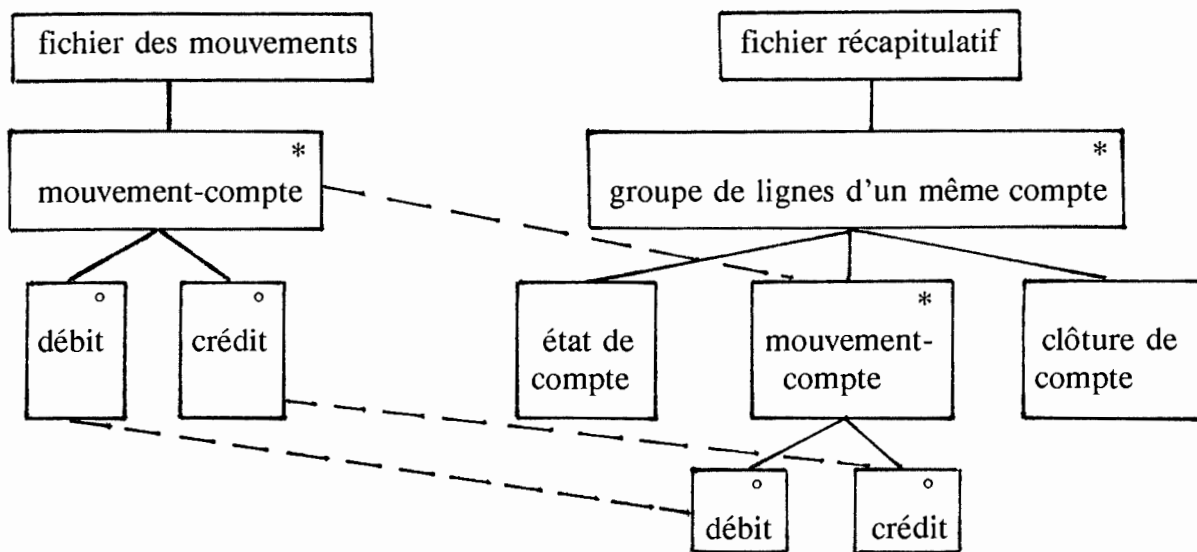


Figure 6 : Diagramme JSP du programme "Récapitulatif des mouvements de compte des clients"

1.3.2. LA METHODE "INTERPROGRAM FUNCTION POINT ANALYSIS"

La méthode d'analyse par point de fonction (IFPA) est un outil de prévision de la charge et du temps de développement des systèmes informatiques indépendamment des outils mis en oeuvre pour la réalisation. Le principe de mise en oeuvre de la méthode repose sur les quatre étapes suivantes :

1. Définition du modèle logique des données (cfr. 1.2.3).
2. Inventaire et description des tâches constituant la partie active de l'application (utilisation possible des modèles tels SHAT ou HIPO)
3. Utilisation de tables de pondération IFPA sur les résultats des points (1) et (2) ci-dessus pour obtenir le nombre de points de fonction. Un point de fonction est une unité de charge de référence indépendante des outils de développement utilisés. Le nombre de points de fonction est directement dépendant de la taille et de la complexité du modèle des données, du nombre et du type de tâches ainsi que de la façon dont les tâches utilisent les données.
4. Application des coefficients de productivité aux résultats précédents pour obtenir une estimation en terme de nombre de jours*hommes. Les coefficients de productivité tiennent compte des aspects suivants :
Utilisation d'outils informatiques CASE, existence d'une documentation claire, taille de l'application (en nombre de points de fonction), complexité des activités de l'utilisateur.

Le calcul définitif des points de fonction se fait en fin d'analyse fonctionnelle puisqu'on dispose à ce moment d'un modèle logique des données et d'une découpe de l'application en tâches. Il est toutefois conseillé de fournir une estimation précoce en fin d'analyse conceptuelle afin de préciser les résultats de l'analyse coûts & bénéfices. On se basera pour cela sur le modèle conceptuel des données et la découpe du système en activités.

Cette méthode a été introduite à la CGER en 1991 par le groupe de support "Méthodes" dans le but de réaliser une estimation plus fiable de la charge de développement du projet. Toutefois, l'introduction de la méthode est encore trop récente pour que l'on puisse juger de sa fiabilité.

Un exemple simplifié d'application de IFPA est proposé en annexe 4.

1.4 LES OUTILS D'AIDE AU DEVELOPPEMENT

Ces dernières années ont vu la prolifération d'outils d'aide au développement de tout genre : du simple outil de "dessin" aux CASE tools (computer-aided software engineering tools) permettant l'automatisation de certaines tâches du développement.

Nous présentons, ci-dessous une description des principaux outils de développement présents à la CGER.

1.4.1. LE DICTIONNAIRE DE DONNEES "DATAMANAGER"

DATAMANAGER est présent depuis longtemps à la CGER. Il constitue le recueil des informations que l'on souhaite conserver sur les informations de l'entreprise et notamment sur les systèmes d'informations existants à la CGER. Il rassemble des méta-informations et les liens qui les unissent. L'avantage de cet outil est qu'il permet de gérer la mémoire de l'entreprise en favorisant l'utilisation de notions communes et en diminuant la redondance et les coûts qui en découlent. Chaque développeur a accès à DATAMANAGER ce qui lui permet de consulter la définition de tous les concepts qu'il doit manipuler.

Le schéma suivant montre que la structure de DATAMANAGER reflète à la fois les différents domaines de l'entreprise (comptes universels, crédit) et les divers environnements d'exploitation (développement, quality assurance, production).

La documentation du système dans DATAMANAGER est structurée en trois dossiers principaux :

FUDOS qui couvre les résultats des analyses conceptuelle et fonctionnelle.

TEDOS qui couvre les résultats de l'analyse technique et de la programmation.

PRODOS qui contient la documentation nécessaire à la mise en production du système.

Les informations sont structurées en différents types de membres. Chaque membre est décrit à l'aide de clauses descriptives et de clauses relationnelles en fonction de son type. Un langage de manipulation de données permet à l'utilisateur d'accéder aux données pour des tâches de consultation ou de modification.

Une description plus détaillée de DATAMANAGER est proposée en annexe 5.

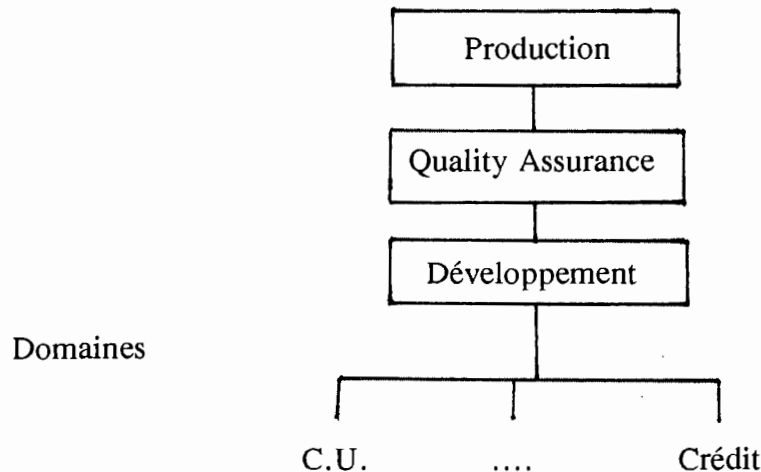


Figure 7 : Structure du dictionnaire "Datamanager"

1.4.2. PALOMA ⁽²⁾

Le logiciel PALOMA "Path Load Matrix" a été développé à la CGER. Il sert à établir le modèle dynamique des données, étape essentielle dans l'architecture des bases de données. En outre, PALOMA permet d'estimer la charge représentée par les tâches d'un système. Notons qu'il fournit des résultats séparés pour les tâches 'batch' et pour les tâches 'interactives' : il fournit deux types de résultats :

Au niveau du modèle conceptuel des données, PALOMA fournit une estimation de la charge supportée par chaque arc reliant les diverses entités du modèle conceptuel des données. L'estimation du trafic sur un arc tient compte du nombre de tâches qui le parcourent, de la fréquence d'exécution de ces tâches sur une base quotidienne ainsi que du mode d'accès à l'extrémité de l'arc (lecture, modification, insertion et suppression). PALOMA fournit au choix, une liste récapitulative pour l'ensemble du modèle, ou une liste détaillée par arc et par point d'accès dans le modèle.

⁽²⁾ Cette présentation de PALOMA est tirée de l'ouvrage "PALOMA 2" (9) réalisé à la CGER par Christian GRAAS, informaticien

Au niveau des tâches, PALOMA fournit une estimation de la charge représentée par chacune des tâches, sur base du parcours qu'elle effectue dans le modèle conceptuel des données, du nombre d'occurrences d'objets mises en oeuvre et de la manière dont on accède aux objets. Comme précédemment, on peut obtenir soit une liste condensée résumant les charges de chaque tâche, soit une liste détaillée explicitant le parcours de chacune de ces tâches.

La force de PALOMA est qu'il a été conçu pour fournir un bénéfice maximum pour un investissement minimum. En effet, le modèle conceptuel des données et les descriptions des tâches du système sont décrits dans le dictionnaire DATAMANAGER. L'utilisation de PALOMA est obligatoire pour tout développement et est prise en charge par le groupe de support 'méthodes'.

Pour cela, il se base sur des informations introduites dans le dictionnaire DATAMANAGER et permet par la même occasion de valider le modèle conceptuel des données.

1.4.3. SUPERPROJECT EXPERT

SUPERPROJECT EXPERT est un outil puissant de gestion de projet et des ressources. Ce logiciel offre des techniques de gestion de projets sophistiquées - PERT, Gantt Charts, Work Breakdown Structure, méthode du chemin critique - pour simplifier la planification, le calcul des coûts, les rapports et la mise à jour de projets, qu'ils soient simples ou complexes. Il ne gère qu'un projet à la fois, ce qui n'est pas toujours le cas des chefs de projet.

Il s'agit d'un outil très puissant mais sa puissance dépend des données fournies en entrée. Il faut noter que la majorité des développeurs estiment que le volume minimum de données à fournir en entrée est trop important et ne disposent pas suffisamment de temps pour le faire. De plus, il s'agit d'un outil isolé sans possibilité d'intégration avec le dictionnaire DATAMANAGER.

Une description plus détaillée de SUPERPROJECT est proposée en annexe 6.

1.4.4. EXCELERATOR ⁽¹⁾

Acquis en 1989 par la CGER, EXCELERATOR est un ensemble d'outils d'analyse et de conception du type UPPER CASE TOOL qui a été introduit pour soutenir la méthode SDM. Il offre une assistance au développement durant les phases d'analyse conceptuelle et fonctionnelle. Ce logiciel permet de définir, vérifier et documenter l'architecture d'un système d'information avant d'en commencer l'analyse détaillée et la programmation. EXCELERATOR ne propose pas de

(1) Cette présentation de EXCELERATOR est tirée de l'ouvrage "CASE TOOLS ? MEER IN HET BIJZONDER : EXCELERATOR" (10) réalisé à la CGER par Martine CLAESSENS, informaticienne.

méthode de développement à suivre et est donc relativement flexible : il permet d'adopter des techniques de structuration diverses orientées processus ou données. Actuellement, la CGER réalise l'intégration de la méthode SDM au logiciel. Vu isolément, EXCELERATOR peut offrir des améliorations qualitatives du développement. Cet outil combiné avec d'autres pourra apporter en plus une augmentation de la productivité. A travers le menu principal du logiciel, montrons les facilités que propose EXCELERATOR :

1. Les graphiques : EXCELERATOR offre six types de diagrammes pour réaliser les modèles graphiques du système tant au niveau des données que des traitements. On trouve notamment le diagramme de flux, le diagramme entité-relation, le diagramme de structure de données selon JSP, le modèle de données logique, ... Ceci permet d'abandonner la réalisation et la modification manuelle des graphiques du système.
2. Le dictionnaire Excelsator : EXCELERATOR se différencie des outils de dessin habituels car il intègre un dictionnaire centralisé qui stocke les données relatives aux nouveaux développements. Il est directement accessible pour des opérations de consultation, d'écriture ou de mise à jour.
3. Ecrans et rapports : Cette option permet de 'prototyper' les écrans et rapports qui seront utilisés dans le système développé.
4. Analyse : Le développeur a la possibilité de valider ses graphiques (exemple : toutes les données sont-elles décrites ? sont-elles normalisées ? ..), de générer des rapports sur le contenu du dictionnaire, de contrôler automatiquement si les données sont correctes et complètes, ...
5. Interface du dictionnaire : Il s'agit de pouvoir importer ou exporter des fichiers du dictionnaire avec possibilité d'interdire les modifications. Ceci est très intéressant car l'objectif de la CGER est de réaliser un environnement de développement composé d'outils CASE intégrés et gravitant autour d'un dictionnaire central (conformément à la philosophie AD-CYCLE d'IBM). Dans ce but, la CGER réalise actuellement l'intégration du dictionnaire EXCELERATOR avec le dictionnaire central DATAMANAGER.
6. Documentation : Cette option permet de réaliser et d'imprimer un document de spécification du système entier qui inclura tous les diagrammes, des descriptions du dictionnaire, les spécifications des écrans et des rapports ainsi que des descriptions du système. Il est possible de définir des canevas de documents et donc d'insérer les standards de documentation déjà définis.
7. Maintenance : Cette option plus technique permet de modifier l'environnement de travail du logiciel (exemple : back-up, modification du mot de passe, ...)

Une description plus détaillée de EXCELERATOR est proposée en annexe 7.

En conclusion de ce premier chapitre, nous pouvons mieux cerner le rôle du groupe de support au développement au sein du département informatique.

Conscients du poids des acquis et habitudes de travail des développeurs avant l'instauration d'une méthode, les équipes de support "Méthodes" et "Outils" ont porté leur choix sur une méthode (SDM) très peu contraignante qui permet aux développeurs de conserver une liberté d'action très importante.

SDM principalement utilisée en tant que check-list, permet de standardiser les résultats mais pas la manière de faire. En ce qui concerne la documentation, SDM cite les différents documents à réaliser mais ne donne que très peu d'indications quant à leur contenu.

C'est pourquoi le groupe de support a choisi des modèles et méthodes au développement qu'ils ont proposés aux développeurs dans le but d'uniformiser la réalisation de certains composants de la documentation. Pour que l'utilisation de ces modèles et méthodes se généralise, le groupe de support a recherché des outils capables de les supporter et donc de générer des documents. La principale tâche de ce groupe est de diffuser ces outils et de fournir tous les conseils et renseignements nécessaires à leur utilisation.

Dans le deuxième chapitre, sera présentée une critique à priori de la documentation proposée par SDM et qui permettra de déduire la documentation minimale devant être réalisée pour tout projet de développement.

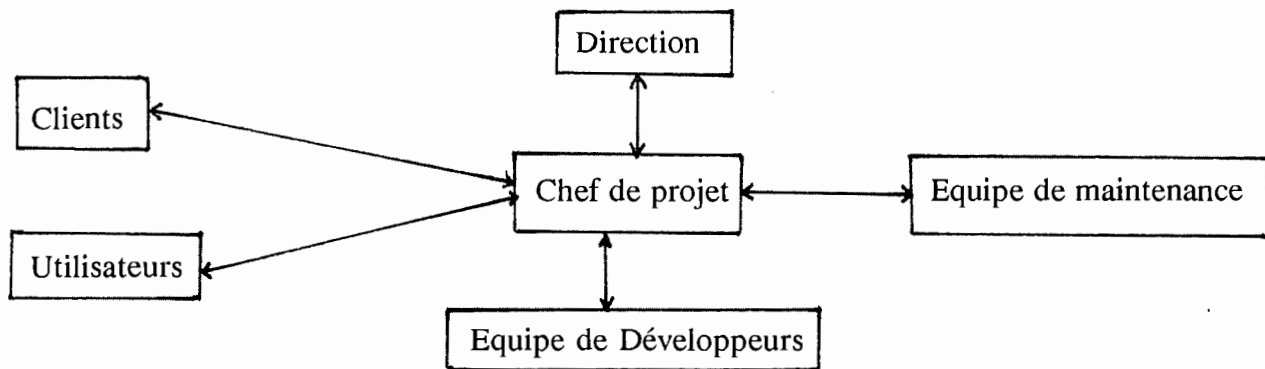
CHAPITRE 2

**Critique théorique de la
documentation proposée par SDM**

2.1 DEMARCHE POURSUIVIE ET CRITERES DE SELECTION

"Nobody reads documentation", tel était le titre "choc" d'un article paru dans les Communications of the ACM de juillet 1991 (M. RETTIG (11)).

Il est vrai que la réalisation de la documentation lors d'un nouveau développement informatique est une activité qui pose encore bon nombre de problèmes dans la plupart des centres informatiques. Trop souvent, l'action de documenter est vue comme une phase isolée et finale du développement d'un système. Par manque de temps et de motivation, on réalise souvent uniquement le strict nécessaire. Or, la documentation doit faire partie intégrante de chaque activité et être contrôlée durant la procédure d'acceptation afin qu'aucun système ne puisse être accepté et transféré sans bonne documentation. Pour montrer la difficulté que représente la documentation, il suffit de montrer les intérêts de chaque partie prenant part au projet de développement :



- Le chef de projet, généralement responsable de la documentation, a un rôle délicat qui consiste à gérer le projet (y compris la réalisation de la documentation) de façon à satisfaire au mieux les exigences parfois contradictoires des différentes parties en présence.
- L'équipe de développeurs qui tentent de réduire les étapes d'analyse autant que possible afin d'arriver rapidement à la phase de codage, de loin la plus appréciée. Un problème majeur est qu'ils se montrent réticents lors de la réalisation de la documentation et en l'absence de contrôle, celle-ci est souvent réduite au minimum et négligée.
- L'équipe de maintenance, contrairement à l'équipe de développement, souhaite avant tout des programmes bien documentés afin qu'ils soient plus faciles à modifier.
A la CGER, la maintenance est généralement prise en charge par le groupe qui a développé le système.
- Les clients exigent que le système soit développé aussi rapidement que possible tout en respectant un budget fixé et sont donc principalement intéressés par les documents de suivi de projet et les analyses coût-bénéfice.
A la CGER, le client est un groupe d'utilisateurs qui sont à l'origine de la

demande de projet et qui en définissent les objectifs, les exigences tout en étant responsables de l'opportunité économique du développement.

- Les utilisateurs désirent voir développer un certain nombre de fonctionnalités répondant à leur besoin, qui soient conviviales, rapides et sûres.
A la CGER, pour chaque projet, un cadre des services utilisateurs est désigné comme promoteur de projet. Il est rendu responsable de toutes les activités nécessaires du côté utilisateur.
- La direction qui apprécie les projets aux objectifs ambitieux mais dont les risques sont limités autant que possible.
A la CGER, une fois la demande de développement accordée, la direction accorde une autonomie très grande aux développeurs.

Dans la méthode SDM, on insiste beaucoup sur la documentation en soulignant son rôle primordial dans la réalisation d'une maintenance rapide et de qualité. Cependant, une critique souvent émise à l'égard de SDM est qu'elle propose de documenter toutes les activités en détail, ce qui représenterait pour les développeurs un travail de rédaction beaucoup trop volumineux étant donné le temps dont ils disposent. De plus, elle est beaucoup trop vague quant à la forme et au contenu de la documentation à réaliser et ne peut donc être utilisée telle qu'elle comme une méthode de documentation.

Dans ce chapitre est proposée une critique à priori de la documentation d'un projet proposée par SDM durant les étapes de développement d'un système, c'est-à-dire de la phase 1 concernant l'analyse conceptuelle à la phase 6 de conversion et de mise en production (la dernière phase étant relative à l'exploitation et à la maintenance du système).

Il est important de souligner que cette critique concerne uniquement des projets développés entièrement au sein de l'entreprise (les projets d'achat de packages ou de systèmes clé sur porte ne seront pas pris en compte).

Chaque phase et chaque activité sera passée en revue et on y distinguera la documentation jugée nécessaire, celle qui peut être utile mais non obligatoire et éventuellement la documentation qui s'avère inutile. Pour chaque documentation conseillée, on mettra en évidence son utilité en la situant par rapport à cinq critères de sélection que nous aurons préalablement fixés. A la fin de chaque phase, un tableau récapitulatif de la documentation sera réalisé.

2.1.1 QU'EST-CE-QUE LA DOCUMENTATION ?

La documentation occupe une place importante dans le développement de projets informatiques. Il suffit pour s'en convaincre de voir le volume de données de toute sorte - documents destinés à informer sur le travail réalisé, résultats de chaque étape, documents utilisés pour réaliser chaque activité - et de tout format - papier, disques, dictionnaire, présentation orale - qui est véhiculé au sein de l'équipe de développement ainsi qu'entre cette équipe et les utilisateurs, le management, les

gens de la production. Pour tenter de structurer toutes ces informations, SDM propose une classification de la documentation en trois grands types:

- la documentation du projet
- la documentation du système
- la documentation historique

1. La documentation du projet :

La documentation du projet concerne principalement les documents qui permettent de diriger, de gérer le projet tels la demande détaillée de projet, les plannings et analyses coût-bénéfice, les rapports d'avancement du projet, les accords et conventions réalisés, les comptes-rendus des discussions, la correspondance ou encore les propositions de modification du travail réalisé, des plannings ou des décisions prises.

Selon le modèle de classification des projets de MAC FARLAN (3) vu précédemment et en considérant uniquement les projets de taille relativement grande, nous pouvons affirmer que plus le projet est risqué, plus la gestion du projet est difficile et moins il est possible de planifier. Ceci s'explique par le fait que l'augmentation du risque est due à une impossibilité croissante de définir correctement et dès le départ les objectifs du projet et/ou au manque de connaissance et d'expérience des développeurs face à la technologie à utiliser. Ces projets comprennent une grande part d'aléas et d'incertitudes qui risquent à tout moment de remettre le travail déjà réalisé en cause et donc de modifier les plannings et les estimations faites.

Exemples :

Pour un projet bien structuré utilisant une technique connue, les risques sont faibles car il est possible de planifier le développement de façon très précise et nous trouverons comme documentation une demande de projet très détaillée, des calendriers et une analyse coût-bénéfice précise ainsi que des rapports d'avancement du projet réguliers.

Par contre, pour un projet peu structuré utilisant une nouvelle technique, les risques sont très élevés et il sera inutile de tenter de planifier le développement. Nous aurons comme documentation les comptes-rendus des discussions, des accords et des conventions prises afin de juger s'il est toujours opportun et acceptable de poursuivre le développement avec la nouvelle technologie compte-tenu des problèmes rencontrés et des dépenses déjà réalisées.

Cette première partie de la documentation du projet relative à la gestion du projet fera généralement l'objet d'un chapitre particulier.

A côté de cela, nous ajoutons les résultats des différentes activités de la méthode.

Cette documentation contient l'information de base pour les activités et phases

suivantes, elle soutient la suite du développement et constitue la base pour la documentation du système. Il est primordial que cette documentation reflète à tout moment la version la plus récente du projet, ce qui signifie que toute modification de l'état actuel du projet est susceptible d'entraîner des mises à jour de la documentation et par conséquent, une migration de la documentation du projet devenue désuète vers la documentation historique. Ce transfert est essentiel si l'on veut éviter le problème de coexistence de versions multiples. SDM propose de structurer cette deuxième partie de documentation du projet relative au développement en suivant les différentes phases de la méthode.

Dès que le projet se termine et que le système est en production, la documentation du projet en entier, c'est-à-dire parties gestion et développement n'a plus de raison d'être et rejoint la documentation historique.

2. La documentation du système :

La documentation du système propose la description du système dans sa version finale et contient toutes les directives et les données nécessaires pour pouvoir utiliser et réaliser la maintenance du système.

La documentation du système naît petit à petit durant le développement du système et se constitue à partir de la documentation du projet. La majeure partie est réalisée durant la phase de programmation et de tests isolés et ce jusqu'à la conversion des données et/ou des programmes et la mise en production où se réalise le contrôle de complétude et de cohérence de la documentation. En effet, ce n'est qu'à partir de ce moment que l'on peut supposer en général que le système ne subira plus de modifications sémantiques importantes.

A l'intérieur de la documentation du système, SDM propose trois catégories différentes :

- **la documentation de l'utilisateur**

La documentation destinée à l'utilisateur lui permet de travailler correctement avec le système en lui permettant d'acquérir les connaissances préliminaires nécessaires.

Exemples :

- * description des écrans et des procédures interactives
- * liste des messages d'erreurs possibles

- **la documentation technique du système**

La documentation technique du système a pour objectif de fournir à l'équipe de maintenance les bases nécessaires pour réaliser au mieux leur tâche. A la CGER, cette documentation contient principalement la documentation des programmes ainsi que de la structure physique des données. Elle ne contient pas de documents réalisés aux niveaux conceptuel et fonctionnel car une modification du système existant atteignant ces niveaux n'est plus considéré

comme un travail de maintenance mais bien comme un développement.

- **la documentation de production**

La documentation de production rend possible l'exécution du système sur machine. Elle contient toutes les instructions dont a besoin le personnel du centre de calcul.

Exemple :

ressources hardware nécessaires, estimation de la durée d'exécution, degré de priorité, ordre de succession des différents modules du programme, fichiers de données nécessaires.

3. La documentation historique :

La documentation historique naît automatiquement durant le développement du système : elle est constituée de toutes les documentations du projet ayant subi des modifications et que l'on souhaite toutefois conserver pour des raisons légales ou encore pour la facturation du projet. Cela est également valable pour la documentation du système durant la phase de maintenance. Cette documentation sera généralement archivée en respectant l'ordre chronologique.

L'activité de documentation n'étant pas une mince affaire, il est nécessaire de désigner des responsables de la documentation. Généralement, durant le développement du système, l'équipe de projet et principalement le chef de projet sont responsables de la documentation du projet et du système. Ensuite, par l'acceptation et le passage en production du système, la responsabilité de la documentation du système est alors léguée au gérant des applications pour la partie utilisateur et au groupe de maintenance pour la partie technique et la documentation de production.

2.1.2 CRITERES DE SELECTION

Les cinq critères de sélection de la documentation utile pour un projet de développement employés sont tirés de la méthode SDM et se basent sur l'utilisation et le rôle de la documentation :

CRITERE DE BON DEROULEMENT DES ACTIVITES FUTURES. (critère n° 1)

Il recouvre les types de documentation qui servent de base au bon déroulement des activités futures : SDM propose une méthode globalement descendante où chaque phase correspond à un niveau de description de plus en plus détaillé. Il en est de même pour la documentation et le développeur chargé de la réalisation d'une ou plusieurs activités se servira de la documentation des activités précédentes pour déterminer le travail qu'il doit faire et s'assurer que ce qu'il fera correspond bien à ce qui a été demandé.

CRITERE DE CONTROLE ET DE GESTION EFFICACE DU PROJET (critère n°2)

Il concerne les types de documentation sans lesquels il serait impossible ou difficile d'assurer un contrôle et une gestion efficace du projet.

D'une part, chaque activité se doit de fournir quelque chose de mesurable et de contrôlable afin de vérifier l'état d'avancement et la qualité du développement. D'autre part, pour gérer efficacement le projet, il est nécessaire de réaliser des plannings, des estimations des coûts et bénéfices ainsi que planifier l'allocation des ressources humaines ou matérielles nécessaires.

CRITERE DE COMMUNICATION ET DE PRISE DE DECISION (critère n°3)

Il s'agit des types de documentation qui facilitent la communication et les décisions à prendre entre toutes les parties prenant part au développement du système : le chef de projet, les clients, l'équipe de travail, les utilisateurs, les dirigeants, mais aussi les tiers (fournisseurs de software, de hardware,...)

CRITERE DE MAINTENANCE EFFICACE DU SYSTEME (critère n°4)

Il concerne les types de documentation qui permettent à d'autres de réaliser une maintenance efficace des programmes réalisés.

CRITERE DE TRANSFERT ET DE BONNE CONNAISSANCE DU SYSTEME (critère n°5)

Il s'agit des types de documentation qui facilitent d'une part le transfert du système à l'équipe chargée de la mise en production et d'autre part, qui assurent une bonne connaissance du système : cela inclut la documentation destinée aux utilisateurs, celle qui permettra aux membres du centre de production d'acquérir une connaissance du système suffisante pour assurer une mise en production satisfaisante.

2.2 CRITIQUE A PRIORI DE LA DOCUMENTATION

Voici maintenant la critique à priori de la documentation de chaque phase de SDM ayant trait au développement. A la fin de chaque phase, un tableau récapitulatif est proposé.

2.2.1 PHASE 1 : ETUDE DE L'EXISTANT ET ANALYSE CONCEPTUELLE

Le but de cette étape est principalement et selon nous de définir les objectifs du projet par l'élaboration d'un **avant-projet de solution** que nous appellerons encore **projet-cadre**.

La réalisation du projet-cadre débute lors de la première activité de l'étape - définir le problème et le champ de l'étude - et se termine par la quatrième activité -

déterminer les objectifs et exigences du système - (nous vous renvoyons à la présentation de SDM page 14).

SDM ne fait pas mention directement d'un tel document et propose de réaliser un document isolé pour chacune des quatre activités . C'est pourquoi, chaque fois que ce sera possible, nous suggérerons des regroupements qui permettront d'alléger le travail de documentation.

La première utilité du projet-cadre se justifie par le critère n°3 (communication) au niveau des rapports entre l'analyste et les demandeurs du projet car le projet-cadre permettra de s'assurer auprès de l'utilisateur qu'on a réellement bien compris le besoin exprimé en organisant des confrontations avec lui et en tenant compte de ses remarques ...

Une fois terminé, le projet-cadre constituera une sorte de **cahier des charges** pour l'analyse fonctionnelle et technique, d'où sa justification par le critère n°1 (activités futures). Finalement, il sera utile, selon le critère n°5 (transfert) à l'équipe de production pour établir les critères d'acceptation du système qui leur permettront de juger régulièrement si le système répond toujours aux objectifs et exigences qui lui ont été fixés.

1.1 Définir le problème et le champ de l'étude

PROCEDE

Il s'agit, au cours de rencontres avec les clients de cerner les besoins et, si le système existe déjà, de déterminer les causes d'insatisfaction ainsi que les points positifs (que l'on pourra conserver) au niveau des activités du système d'information, des technologies utilisées ou de l'organisation du travail.

DOCUMENTATION

PROPOSITION SDM

Pour cette activité, SDM propose de réaliser **un document descriptif** intitulé "définition du problème" et qui contiendra :

- les objectifs du projet
- les problèmes actuels
- l'étendue de l'étude : services et personnes concernés, données recherchées.
- les limites et restrictions auxquelles on doit faire face : limites budgétaires, temporelles, en personnel, organisationnelles, sociales, matérielles.
- les conventions au niveau des rapports à fournir : destinataires, niveau de détail, façon de faire, fréquence.
- le plan de réalisation de la phase.

EVALUATION

Il est certain que tout projet doit être basé sur une mission bien définie ainsi que sur une explication claire du problème. Toutefois, nous savons que cela n'est pas toujours possible.

Pour un projet peu structuré, les développeurs auront besoin d'une collaboration étroite des utilisateurs. C'est pourquoi dans ce cas, ce document rencontre totalement le critère n°3 (communication) car il permet de mettre les personnes concernées au courant en leur présentant ce que l'on envisage de faire, ce que l'on attend d'eux, le temps que ça va durer, et ainsi informés, ils seront normalement plus coopératifs (notamment pour faire accepter le risque que comprend un tel projet par la direction).

Pour un projet bien structuré, ce document, selon le critère n°1 (activités futures) constitue le point de départ du projet et il doit absolument être réalisé avec beaucoup de rigueur car une mauvaise formulation du problème dès le départ peut coûter très cher par la suite.

DESTINATAIRES

Ce document sera un constituant du projet-cadre. Il sera destiné à tous les membres de l'organisation concernés par l'identification du projet, pour rappel, les clients, les utilisateurs, l'équipe de développeurs, la direction et l'équipe de maintenance.

Ce document doit donc utiliser un vocabulaire commun à toutes ces personnes, il ne doit pas faire allusion à la technique pour ne pas perturber l'utilisateur et doit être suffisamment synthétique pour pouvoir être lu par les membres de la direction mais doit tout de même être assez complet pour être utile aux analystes.

1.2 et

1.3 Rassembler les données sur la situation existante, les analyser et les évaluer

1.4 Déterminer les objectifs et exigences du nouveau système

PROCEDE

Il s'agit de rassembler les données utiles concernant aussi bien l'organisation, les activités de l'entreprise que le système existant afin de découvrir les problèmes existants et pouvoir en déduire les objectifs et exigences que devra remplir le nouveau système. Nous avons groupé ces trois activités car elles n'ont qu'un but, celui de définir les exigences et objectifs du nouveau système. L'étude et l'analyse de la situation existante n'étant qu'un moyen d'y parvenir dans les cas où le nouveau système est dépendant de l'ancien ou lorsque les clients n'ont pu déterminer avec suffisamment de précision

leurs attentes.

DOCUMENTATION

PROPOSITION SDM

SDM est générale quant aux données à rassembler et propose seulement des exemples tels les procédures de traitement, les responsables, les souhaits et suggestions d'amélioration, les analyses coût/bénéfice du système actuel, ... Ensuite, elle propose de réaliser un document intitulé "**analyse de la situation existante**" qui décrira les lacunes au niveau de la qualité des informations produites par le système d'information, des performances des traitements au point de vue des délais d'exécution, de la fiabilité ou encore de la sécurité. Ensuite, dans l'activité suivante, elle suggère un document intitulé "**objectifs et exigences du nouveau système**" tout en citant les principales rubriques qu'il devra contenir accompagnées d'un exemple mais sans toutefois aborder la forme du contenu. Ces rubriques sont :

- les objectifs informationnels, organisationnels et économiques du projet.
Ex. : Le personnel actuel doit être capable de travailler avec le nouveau système et le maîtriser.
- les critères d'efficacité du nouveau système.
Ex. : Le nouveau système doit réduire le temps de traitement d'une facture de 5 %
- les limites et contraintes existantes ainsi que les moyens et ressources exigés
Ex. : Le nouveau système ne doit pas utiliser une capacité disque supérieure à celle du système actuel.

Ainsi, SDM présente les principales rubriques que contiendra le document mais n'aborde pas la manière de décrire les informations car chaque analyste adoptera une description des objectifs la plus familière aux utilisateurs avec lesquels il travaille.

EVALUATION

Nous pensons qu'il est superflu de vouloir réaliser deux documents séparés et que ces informations peuvent très bien être regroupées en un document unique intitulé "**objectifs et exigences du système**" qui aurait le même contenu que le document SDM mais auquel on ajouterait une description des problèmes rencontrés dans le système existant et qui servirait à justifier la réalisation d'un nouveau système.

De plus, SDM ne tient pas compte du fait qu'il existe des projets très peu structurés au départ et qui par conséquent ne permettent pas de définir dès maintenant avec certitude les objectifs et les exigences du système développé. Toutefois, quel que soit son degré de précision, ce document est **indispensable** tout d'abord selon le critère n° 1 (activités futures) car il va contribuer à déterminer les solutions possibles susceptibles de rencontrer les objectifs fixés ainsi qu'à déterminer les critères de sélection qui permettront d'évaluer les différentes solutions et d'en sélectionner une.

Ensuite, en accord avec le critère n° 3 (communication et prise de décision) il servira de document de base pour l'acceptation du projet en mettant en avant toutes les améliorations qu'il apportera au système actuel. Il permettra également de discuter avec les utilisateurs afin de s'assurer que tous les problèmes actuels ont bien été soulevés ainsi qu'avec les responsables opérationnels qui jugeront si les limites et exigences du nouveau système sont bien réalistes.

DESTINATAIRES

Ce document clôturera le projet-cadre et sera donc destiné comme le précédent, aux clients, aux utilisateurs, à la direction, à l'équipe de développement et à l'équipe de maintenance.

1.5 Arrêter les points qui restent à résoudre et les hypothèses de base

PROCEDE

Il s'agit de réaliser une liste des problèmes rencontrés et non encore résolus ainsi que des hypothèses de base dont il faudra tenir compte. Cette liste sera gérée et mise à jour jusqu'à ce que le développement soit terminé.

DOCUMENTATION

PROPOSITION SDM

SDM propose de réaliser un document sous forme de liste qui reprend tous les problèmes rencontrés dans l'ordre chronologique et d'y indiquer les dates prévues de résolution, les moyens d'aide possibles, ainsi que les actions à entreprendre ou prises pour y remédier. Une seconde liste énumèrera les hypothèses de base concernant les ressources qui seront utilisées par le système (unités/machine, personnel, capacité disque, ...).

EVALUATION

De manière générale, ce document intitulé "**problèmes non résolus et hypothèses de base**" ne peut être qualifié d'indispensable puisqu'il ne constitue pas un but en soi. L'important étant uniquement de ne négliger aucun problème et de n'oublier aucune hypothèse quelle que soit la façon dont on s'y prend pour les gérer.

Ce document ne sera utile que dans le cas de projets utilisant une nouvelle technologie car de nombreux problèmes techniques risquent de surgir tout au long du développement, problèmes qui seront relativement compliqués et longs à résoudre vu le manque d'expérience des développeurs face à cette technique. Selon le critère n°2 (gestion), cette liste permettra au chef de projet de s'assurer qu'il n'a oublié aucun problème et qu'il n'a négligé aucune hypothèse de base. Ajoutons que dans ce cas-ci, un problème oublié et donc non résolu au stade actuel risque de réapparaître plus tard et de remettre en cause tout le travail réalisé depuis le moment où il est apparu.

DESTINATAIRES

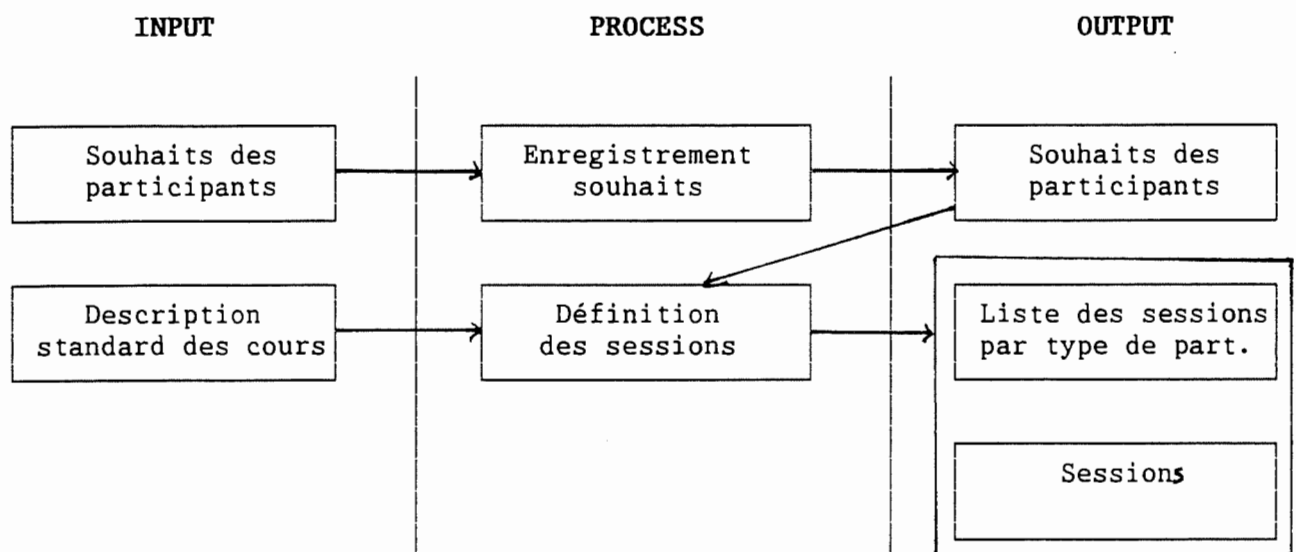
Il s'agit d'un document réalisé par et pour le chef de projet.

1.6 Réaliser un schéma du système

PROCEDE

Cette activité permet, sur base des exigences du système et de l'évaluation de la situation existante, de réaliser un schéma global du nouveau système d'information.

Exemple du schéma global des activités d'un système de "Gestion des sessions de formation à SDM"



DOCUMENTATION

PROPOSITION SDM

SDM propose de réaliser ce schéma en définissant tout d'abord pour chaque application, les entrées et sorties globales en d'en déduire ensuite les fonctions globales nécessaires pour obtenir les sorties à partir des entrées. Pour se faire, elle conseille de faire appel à la technique HIPO déjà vue précédemment.

EVALUATION

Vu le caractère global de ce schéma, lorsqu'il s'agit de la refonte d'un ancien système, il est tout à fait possible que le schéma global de l'ancien système puisse encore être celui du nouveau et cette activité devient alors superflue.

Ce document intitulé "**schéma global du nouveau système**" est **indispensable** selon le critère n°3 (communication et décision) car il facilite la discussion (un dessin vaut mieux que mille mots) avec les utilisateurs qui verront apparaître les différentes fonctionnalités qui seront développées au sujet desquelles ils pourront faire des commentaires et détecter les éventuelles anomalies de type structurelles ou fonctionnelles.

De plus, selon le critère n°1 (activités futures), ce document permet de visualiser les données en entrée et donc d'élaborer par la suite une structure des besoins en information. Au niveau des traitements, il servira de base pour la décomposition hiérarchique des traitements qui aura lieu durant l'analyse fonctionnelle.

DESTINATAIRES

Les utilisateurs et l'équipe de développement.

- 1.7 Déterminer les outils et les solutions possibles**
- 1.8 Evaluer les solutions et opérer une sélection**

PROCEDE

Il s'agit d'élaborer des solutions possibles répondant aux exigences minimales du nouveau système puis de déterminer une procédure de sélection basée sur des critères de choix (ils porteront sur le coût, les aspects de développement, le nombre de fonctions, la flexibilité, la sécurité, les délais, la qualité des résultats, ...). Chaque solution sera évaluée et la meilleure sera retenue.

DOCUMENTATION

PROPOSITION SDM

SDM estime qu'il est **indispensable** de réaliser un document qui regroupera l'énumération des solutions possibles avec leurs avantages et inconvénients, une présentation des facteurs d'évaluation des solutions, de la procédure de sélection suivie ainsi que les résultats de l'évaluation des solutions par rapport aux facteurs d'évaluation.

EVALUATION

Il est certain que pour un projet devant être développé avec les moyens existants, le problème de choix d'une solution disparaît et le travail se borne à contrôler que les moyens existants sont en nombre suffisant et qu'ils seront disponibles en temps voulu.

Dans le cas d'un projet impliquant le choix d'une nouvelle technologie le document intitulé "**solutions alternatives et sélection**" est **indispensable** car, selon le critère n°3 (communication et décision), il servira à bien déterminer les avantages et inconvénients de chaque solution, d'en déterminer le facteur risque qui constituera le facteur principal pour l'acceptation de la solution proposée par la direction.

Dans le cas d'une solution entraînant des changements dans la façon de travailler des utilisateurs, ce document, toujours selon le même critère permettra de les informer suffisamment tôt pour qu'ils puissent s'organiser et réaliser si nécessaire des sessions de formation.

Ce document mérite beaucoup d'attention car le choix d'une mauvaise solution entraîne par la suite des coûts supplémentaires considérables ainsi que des retards importants.

Nous ajouterons que ce document se prête très bien à une standardisation. En effet, la procédure de sélection ainsi que les critères de choix sont généralement standardisés au sein de l'entreprise.

Le document réalisé se présentera sous la forme d'un tableau où les alternatives seront comparées et classées en suivant une procédure de valorisation des différents facteurs d'évaluation.

La figure ci-dessous présente la procédure de sélection standard à la CGER : la méthode de Benson.

FACTEUR D'APPRECIATION	ESTIME PAR	VALEUR 0.....5	POIDS	TOTAL
INTEGRATION A LA STRATEGIE			20	
SOURCE D'UN AVANTAGE COMPETITIF			16	
SOURCE D'INFORMATION DE GESTION			10	
PREVENTION D'UN HANDICAP SUR LE PLAN COMPETITIF			10	
PREVENTION D'UN RISQUE ORGANISATIONNEL			8	
STRATEGIQUE POUR L'ARCHITECTURE DU SYSTEME D'INFORMATION			12	
PREVENTION DES INCERTITUDES QUANT AUX SPECIFICATIONS			9	
PREVENTION QUANT AUX INCERTITUDES TECHNIQUES			9	
PREVENTION DU RISQUE LIE A L'INFRASTRUCTURE "IS"			6	
TOTAL PONDERE DES APPRECIATIONS			100	
RETURN ON INVESTMENT (ROI)				

- FIGURE 8 : METHODE DE BENSON -

DESTINATAIRES

Le comité de direction, les utilisateurs et l'équipe de développement.

1.9 Déterminer les problèmes de conversion et de mise en production

PROCEDE

Les problèmes de conversion concernent la réutilisation des données et/ou de programmes de l'ancien système dans le nouveau tandis que les problèmes de mise en production concernent principalement les séances de formation qui devront être dispensées aux utilisateurs afin qu'ils s'adaptent correctement au nouveau système.

SDM propose également de fixer à ce stade et en collaboration avec les utilisateurs, les critères d'acceptation du nouveau système qui seront utilisés durant le test d'acceptation du système.

DOCUMENTATION

PROPOSITION SDM

SDM propose de réaliser un plan général des opérations de conversion ainsi qu'un document qui présente les critères d'acceptation et qui sera utilisé par les utilisateurs chargés de réaliser le test d'acceptation du nouveau système.

EVALUATION

Les problèmes de conversion de données et de programmes s'avèrent très complexes lorsqu'il s'agit d'un projet faisant appel à une nouvelle technologie ou à un nouveau langage de programmation. Et nous pensons que cette activité dans le cadre d'un tel projet est à ce point complexe et longue qu'il est préférable d'en faire un projet particulier avec sa problématique propre. Au contraire, lorsqu'il s'agit d'un projet utilisant les moyens existants, il peut s'avérer nécessaire de convertir certaines données mais ce travail sera limité et relativement simple à réaliser et c'est pourquoi nous pensons que les problèmes de conversion à prévoir peuvent très bien être intégrés au document "**problèmes non résolus**" de l'activité 1.5 - arrêter les points non résolus et les hypothèses de base -.

Au niveau de la mise en production, la préoccupation principale est la formation des utilisateurs du nouveau système et nous pensons que le même raisonnement peut s'appliquer à ce problème. Ajoutons simplement que dans les entreprises de grande taille, la formation est généralement prise en charge par une équipe particulière et constitue donc un projet séparé.

1.10 Planning global et vue d'ensemble des coûts et bénéfices

PROCEDE

Il s'agit de réaliser globalement un planning des phases ultérieures et en particulier de l'analyse fonctionnelle.

DOCUMENTATION

PROPOSITION SDM

SDM propose que ce planning soit inclu dans un document qui fournira une estimation des durées et échéances des étapes futures, une analyse coût-bénéfice ainsi que les besoins en ressources humaines et matérielles.

EVALUATION

Nous avons remarqué que SDM ne reconnaît pas le fait que certains projets à haut risque sont pratiquement impossibles à planifier vu leur manque de prévisibilité : il s'agit de projets dont les objectifs sont mal définis et qui font appel à une nouvelle technologie. Ces projets à haut risque ne sont confiés qu'à des chefs de projet exceptionnels. Le développement d'un tel projet ne se fait pas sans mal et peut susciter des retours en arrière importants. Or, SDM ne tient pas compte de tels projets car en respectant le modèle de développement "waterfall" vu au

premier chapitre, elle considère que tout développement respecte le parcours séquentiel des différentes phases avec des retours en arrière limités à la phase précédente et qui ne peuvent se faire qu'après en avoir introduit la demande selon une procédure bien définie. Nous pouvons dire qu'il s'agit là d'un manque manifeste de flexibilité de SDM.

Si nous considérons les projets à risque relativement faible, il est certain en accord avec le critère n°2 (gestion) que la réalisation d'un document intitulé "**planning global**" est indispensable pour contrôler le travail de l'équipe de développement et faire en sorte que les délais et budget initialement prévus soient respectés.

Pour des projets bien définis, il existe des outils de planification puissants et très complets (par exemple Superproject présenté p 22) qui proposeront au chef de projet un éventail assez large de techniques grâce auxquelles il réalisera un plan de développement, une analyse coût-bénéfice, un calendrier des points de décision à prendre ainsi qu'une estimation des besoins en personnel et en ressources informatiques très précis.

DESTINATAIRES

Les clients, l'équipe de développement et la direction.

1.11 Rédaction du rapport d'analyse conceptuelle

PROCEDE

Rédiger le rapport final de l'analyse conceptuelle consiste à réaliser la synthèse des résultats de cette première phase. Ce dossier une fois approuvé constituera le point final de la première étape et figurera dans le dossier de documentation du projet.

DOCUMENTATION

PROPOSITION SDM

Selon SDM, le contenu du rapport final devra contenir un résumé des principaux résultats de la phase et être capable de répondre aux questions suivantes :

- Le problème est-il bien défini ?
- La suite du développement a-t-elle un sens ?
- Si oui, comment cela va-t-il se passer?
- Quels problèmes devront être résolus?
- Quels sont les coûts et bénéfices estimés du nouveau système?

Idéalement, ce document sera divisé en deux parties : un résumé pour la direction et les informations détaillées pour les utilisateurs et l'équipe de développement.

EVALUATION

Pour SDM et conformément au modèle de développement "Waterfall" qu'elle applique, la condition indispensable pour passer d'une phase de développement à la suivante est l'existence d'une documentation complète de cette phase. C'est pourquoi ce document intitulé "**rapport d'analyse conceptuelle**" est **indispensable** selon le critère n°3 (décision), puisqu'il permet aux différentes parties prenant part au projet de juger si leurs exigences respectives sont bien remplies et si on peut passer à la phase d'analyse fonctionnelle.

Ensuite, selon le critère n°1 (activités futures), il servira de base à la réalisation de l'analyse fonctionnelle car il présentera une formulation correcte du problème, ne laissant place qu'à un minimum d'ambiguïté (en théorie, aucune) afin que le nouveau système réponde bien aux besoins.

Cependant, ce document est relativement redondant puisqu'il consiste à résumer les documents réalisés durant cette étape. Nous pouvons dès lors penser à automatiser sa création tout au long de l'étape. Ceci est possible à condition que la documentation de l'étape soit entièrement réalisée par un seul outil ou alors que les différents outils utilisés soient parfaitement intégrés, ce qui n'est pas certain pour tous les centres de développement.

DESTINATAIRES

Les utilisateurs, les clients, l'équipe de développement et la direction.

TABLEAU RECAPITULATIF

Phase 1 : Analyse Conceptuelle

Activité	Document	I ¹	U ²	S ³	Critère n°1	Critère n°2	Critère n°3	Critère n°4	Critère n°5
1.1 Définition du problème		X			XX		XXX		
1.2 Analyse de la situation & 1.3 existante et définition & 1.4 des objectifs et exi- gences du système		X			XXX		XX	XX	
1.5 Problèmes non résolus et hypothèses de base				X		X			
1.6 Schéma global du système		X			XXX		XXX		
1.7 Solutions alternatives & 1.8 et sélection		X							
1.9 Problèmes de conversion et de mise en production				X					
1.10 Planning global						XXX			
1.11 Rapport d'analyse conceptuelle					XX		XXX		

I¹ = indispensable

U² = utile

S³ = superflu

Critère n°1 : bon déroulement des activités futures

Critère n°2 : contrôle et gestion efficace du projet

Critère n°3 : communication et prise de décision

Critère n°4 : maintenance efficace du système

Critère n°5 : transfert et bonne connaissance du système

2.2.2 PHASE 2 : ANALYSE FONCTIONNELLE DETAILLEE

Le but de cette étape est de produire les spécifications d'une solution fonctionnelle détaillée, la plus complète possible mais aussi indépendante que possible de tout moyen de réalisation. On se basera pour se faire sur les résultats de l'analyse conceptuelle. Du point de vue de la documentation, le noyau indispensable de l'analyse fonctionnelle est constitué de :

- Au niveau des traitements : le schéma de la découpe du système en sous-systèmes, la description des entrées, sorties et traitements de chaque sous-système ainsi que les interfaces entre sous-systèmes.
- Au niveau des données : le modèle logique des données.
- Au niveau des procédures manuelles : la définition des interfaces homme/machine nécessitées par chaque sous-système.

2.1 Spécification des exigences du système y compris les exigences futures

2.2 Détermination du cadre de fonctionnement du nouveau système

PROCEDE

Il s'agit de détailler les exigences actuelles et futures du nouveau système ainsi que son cadre de fonctionnement, c'est-à-dire la structure des départements concernés, la méthode de travail, les moyens nécessaires, les aspects de sécurité et de confidentialité.

DOCUMENTATION

PROPOSITION SDM

SDM propose de se baser sur les résultats des activités 1.4 - déterminer les objectifs et exigences du système - et 1.8 - choix d'une solution - de l'analyse conceptuelle et de les détailler davantage en fonction des commentaires et critiques émis lors du point de contrôle en fin d'analyse conceptuelle.

EVALUATION

Pour les projets peu risqués, nous estimons que ces deux mises à jour sont superflues étant donné que, d'une part, les exigences du nouveau système sont précisées dès l'élaboration du projet-cadre et que d'autre part, le cadre de fonctionnement du système a déjà été défini lors de la présentation de la solution retenue. Pour un projet risqué, les résultats de l'analyse fonctionnelle sont beaucoup plus vagues et susceptibles d'importantes modifications à la fin de cette première phase. Il est alors nécessaire de mettre à jour les objectifs du système dans le projet-cadre. Cela se justifie par le critère n° 1 (activités futures) car le développement ne peut se dérouler correctement s'il ne repose pas sur une définition

exacte des objectifs poursuivis. Le projet-cadre modifié sera à nouveau soumis aux parties intéressées pour accord.

2.3 Diviser le système en sous-systèmes et les décrire.

2.4 Définition de l'input et de l'output de chaque sous-système ainsi que des interfaces

2.5 et

2.6 Elaboration des diagrammes des traitements et description des traitements

PROCEDE

Cette étape centrale basée sur le schéma global du système élaboré durant la première phase, consiste à décomposer le système en sous-systèmes indépendants qui seront développés séparément. On décrira ensuite les entrées, sorties et traitements de chaque sous-système ainsi que les interfaces entre les sous-systèmes.

DOCUMENTATION

PROPOSITION SDM

SDM invite à réaliser un document intitulé "**schéma de la découpe en sous-systèmes et descriptions**" qui sera un composant fondamental de la documentation du projet. Pour se faire, SDM cite les modèles et techniques pouvant être utilisés tels HIPO, SHAT, tables de décision.

EVALUATION

Ce document est absolument indispensable (selon le critère n° 1: activités futures) pour poursuivre le développement et principalement la réalisation de la structure logique des données et la description des programmes. En outre, le document constituera une base de discussion avec les utilisateurs (en accord avec le critère n° 3 (communication)) au sujet de la définition des besoins en information ainsi que les interfaces homme/machine utilisées par le système.

DESTINATAIRES

L'équipe de développement et les utilisateurs.

2.7 Spécification des exigences en matière de sécurité et de confidentialité

PROCEDE

Il s'agit de reprendre les définitions des mesures de sécurité et de confidentialité réalisées lors de l'analyse conceptuelle et de les détailler au niveau des sous-systèmes désormais connus.

DOCUMENTATION

PROPOSITION SDM

SDM propose de réaliser pour chaque sous-système, une liste de tous les incidents possibles lors de son exécution, de leurs conséquences et probabilité de réalisation. Sur base de cette liste, on réalisera un document intitulé "**exigences détaillées en matière de sécurité et de confidentialité**" qui présentera sous forme de règles toutes les mesures de sécurité à prendre pour éviter les incidents cités dans la liste.

Exemple : la reconstruction des fichiers âgés de moins de trois mois doit être possible endéans une journée.

EVALUATION

Nous pensons qu'il est superflu de vouloir définir les mesures de sécurité indépendamment des technologies qui seront utilisées. Souvent, le développeur étudie toutes les possibilités qu'offrent les technologies utilisées en matière de sécurité et retient celles qui lui paraissent nécessaires. Or, pour un projet utilisant une technique bien connue, les mesures de sécurité que l'on peut prendre sont bien connues dès le départ et peuvent être correctement définies lors de la détermination des exigences du système à l'étape 1.4 - objectifs et exigences du système -. Pour un projet utilisant une nouvelle technologie, la définition des mesures de sécurité sera plutôt réalisée lors de l'analyse technique lorsque les limites de la technologie seront mieux connues.

2.8 Détermination des problèmes humains et de leur solution

PROCEDE

Cette activité a pour objectif de prévoir quels seront les impacts du nouveau système sur la structure de l'organisation, sur la méthode de travail des futurs utilisateurs ainsi que les compétences supplémentaires que devront éventuellement avoir les utilisateurs. L'objectif est de préparer au mieux les futurs utilisateurs afin qu'ils s'adaptent facilement au nouveau système. Cette activité ne sera donc réalisée que pour les projets utilisant une technologie nouvelle pour l'utilisateur et qui entraîne des modifications importantes dans son environnement de travail et dans sa manière de travailler. Aucune documentation particulière n'est envisagée durant cette activité.

2.9 Conception de la structure logique des données

PROCEDE

Cette activité consiste à réaliser le schéma logique des données en se basant sur le schéma conceptuel des données (réalisé à l'étape 1.6) ainsi que la description des besoins en information des différents sous-systèmes (définis à l'étape 2.4).

DOCUMENTATION

PROPOSITION SDM

SDM estime que le résultat de cette activité intitulé "**schéma logique des données**" constitue la documentation principale des besoins en information du système. Le document contiendra le schéma accompagné d'une indication des accès nécessaires ainsi que d'une quantification de ces accès. SDM laisse le développeur libre de choisir la technique de représentation pour ce schéma.

EVALUATION

Nous sommes tout à fait d'avis que ce document constitue avec le document "schéma de la découpe en sous-systèmes et descriptions" de l'activité 2.3, la documentation principale du système. Il constitue selon le critère n°1 (activités futures) une étape inévitable pour la réalisation des bases de données puisqu'il permettra de réaliser le schéma physique des données lors de la phase d'analyse technique.

DESTINATAIRES

L'équipe de développement.

2.10

/12 Spécifier les facilités requises en communication de données, en hardware et en software

PROCEDE

Il s'agit de déterminer toutes les ressources hardware, software et en télécommunication qui seront nécessaires lors de la mise en production du système.

DOCUMENTATION

PROPOSITION SDM

SDM estime qu'il est temps à ce stade d'avancement du projet de réaliser un cahier des charges complet qui permettra aux personnes responsables des machines, logiciels de base et moyens de télécommunication, d'effectuer les réservations nécessaires ainsi que de lancer les appels d'offre si des achats s'avèrent nécessaires.

EVALUATION

Ce cahier des charges est en fait le prolongement des activités 1.7 et 1.8 - déterminer les outils et les solutions possibles - et - évaluer les solutions et opérer une sélection - où une première version succincte du cahier des charges a été réalisée. Nous pensons que ce document est **indispensable** selon le critère n° 1 (activités futures) car il est primordial que toutes les ressources matérielles nécessaires soient disponibles pour réaliser les phases de programmation, de conversion et de mise en production. Aucun oubli ne peut être toléré dans ce document sous peine d'entraîner des retards importants lors de la réalisation du système. Ajoutons que dans le cas d'un projet nécessitant des achats importants, le cahier des charges doit être réalisé le plus tôt possible car la période se situant entre le lancement des premiers appels d'offre, le choix des fournisseurs et finalement les délais de livraison peut s'avérer très longue.

DESTINATAIRES

Les responsables du matériel et des logiciels, le responsable des achats. A la CGER, la responsabilité du matériel et des logiciels est du ressort de groupes bien particuliers.

2.13 Elaborer un plan pour la poursuite du développement et la mise en production

PROCEDE

Cette activité consiste à réaliser un planning détaillé de la phase d'analyse technique et de préciser le planning des phases ultérieures.

DOCUMENTATION

PROPOSITION SDM

Le système à développer étant bien déterminé en fin d'analyse fonctionnelle, il est possible de réaliser un document intitulé "**planning détaillé de l'analyse technique et de réalisation du système**" qui contiendra une description des activités à réaliser, une estimation minimale et maximale

des dates de début et de fin de chaque activité, le personnel nécessaire et la manière dont ils se répartiront les tâches et les responsabilités, une estimation des charges ainsi que la fixation des points de contrôle. De plus, les besoins en logiciel, matériel et télécommunication étant désormais connus, il est intéressant de réaliser une nouvelle analyse coûts/bénéfices beaucoup plus pertinente.

EVALUATION

La suite du développement va maintenant être divisée en un certain nombre de sous-projets correspondants aux différents sous-systèmes définis et vont être répartis entre les différents programmeurs de l'équipe. Cette nouvelle répartition des responsabilités va rendre plus difficile la tâche du chef de projet qui devra contrôler simultanément l'avancement des différents sous-projets et répartir les ressources de manière à ce que les différentes parties soient toutes terminées pour la date prévue pour le test d'intégration. On comprend dès lors aisément qu'il soit **indispensable** en raison du critère n°2 (contrôle et gestion) de réaliser des plannings aussi détaillés que possible. Il faut noter toutefois que pour un projet utilisant une nouvelle technologie, les incertitudes au niveau de la phase d'analyse technique sont trop importantes pour envisager de réaliser un tel planning.

DESTINATAIRES

L'équipe de développement pour le planning et la direction, les clients pour la partie coûts-bénéfices.

2.14 Rédaction du rapport de l'analyse fonctionnelle

PROCEDE

L'analyse fonctionnelle se termine par la rédaction d'un **rapport final** qui contiendra une synthèse des résultats de cette phase et qui permettra après approbation, de passer à l'analyse technique du projet. Ce document ira rejoindre le rapport de l'analyse conceptuelle dans le dossier de documentation du projet.

DOCUMENTATION

PROPOSITION SDM

Comme pour le rapport final de la première phase, SDM propose de structurer ce document en deux parties: une synthèse des résultats pour la direction et une présentation bien détaillée des résultats pour l'équipe de développement. De plus, SDM souligne le fait que les résultats de l'analyse fonctionnelle peuvent être différents de ceux de l'analyse

conceptuelle en raison des différentes remarques qui auront été émises lors du point de contrôle qui se situe entre les deux phases. C'est pourquoi SDM invite à décrire ces différences dans ce rapport.

EVALUATION

Nous ne sommes pas d'accord avec la méthode lorsqu'elle reconnaît des différences entre les résultats finaux des deux premières phases car cela va à l'encontre du principe de cohérence de la documentation. Nous pensons que l'apparition de différences doit obligatoirement s'accompagner d'une mise à jour de tous les documents concernés déjà existants. Après cette importante remarque nous estimons qu'il est **indispensable**, en raison du critère n°3 (communication et décision) de réaliser un rapport de fin d'analyse fonctionnelle sur base duquel les différentes parties prenant part au projet décideront s'il est opportun de passer à l'analyse technique. Pour cela, ils jugeront si les spécifications du système sont suffisamment détaillées et si les nouvelles estimations des plannings et des coûts satisfont toujours le client et la direction. Il est également indispensable selon le critère n°1 (activités futures) puisqu'il va constituer l'information de base pour l'analyse technique. Notons que comme ce document ne fait que reprendre les différents résultats de la phase, on peut très bien envisager de le réaliser automatiquement afin d'éliminer toute redondance.

DESTINATAIRES

Les clients, les utilisateurs, l'équipe de développement et la direction.

TABLEAU RECAPITULATIF

Phase 2 : Analyse fonctionnelle

<u>Activités</u>	<u>Documents</u>	<u>Utilité</u> ¹	<u>Critères</u> ²				
			1	2	3	4	5
2.1 & 2.2	Détail des exigences et du cadre de fonctionnement du système	S					
2.3 & 2.4 & 2.5/6	Schéma de la découpe en sous-systèmes et descriptions	I	***		**		
2.7	Exigences détaillées en matière de sécurité et de confidentialité	S					
2.9	Schéma logique des données	I	***				
2.10/12	Cahier des charges complet	I	***				
2.13	Planning détaillé de l'analyse technique	I		***			
2.14	Rapport final de l'analyse fonctionnelle		***		***		

¹ Utilité : S = Superflu U = Utile I = Indispensable

² Critères :

Critère n°1 : Critère de bon déroulement des activités futures

Critère n°2 : Critère de contrôle et de gestion efficace du projet

Critère n°3 : Critère de communication et de prise de décision

Critère n°4 : Critère de maintenance efficace du système

Critère n°5 : Critère de transfert et de bonne connaissance du système

2.2.3 PHASE 3 : ANALYSE TECHNIQUE

Sur base des résultats de l'analyse fonctionnelle qui ont permis de déterminer en détail ce que le système doit faire, on va déterminer **comment** le système doit travailler, c'est-à-dire présenter une solution correcte, efficace et exécutable sur une machine réelle. Les deux résultats principaux de cette phase sont la définition de la structure physique des données ainsi que la spécification des programmes.

3.1 Conception des procédures manuelles

PROCEDE

Les procédures manuelles permettent de décrire toutes les tâches qui seront accomplies par l'utilisateur lors de l'utilisation du système.

DOCUMENTATION

PROPOSITION SDM

SDM insiste fort sur la réalisation d'un "**manuel d'utilisation**" qui fournira aux utilisateurs toutes les informations et directives nécessaires à la bonne utilisation du système. Cette documentation sera constituée d'une description du matériel à utiliser, d'une présentation des différentes fonctionnalités offertes par le système ainsi qu'une définition des procédures relatives à l'introduction des données, à l'interprétation des données en sortie ou encore à la correction des erreurs de l'utilisateur.

EVALUATION

Les manuels d'utilisation sont certes **utiles** selon le critère n°5 (connaissances) pour permettre à l'utilisateur de bien connaître le système qu'il va devoir utiliser. Toutefois nous ne sommes pas certains que cette tâche doit être confiée aux développeurs car ceux-ci n'ont pas du tout la même vision du système et les manuels qu'ils réalisent ne répondent pas toujours aux besoins des utilisateurs : ils sont souvent trop volumineux et trop peu maniables pour être utilisés quotidiennement et ne servent que dans les premiers temps lors de l'apprentissage du système. Une solution intéressante serait de faire écrire ces manuels par un utilisateur intégré à l'équipe de développement mais elle n'est pas envisagée à la CGER en raison du coût supplémentaire qu'elle engendrerait. D'autres moyens tels l'aide en ligne ou les séances de démonstration s'avèrent plus intéressantes.

DESTINATAIRES

Les personnes chargées de la formation et les utilisateurs.

3.2 Conception des formulaires et de l'ensemble des input et output de l'ordinateur

PROCEDE

Cette activité est le prolongement de l'activité 2.4 - définir l'input et l'output par sous-système et les interfaces - où on avait déterminé le but et le contenu des différentes interfaces homme/machine du système (écrans, formulaires, ...). A présent, on va fixer leur aspect physique.

DOCUMENTATION

PROPOSITION SDM

SDM propose de réaliser en collaboration avec l'utilisateur un document intitulé "**input et output de l'ordinateur**" qui décrira l'aspect physique de chaque entrée et sortie de l'ordinateur.

EVALUATION

Cette documentation est **indispensable** pour l'utilisateur car il s'agit pour lui de la seule partie visible du système. Selon le critère n°3 (communication et décision), ce document permettra de connaître l'avis de l'utilisateur concernant les conventions prises au niveau des noms ainsi que les aspects ergonomiques des différentes interfaces. Ensuite, ce document servira de base, en accord avec le critère n°1 (activités futures), pour la réalisation proprement dite des interfaces. Actuellement, la documentation papier des interfaces à l'écran est de plus en plus remplacée par des prototypes réalisés grâce à des générateurs d'écrans à partir de spécifications relativement sommaires. L'avantage est qu'il est beaucoup plus facile de discuter avec l'utilisateur sur base d'un modèle concret plutôt que sur des spécifications.

DESTINATAIRES

Les utilisateurs et l'équipe de développement.

3.3 Conception de la structure de stockage

PROCEDE

Cette activité qui consiste à réaliser le schéma physique des données constitue la dernière étape avant la réalisation proprement dite des bases de données. Il s'agit en fait d'adapter le schéma logique des données en fonction de la technologie choisie pour réaliser les bases de données tout en tenant compte des aspects de sécurité, de confidentialité, des exigences et critères de performance du système d'information ainsi que des limites et

possibilités des moyens de support, appareils et programmes disponibles.

DOCUMENTATION

PROPOSITION SDM

SDM propose que le résultat fasse l'objet d'un document intitulé "**schéma physique des données**" qui contiendra une description de tous les fichiers de données à réaliser et de tous les records contenus dans ces fichiers. Pour ce faire, SDM ne propose aucun modèle ni aucune technique de présentation.

EVALUATION

Ce document est **indispensable** selon le critère n°1 (activités futures) puisque c'est sur lui que va reposer la réalisation des bases de données. De plus, la structure physique des bases de données est variable dans le temps : volume des données, nécessité de nouveaux accès, ... et une bonne documentation est indispensable selon le critère n°4 (maintenance) pour que les modifications ultérieures puissent se faire aisément.

DESTINATAIRES

L'équipe de développement et l'équipe de maintenance.

3.4 Conception des mesures de sécurité

PROCEDE

Il s'agit de concevoir en détail toutes les mesures de sécurité et de confidentialité qui devront être prises au niveau des appareils, des réseaux de communication de données, des programmes, des supports de données, du comportement des personnes, ... lors de la réalisation et de la mise en production du système.

DOCUMENTATION

PROPOSITION SDM

SDM propose de réaliser un document intitulé "**mesures de sécurité concrètes**" qui aura la forme d'une liste de toutes les mesures de sécurité à prendre à tous les niveaux, que ce soit pour les programmes, les bases de données, les actions humaines, les réseaux, les accès aux locaux, ...

EVALUATION

Pour un projet de taille respectable, la réalisation d'une telle liste représente un travail énorme compte-tenu de toutes les mesures à prendre. C'est pourquoi nous pensons qu'elle ne doit pas faire l'objet d'un document particulier. La conception des mesures de sécurité relatives aux bases de données sera réalisée en même temps que le schéma physique des données, les mesures relatives aux programmes feront partie de l'activité 3.7 concernant la description des programmes, les mesures concernant le comportement des personnes feront partie de la définition des procédures manuelles et ainsi de même pour tous les types de mesures. Toutefois, il est utile selon le critère n°5 (connaissance du système) de réaliser un document de consolidation où toutes les mesures dispersées sont synthétisées.

DESTINATAIRES

L'équipe de développement.

3.5 et

3.6 **Elaboration des descriptions de programmes et schémas**

PROCEDE

Il s'agit de spécifier la logique des différents programmes à écrire ainsi que les données qu'ils vont utiliser.

DOCUMENTATION

PROPOSITION SDM

SDM propose de réaliser un document intitulé "**spécifications des programmes**" qui présentera la manière dont l'application se divise en programmes eux-mêmes divisés en un certain nombre de modules. Ensuite, SDM invite à écrire les spécifications de chaque programme en pseudo-code afin de bien montrer sa logique. Elle suggère également de bien spécifier les variables qui seront échangées entre les différents programmes, les paramètres et les constantes. SDM consacre un chapitre aux différentes techniques (pseudo-code, diagrammes de Nassi-Schneiderman, tables de décision, ...) pouvant être utilisées pour réaliser les spécifications des programmes.

EVALUATION

Pour un projet de grande taille des spécifications telles qu'elles sont proposées par SDM (pseudo-code ou diagrammes) sont difficilement envisageables vu le grand nombre de programmes qu'il contient d'autant

plus que les programmeurs sont souvent sensés pouvoir encoder à partir des spécifications fonctionnelles des traitements. Cependant, l'absence totale de spécifications permet à chaque programmeur de coder à sa manière, ce qui risque de poser de gros problèmes lorsque l'équipe de maintenance devra modifier le programme et il n'est pas rare que des programmes ne nécessitant que de petites adaptations soient totalement réécrits faute de documentation suffisante. C'est pourquoi nous estimons qu'une documentation minimale des programmes est **indispensable** en fonction du critère n°4 (maintenance). En ce qui concerne la manière de réaliser ces spécifications, nous resterons plus général en exigeant simplement qu'elles soient suffisamment claires pour ne permettre aucune ambiguïté concernant la logique du programme et les données qu'il manipule.

DESTINATAIRES

L'équipe de développement et l'équipe de maintenance.

3.7 Description des programmes standards à utiliser

PROCEDE

Il s'agit de faire la liste de tous les programmes déjà existants pouvant être réutilisés ainsi que les programmes standards qu'il est possible d'acquérir auprès des maisons de software. L'acquisition de tels programmes peut remettre certaines exigences des utilisateurs en question. Il faut donc être très prudent dans les choix car les exigences standards auxquelles répond le logiciel peuvent s'avérer insuffisantes aux yeux des futurs utilisateurs. Cette activité ne nécessite pas la réalisation de documentation particulière.

3.8 Elaboration d'un plan détaillé de programmation et de test

PROCEDE

Une fois arrivé à la fin de l'analyse technique, il s'agit de réaliser un planning détaillé de la phase suivante, la phase de programmation qui inclut le codage et les tests des différents programmes.

DOCUMENTATION

PROPOSITION SDM

SDM fournit la liste des différents constituants du plan de programmation et de test :

- un planning temporel précis
- une répartition des tâches entre les membres de l'équipe de projet

- le procédé de test à suivre
- les facilités de test nécessaires (machine, disque,...)
- la manière de simuler une situation pratique
- la manière d'analyser les résultats de test

EVALUATION

Nous pensons en accord avec le critère n° 2 (contrôle et gestion), qu'il est **indispensable** de réaliser un plan précis de la phase de programmation et de tests isolés car le premier objectif des développeurs est bien de fournir au client et aux utilisateurs des programmes bien écrits, qui respectent les spécifications faites et qui contiennent le moins d'erreurs possible (en théorie, aucune). Ce plan proposera à chaque programmeur le planning à respecter pour coder et tester le programme dont il est responsable; ce planning devra bien tenir compte de la complexité de chaque programme, du langage de programmation utilisé ainsi que des connaissances et de l'expérience de chaque programmeur afin que les délais imposés ne l'obligent pas à négliger la lisibilité de ces programmes ou à passer certains cas de test faute de temps. En ce qui concerne la description des procédés de test et d'analyse des résultats, nous pensons que ce n'est pas un bon choix de les documenter au niveau du plan de test car ces procédés sont suffisamment stables que pour être standardisés pour tous les développements de l'équipe de projet ou même du département informatique.

DESTINATAIRES

L'équipe de développement, l'équipe de production et la direction.

3.9 Rédaction du rapport d'analyse technique

PROCEDE

La rédaction du rapport d'analyse technique consiste à résumer les principaux résultats de cette phase ainsi qu'à présenter les problèmes spécifiques rencontrés.

DOCUMENTATION

PROPOSITION SDM

SDM estime que ce document est surtout important dans les organisations où la responsabilité et le travail des analystes-système et des programmeurs sont nettement séparés ou bien lorsque la programmation et les tests sont confiés à des tiers. Dans de telles conditions, le rapport assurera la communication entre les deux parties. SDM présente le contenu comme suit :

- un résumé des principaux résultats de cette phase
- les problèmes spécifiques rencontrés
- les changements apportés au système par rapport aux résultats de l'analyse fonctionnelle.
- une révision éventuelle de l'analyse coût & bénéfice suite aux changements apportés.
- toutes les conventions et remarques concernant la suite du développement.

EVALUATION

Nous pensons également que ce document est **superflu** si tous les membres de l'équipe de projet travaillent dans le même lieu car ils peuvent consulter librement tous les documents réalisés au sein de l'équipe et, en ce qui concerne les directives ou remarques dont ils doivent tenir compte pour programmer, les discussions informelles entre les personnes est certainement le moyen de communication le plus efficace. De plus, ce rapport n'intéresse pratiquement plus l'utilisateur excepté la partie concernant la définition des procédures manuelles et des interfaces homme/machine. Ce rapport sera utile en fonction des critères n° 1 (activités futures) et n° 3 (communication) si les rapports informels entre le chef de projet, les analystes et les programmeurs sont difficiles en raison de la distance physique qui les sépare. Le rapport constituera dans ce cas le cahier des charges du programmeur.

DESTINATAIRES

Les programmeurs de l'équipe de projet.

TABLEAU RECAPITULATIF

Phase 3 : Analyse technique

<u>Activités</u>	<u>Documents</u>	<u>Utilité¹</u>	<u>Critères²</u>				
			1	2	3	4	5
3.1	Manuels d'utilisation	U					**
3.2	Input et output de l'ordinateur	I	***		***		
3.3	Schéma physique des données	I	***			***	
3.4	Mesures de sécurité et de confidentialité (synthèse)	U					**
3.5 & 3.6	Spécifications des programmes	I	**			***	
3.8	Plan du programme	I		***			
3.9	Rapport de l'analyse technique	S/U	*		**		

¹ Utilité : S = Superflu -U = Utile I = Indispensable

² Critères :

Critère n°1 : Critère de bon déroulement des activités futures

Critère n°2 : Critère de contrôle et des gestion efficace du projet

Critère n°3 : Critère de communication et de prise de décision

Critère n°4 : Critère de maintenance efficace du système

Critère n°5 : Critère de transfert et de bonne connaissance du système

2.2.4 PROGRAMMATION ET TESTS ISOLEES

Il s'agit de la réalisation proprement dite du système qui implique l'acquisition du matériel, le codage des applications puis l'intégration du logiciel sur l'infrastructure matérielle. L'objectif est d'aboutir à un ensemble en état de fonctionnement et répondant aux souhaits des demandeurs.

4.1 Descriptions des tâches

PROCEDE

Une tâche est une ou plusieurs procédures ou une partie de procédure manuelle effectuée(s) par une seule personne. La description des tâches est donc le prolongement de l'activité 3.1, la conception des procédures manuelles. Ce document peut servir à la formation des utilisateurs, à déterminer comment et par qui les différentes tâches seront réalisées ou pour déterminer le personnel nécessaire.

DOCUMENTATION

PROPOSITION SDM

SDM propose de réaliser un document intitulé "**description de tâches**" qui définit chaque tâche en fonction des procédures qu'elle regroupe et qui détermine le profil de la personne qui exécutera la tâche. On déterminera également avec quel degré de précision chaque tâche devra être accomplie et le taux d'erreur admis.

EVALUATION

Nous estimons que la réalisation d'un document décrivant les tâches manuelles est **superflue** dans la plupart des cas car les descriptions des procédures manuelles et les manuels d'utilisation sont suffisants pour organiser le travail des utilisateurs. Ce document sera utile ou même indispensable dans le cas d'un système à risque où aucune erreur n'est permise, mais nous pensons que ce travail n'est pas du ressort de l'équipe de projet (qui donne la priorité aux étapes de codage) mais plutôt des responsables des utilisateurs et des personnes chargées de la formation. Toutefois, le département informatique de la CGER ne développant pas de projets à haut risque, ce document y est superflu.

4.2 Définition des exigences auxquelles le personnel et l'environnement de travail doivent satisfaire

PROCEDE

Après avoir réalisé la description des tâches, il s'agit de définir les exigences concernant le personnel et l'environnement de travail, c'est-à-dire le volume et le niveau du personnel à rassembler, les engagements éventuels à prévoir, les formations à dispenser, les adaptations éventuelles de l'environnement de travail (espace, prévention des accidents, lumière, ...). Cette activité n'entraîne pas de réalisation de documentation.

4.3 Détail des descriptions des programmes

PROCEDE

Le programmeur disposant des spécifications des programmes réalisées durant l'analyse technique, va les détailler suffisamment pour pouvoir réaliser le codage.

DOCUMENTATION

PROPOSITION SDM

SDM estime qu'il est nécessaire de réaliser des spécifications plus détaillées lorsqu'il s'agit de programmes très complexes. Il s'agit de diviser le programme en un certain nombre de modules plus simples et de bien documenter les liens entre ces modules à l'aide d'un schéma présentant la structure du programme. Pour la documentation des modules on n'utilisera plus de schémas mais on se servira de commentaires placés directement dans le programme source et qui présenteront le but de chaque module.

EVALUATION

Les programmeurs chargés de cette documentation ont souvent tendance à la négliger afin de passer au plus vite à l'étape de codage proprement dite. Si l'absence de spécifications détaillées n'a pas d'effet direct sur le bon déroulement du codage, elle est souvent la cause de gros problèmes lors de la mise en production des programmes ainsi que durant la maintenance. En effet, le schéma de la découpe du programme en modules est indispensable pour l'équipe de production car pour exécuter un programme, elle a besoin de connaître l'ordre d'exécution des différents modules qui le composent ainsi que les informations qu'ils échantent. Cette documentation est donc rendue **indispensable** par le critère n°5 concernant le transfert du système en production et peut être facilement standardisée à l'aide de modèles de représentation tels que

SHAT (vu page 17). De plus, une documentation minimale des différents modules d'un programme est indispensable selon le critère n°4 (maintenance), pour que l'équipe chargée de la maintenance du programme puisse lire et comprendre sans trop d'ambiguïté la logique du programme à modifier. Toutefois, nous pensons que SDM est trop exigeante lorsqu'elle conseille de documenter tous les programmes en utilisant le pseudo-code car vue sous un autre angle, une documentation excessive est fort pénalisante lors des étapes de maintenance des programmes car la mise à jour de toute cette documentation devient une tâche très lourde et risque d'être négligée. Après quelques temps, on ne dispose plus que d'une documentation désuète et incohérente. Il s'agit donc de trouver le juste milieu.

DESTINATAIRES

Les programmeurs, l'équipe de production et l'équipe de maintenance.

4.4 Codification des programmes

PROCEDE

Il s'agit de traduire les spécifications des programmes en instructions écrites à l'aide d'un langage compréhensible par la machine. Celles-ci constitueront les programmes-sources qui devront être aussi simples et clairs que possible.

DOCUMENTATION

PROPOSITION SDM

Comme nous l'avons dit au point précédent, SDM insiste sur l'utilité d'insérer des commentaires dans les programmes-sources afin d'en augmenter la lisibilité.

4.5 Compilation et correction des programmes

PROCEDE

Cette activité consiste à compiler les programmes-sources afin de détecter puis de corriger les erreurs de codage pour obtenir finalement la version finale des programmes-objets. Aucune création de document n'est nécessaire si ce n'est la mise à jour de la liste des programmes disponibles dans la bibliothèque de l'organisation.

4.6 Constitution des données de test

PROCEDE

Il s'agit de construire les cas de test ainsi que les données de test qui permettront au programmeur de vérifier le bon fonctionnement de son programme. Ces informations pourront être réutilisées lors de la phase suivante pour réaliser les tests d'intégration et d'acceptation du système.

DOCUMENTATION

PROPOSITION SDM

SDM fournit quelques mesures à respecter pour réaliser le document intitulé "**cas et données de test des programmes**" : les cas de test doivent être conçus de manière à envisager toutes les situations possibles, à parcourir tous les pas possibles du programme tout en testant sa logique et en utilisant toutes les sortes de données possibles. En ce qui concerne les données de test, SDM propose différentes manières de les obtenir : en imaginant des cas réels, en faisant des extractions des fichiers existants ou en utilisant un générateur de données de test.

EVALUATION

Une bonne définition des cas et des données de test est **indispensable** selon le critère n° 1 (activités futures) au bon déroulement et à l'efficacité du test du programme. De la qualité de ce document va directement dépendre la qualité du nouveau système qui sera soumis aux tests d'acceptation et par conséquent, la satisfaction de l'utilisateur.

DESTINATAIRES

Le programmeur chargé de réaliser le test du programme et les personnes qui réaliseront les tests d'intégration et d'acceptation.

4.7 Tests des programmes

4.8 Compléter la documentation des programmes

PROCEDE

Durant cette étape, le programmeur teste le bon fonctionnement de son programme qui inclut le test d'intégration des modules à l'intérieur du programme, le test des importations et exportations des données ainsi que le test de la logique de fonctionnement à l'intérieur du programme. Puis, il procède à la correction des erreurs détectées. Une fois la version finale du programme atteinte, il s'agit de vérifier et de compléter éventuellement la documentation du programme en fonction des modifications qui auront été

réalisées suite aux tests.

DOCUMENTATION

PROPOSITION SDM

SDM exige la réalisation d'un dossier pour chaque programme qui contiendra les spécifications du programme et les schémas du programme qui auront été vérifiés et éventuellement mis à jour, le programme source ainsi que les informations de test (données et cas de test, résultats des tests). SDM demande également de vérifier si aucune mise à jour de la documentation de l'analyse technique n'est nécessaire suite aux modifications réalisées lors du test du programme.

EVALUATION

Nous estimons qu'il est **indispensable** de tester la documentation des programmes au niveau de la cohérence, de la concision et de la lisibilité et d'effectuer les mises à jour nécessaires en raison des critères n°5 (transfert) et n°4 (maintenance) car la mise en production du système ainsi que sa maintenance ne peuvent être efficaces sans une documentation correcte des programmes. La proposition de rassembler toute la documentation relative à un programme en un seul dossier nous paraît utile pour faciliter la maintenance future du système. Toutefois, comme ce dossier ne contiendra rien de nouveau, il s'avèrera superflu si l'organisation dispose d'un système de documentation centralisé et accessible facilement. Enfin, nous pensons que les programmes sont encore susceptibles d'être modifiés suite aux tests d'intégration et d'acceptation, c'est pourquoi nous placerions la réalisation de ce dossier à la fin de la phase suivante, lorsque tous les tests sont terminés.

DESTINATAIRES

Le dossier du programme sera utilisé par l'équipe de maintenance.

TABLEAU RECAPITULATIF

Phase 4 : Programmation et tests isolés

<u>Activités</u>	<u>Documents</u>	<u>Utilité¹</u>	<u>Critères²</u>				
			1	2	3	4	5
4.1	Description des tâches manuelles	S					
4.3	Description détaillée des programmes	I				***	***
4.6	Cas et données de test des programmes	I	***				
4.7 & 4.8	Dossiers de documentation des programmes	U				**	

¹ Utilité : S = Superflu U = Utile I = Indispensable

² Critères :

Critère n°1 : Critère de bon déroulement des activités futures

Critère n°2 : Critère de contrôle et de gestion efficace du projet

Critère n°3 : Critère de communication et de prise de décision

Critère n°4 : Critère de maintenance efficace du système

Critère n°5 : Critère de transfert et de bonne connaissance du système

2.2.5 REALISATION DES TESTS

Durant cette phase, toutes les parties qui constituent le système vont être testées ensembles. On distingue trois types de tests :

- les tests des procédures manuelles et des facilités.
- les tests du système d'information (fonctionnement global du système).
- les tests d'acceptation.

5.1 Elaboration d'un plan de test détaillé

PROCEDE

Un plan global de cette phase a été réalisé en fin d'analyse technique. Ce plan va maintenant être détaillé au niveau de la répartition des tâches et modifié en fonction des résultats des tests des programmes, des éventuels retards durant la phase de programmation ou concernant la livraison d'appareils ou de logiciels.

DOCUMENTATION

PROPOSITION SDM

SDM propose de réaliser un **plan de test** qui détaillera tous les composants qui devront être testés, les appareils et fichiers nécessaires pour réaliser les tests, les plannings des différents tests ainsi que la manière dont les résultats devront être présentés.

EVALUATION

SDM reste très générale quant à la manière de réaliser ce plan. C'est regrettable vu l'importance d'un tel document. En effet, un plan de test est **indispensable** selon le critère n°2 (contrôle et gestion) pour organiser et contrôler au mieux cette étape dont va dépendre la qualité du système et la satisfaction des futurs utilisateurs. La réalisation de ce plan est très complexe car il s'agit de coordonner le travail de nombreuses personnes (équipe de développement, équipe de production et utilisateurs) et de planifier des activités dont la durée est très difficile à estimer en raison de l'imprévisibilité des résultats.

DESTINATAIRES

Le chef de projet et les testeurs.

5.2 Installation du matériel, du software et préparation de l'environnement de travail

PROCEDE

Cette étape consiste à contrôler l'installation du nouveau matériel et des logiciels achetés à l'extérieur ainsi que la préparation de l'environnement de travail afin que tout soit prêt pour la date prévue de mise en production du système. Cette activité ne nécessite pas la réalisation de documentation particulière.

5.3 Définition des unités de traitement (jobs) et de leurs conditions d'exécution (job control)

PROCEDE

Cette activité consiste à regrouper les programmes en unités de traitement exécutables par la machine et de rassembler toutes les informations nécessaires à leur exécution pour constituer un 'job control'.

DOCUMENTATION

PROPOSITION SDM

SDM propose de réaliser pour chaque job un document appelé "**job control**" qui définira les différents programmes à utiliser, les points de contrôle durant l'exécution du job, le temps d'exécution maximum, les priorités du job ainsi que les mesures de sécurité à prendre.

EVALUATION

Vu le caractère très technique de ce document, il est normal que SDM reste vague au niveau de sa forme. Ce document doit **obligatoirement** être réalisé selon le critère n°5 (transfert) car sans lui, il n'est pas possible de transférer le système en production et de l'exécuter. Il s'agit d'un document très spécifique et technique qui fait généralement l'objet d'une standardisation de la part de l'équipe de production.

DESTINATAIRES

L'équipe de production.

5.4 Test des formations, des moyens d'aide pour l'utilisation du système ainsi que des procédures manuelles.

PROCEDE

SDM ne présente aucun procédé à suivre pour réaliser les tests des formations, des moyens d'aide et des procédures manuelles et nous pensons qu'il n'en existe aucun si ce n'est en les mettant en pratique et en observant les réactions des personnes.

Exemple : enthousiasme des personnes durant la formation, taux d'utilisation des manuels, contrôle du respect des procédures manuelles. Cette activité ne nécessite pas de création de documentation.

5.5 Rassembler les données pour le test d'intégration du système et le test d'acceptation

PROCEDE

Il s'agit de réaliser un document intitulé "**spécifications des tests**" qui présentera le contenu de chaque test ainsi que les données qui seront utilisées.

DOCUMENTATION

PROPOSITION SDM

SDM insiste fort sur la qualité de ce document car de lui va dépendre la qualité du système développé. C'est pourquoi elle suggère d'écrire des "**spécifications des tests**" qui devront contenir :

- le but de chaque test ainsi que les données dont il aura besoin.
- ce qui devra être fait et quand
- ce qui doit être recherché et ce qui doit être constaté".

EVALUATION

Comme on peut le remarquer, SDM est très vague quant au contenu des spécifications de test et cela malgré le fait qu'elle attache une grande importance à ce type de documentation. Nous pensons qu'il s'agit d'une documentation qui est souvent négligée par les développeurs en raison du fait qu'ils ont déjà réalisé un tel plan pour les tests individuels des programmes et qu'ils ne se sentent plus directement concernés par la réalisation de tests d'intégration et d'acceptation, les premiers étant du ressort de l'équipe de production, les seconds étant effectués par des utilisateurs. Toutefois, nous estimons qu'il est **indispensable** selon le critère n° 1 (activités futures) de définir précisément tous les cas de test possibles ainsi que les fichiers de données à utiliser, aussi bien pour le test d'intégration que le test d'acceptation de manière à minimiser la possibilité qu'une erreur ne soit pas détectée lors des tests. Il est clair que la correction d'une erreur alors que le système est en production coûte très cher et pose de gros problèmes (on consultera à ce sujet B.W. BOEHM (12).

En ce qui concerne le test d'acceptation, nous pensons qu'il est préférable que la documentation soit réalisée par les utilisateurs eux-mêmes car il est difficile pour les développeurs (qui connaissent parfaitement le fonctionnement du système) de mettre en évidence les imperfections de leur propre travail.

DESTINATAIRES

La partie liée aux tests d'intégration concerne les programmeurs et l'équipe de production et la partie concernant les tests d'acceptation est destinée aux utilisateurs.

5.6 Test d'intégration du système

PROCEDE

Le test d'intégration du système inclut divers tests tels les liens entre les programmes, la séquence d'exécution des programmes, les job-control, l'utilisation des fichiers et bases de données corrects, les mesures de sécurité (réexécution, règles de reconstruction), les dialogues entre l'utilisateur et le système, les critères de prestation fixés, la documentation destinée à l'équipe de production et de maintenance.

DOCUMENTATION

PROPOSITION SDM

Au terme du test, SDM suggère de réaliser un document intitulé **"résultats du test d'intégration"** qui contiendra la liste des erreurs et problèmes importants rencontrés ainsi que des mesures prises pour y remédier.

EVALUATION

Ce document est **indispensable** selon le critère n°1 (activités futures) car la correction des erreurs rencontrées peut entraîner des modifications au niveau des résultats des phases d'analyse technique et de programmation qui devront partiellement être réexécutées. Il est important de déterminer tous les documents existants qui devront être mis à jour suite aux corrections qui vont être apportées au système, et cela afin d'assurer la cohérence de la documentation pour toutes les étapes du développement.

DESTINATAIRES

L'équipe de développement.

5.7 Exécution du test d'acceptation

PROCEDE

Le test d'acceptation permet d'obtenir l'approbation du système par les clients et entraîne le transfert des responsabilités du système, de l'équipe de développement au client ou à l'équipe de production. Généralement réalisé sous la responsabilité du client, ce test permet de découvrir les erreurs de logique non détectées lors du test du système, les différences entre ce que fait réellement le système et ce que le client attendait ainsi que les critères de prestation (définis aux étapes 1.4, 1.9 et 2.1) auxquels le système ne satisfait pas.

DOCUMENTATION

PROPOSITION SDM

SDM exige de réaliser un rapport de test intitulé "**rapport du test d'acceptation**" qui contiendra une description détaillée de tous les problèmes rencontrés au test et des solutions prises pour y remédier. Ce rapport sera soumis à l'approbation du client. En cas d'acceptation, l'équipe de projet délèguera ses responsabilités au client.

EVALUATION

Ce rapport est **indispensable** tout d'abord en raison du critère n°3 (communication et décision) car il permettra au client de connaître et d'évaluer les améliorations qui vont être apportées au système suite aux critiques qu'il aura émises lors du test. Ensuite, comme pour le rapport du test du système, il permettra, selon le critère n°1 (activités futures) de réaliser les mises à jour nécessaires au niveau des résultats des phases d'analyse technique et de programmation. Si le client estime que certains objectifs ne sont pas remplis par le système, les conséquences peuvent être très lourdes car les résultats de l'analyse fonctionnelle sont remis en cause. Cependant, nous pensons que ce genre de problème peut être évité par une collaboration entre utilisateurs et développeurs tout au long du développement.

DESTINATAIRES

L'équipe de développement, les testeurs et les clients.

5.8 Rédaction du rapport "Résultats de test"

PROCEDE

Cette activité consiste à réaliser un rapport **récapitulatif** des résultats de tous les tests effectués, des problèmes découverts et de leurs solutions.

DOCUMENTATION

PROPOSITION SDM

Ce rapport concerne les tests des programmes qui ont été réalisés durant la phase de programmation ainsi que les tests d'intégration et d'acceptation du système réalisé durant cette phase. "Il devra contenir :

- Une vue plus technique des résultats des tests telle le plan de test et la manière dont il a été réalisé, le nombre d'erreurs constatées, ...
- Une description et une analyse des problèmes signalés.
- Une présentation des améliorations apportées pour éviter de tels

- problèmes à l'avenir.
- Une révision de l'analyse coût & bénéfice".

EVALUATION

Nous pensons que ce rapport est **superflu** car il est redondant avec les documents "résultats du test d'intégration" et "rapport du test d'acceptation" réalisés à la fin des deux tests respectifs. Quant aux résultats des tests isolés des différents programmes, nous ne pensons pas qu'ils puissent intéresser le client dont le soucis actuel est que le système soit rapidement mis en production.

TABLEAU RECAPITULATIF

Phase 5 : Tests du système

<u>Activités</u>	<u>Documents</u>	<u>Utilité¹</u>	<u>Critères²</u>				
			1	2	3	4	5
5.1	Plan de tests	I		***			
5.3	Job control (pour chaque job)	I					***
5.5	Spécification des tests	I	**				
5.6	Résultats du test d'intégration	I	***				
5.7	Rapport du test d'acceptation	I	**		***		
5.8	Récapitulatif des résultats des tests	S					

¹ Utilité : S = Superflu U = Utile I = Indispensable

² Critères :

Critère n°1 : Critère de bon déroulement des activités futures

Critère n°2 : Critère de contrôle et des gestion efficace du projet

Critère n°3 : Critère de communication et de prise de décision

Critère n°4 : Critère de maintenance efficace du système

Critère n°5 : Critère de transfert et de bonne connaissance du système

2.2.6 CONVERSION ET MISE EN PRODUCTION

Cette phase a pour objectif principal de préparer le transfert en production du système développé ainsi que de préparer l'environnement dans lequel il devra fonctionner. Ceci inclut notamment la formation et la motivation des personnes qui utiliseront le système, qui en assureront le fonctionnement ou la maintenance, la mise en place des appareils nécessaires, la conversion des fichiers et bases de données existantes ainsi que la mise en place d'une nouvelle méthode de travail.

6.1 Réalisation d'un plan détaillé de l'étape de conversion et de mise en production

PROCEDE

Il s'agit de détailler le plan global de conversion et de mise en production inclut dans le plan total réalisé à la fin de l'analyse conceptuelle (activité 2.13).

DOCUMENTATION

PROPOSITION SDM

SDM propose de réaliser un **plan de la phase** qui fixera les responsabilités pour les étapes de formation, de conversion et de mise en production du système et qui contiendra une liste de tous les problèmes susceptibles de se poser lors de ces étapes et des moyens d'aide pour les résoudre.

EVALUATION

Il est **indispensable** selon le critère n°2 (contrôle et gestion) de réaliser un plan de la phase qui permettra de combiner les plannings de formation, de conversion et de mise en production de façon à établir le chemin critique jusqu'à la date prévue de mise en production du système. Ce plan devra également prévoir des lignes de communication directes entre le chef du projet, les formateurs, les personnes effectuant les conversions et l'équipe de production, afin de faciliter les prises de décision lorsqu'un problème se pose.

6.2 Formation du personnel informatique

PROCEDE

Si la mise en production du nouveau système requiert la mise en place de moyens hardware nouveaux, il est nécessaire de former le personnel informatique pour qu'il puisse en assurer le fonctionnement et qu'il puisse réagir en cas de situation anormale. Aucune réalisation de documentation n'est nécessaire.

6.3 Elaboration d'instructions de conversion et de mise en production

PROCEDE

Cette activité consiste à préparer en détail toutes les opérations à réaliser et à déterminer leur ordre de succession afin que l'étape de conversion des données et de préparation du système pour la mise en production ne dépasse pas la durée estimée dans le planning. Cette activité se base sur le plan réalisé en début de phase et n'entraîne pas de documentation particulière.

6.4 Conversion des données

PROCEDE

Cette étape consiste à transférer les données existantes vers la nouvelle structure de stockage développée.

DOCUMENTATION

PROPOSITION SDM

SDM propose de réaliser un "**rapport final de conversion des données**" qui contiendra la liste des problèmes rencontrés et des mesures prises pour y remédier ainsi que les modifications éventuelles du planning que ces problèmes ont entraînées.

EVALUATION

Nous savons que les opérations de conversion des données peuvent être complexes et très longues, c'est pourquoi elles font souvent l'objet d'un projet particulier. Dans le cas de conversions relativement limitées, nous pensons qu'il est **superflu** de réaliser un rapport final car les problèmes rencontrés sont peu nombreux et les leçons qui en seront tirées ne serviront à rien puisqu'il s'agit d'une activité qui ne sera réalisée qu'une seule fois en principe.

6.5 Donner des renseignements sur le nouveau système

PROCEDE

Cette activité a pour but de s'assurer que les futurs utilisateurs se sentent concernés par le nouveau système et qu'ils ne présenteront pas de résistance au changement. Pour cela, il faut veiller à ce qu'une ligne de communication directe soit possible entre les développeurs et les utilisateurs afin que ces derniers puissent poser des questions ou obtenir des renseignements supplémentaires, que ce soit oralement ou par écrit. Cette activité n'inclut pas la réalisation de documentation particulière.

6.6 Formation des utilisateurs

PROCEDE

Cette activité consiste à présenter à chaque utilisateur sa tâche spécifique par rapport au système, ses responsabilités ainsi que la manière d'exécuter sa tâche. Cette formation se fera via des cours, des démonstrations ou encore des manuels d'utilisation. Aucune documentation particulière n'est réalisée à cette étape; elle a été réalisée avant, (manuels d'utilisation, descriptions des procédures manuelles).

6.7 Formation du personnel de maintenance

PROCEDE

Il s'agit de mettre à la disposition de l'équipe de maintenance toute la documentation existante lui permettant d'acquérir les connaissances nécessaires concernant les moyens hardware utilisés, les programmes, les fichiers et bases de données utilisées ainsi que les lignes de communication établies entre elles, les utilisateurs et l'équipe de production. Aucune documentation particulière n'est réalisée à cette étape.

6.8 Mise en production du système et transfert des responsabilités

PROCEDE

Cette étape consiste essentiellement à vérifier que tous les problèmes rencontrés durant les phases de test et de conversion ont été résolus et que le système est maintenant prêt à fonctionner. Après ce contrôle, il ne reste plus qu'à lancer le système, ce qui marquera la fin du projet. C'est également durant cette étape que la documentation du système est contrôlée puis clôturée. Quant à la documentation du projet, elle n'a plus de raison d'être et rejoint la documentation historique.

DOCUMENTATION

PROPOSITION SDM

SDM suggère tout d'abord de réaliser un rapport de cette phase 6 mais ne donne aucune indication quant à son contenu. Deuxièmement, elle propose la réalisation d'un "**rapport global du projet**" quelque temps après le lancement du système dans lequel on comparera le système réalisé aux attentes de départ, on rappellera les problèmes majeurs rencontrés durant ce développement et desquels il sera possible de tirer des leçons.

EVALUATION

Nous ne voyons pas l'utilité de réaliser un rapport final pour cette phase de conversion et de mise en production. En ce qui concerne la réalisation d'un rapport d'évaluation générale du projet, elle permettra à l'équipe de développement de jeter un oeil critique sur la manière dont elle a mené ce projet et peut-être tirer des leçons pour les projets futurs. Toutefois, il s'agit d'un rapport dangereux, politiquement car il permettrait de rendre l'équipe de développement responsable de ces erreurs.

TABLEAU RECAPITULATIF

Phase 6 : Conversion et mise en production

<u>Activités</u>	<u>Documents</u>	<u>Utilité¹</u>	<u>Critères²</u>				
			1	2	3	4	5
6.1	Plan détaillé de conversion et mise en production	I		***			
6.4	Rapport final de conversion des données	S					
6.8	(Rapport global du projet Rapport final de la phase	S					

¹ Utilité : S = Superflu U = Utile I = Indispensable

² Critères :

Critère n°1 : Critère de bon déroulement des activités futures

Critère n°2 : Critère de contrôle et des gestion efficace du projet

Critère n°3 : Critère de communication et de prise de décision

Critère n°4 : Critère de maintenance efficace du système

Critère n°5 : Critère de transfert et de bonne connaissance du système

2.3 Tableau récapitulatif de la documentation minimale d'un projet

Suite à cette critique à priori des documents proposés par SDM, nous vous proposons un tableau récapitulatif qui, pour chaque phase de SDM, présente la documentation que nous estimons être indispensable pour tout projet de développement de système d'information. Ainsi présenté, ce tableau offre au développeur une 'checklist' de la documentation à réaliser tout au long du développement.

PHASE 1 : Analyse conceptuelle Définition du problème Objectifs et exigences du nouveau système Schéma global du nouveau système Solutions alternatives et sélection Planning global de la suite du développement Rapport d'analyse conceptuelle
PHASE 2 : Analyse fonctionnelle Schéma de la découpe en sous-systèmes et descriptions Schéma logique des données Cahier des charges complet Planning détaillé de l'analyse technique Rapport final de l'analyse fonctionnelle
PHASE 3 : Analyse technique Input et output de l'ordinateur Schéma physique des données Spécifications des programmes Plan de programmation et de test
PHASE 4 : Programmation et tests isolés Description détaillée des programmes Cas et données de test des programmes
PHASE 5 : Tests Plan de tests Job control (pour chaque job) Spécification des tests Résultats du test d'intégration Rapport du test d'acceptation
PHASE 6 : Conversion et mise en production Plan détaillé de conversion et de mise en production

En conclusion de ce deuxième chapitre, nous retiendrons essentiellement les différents composants de la documentation que nous avons dégagés de la méthode SDM ainsi que l'utilité que nous accordons à leur réalisation.

En plaçant les six tableaux récapitulatifs de la documentation de chacune des phases de développement de SDM, on dispose alors d'une check-list qui permet aux développeurs de vérifier leur documentation. Ils pourront contrôler si la documentation minimale a bien été réalisée.

Dans le troisième chapitre est proposée une seconde critique de la documentation de SDM, basée non plus sur des critères d'évaluation bien définis mais sur l'expérimentation des développeurs de la CGER.

CHAPITRE 3

**Résultats de l'enquête, des
interviews et interprétation**

Pour réaliser ce chapitre, nous nous basons sur les interviews et enquêtes qui ont été menés dans les trois DSI du département informatique.

3.1 PRÉSENTATION DE L'ENQUÊTE

L'enquête s'est déroulée en deux phases : tout d'abord, des interviews ont été menés au sein des 3 DSI du département informatique de la CGER à Bruxelles. Les personnes interviewées étaient :

CHEF DE PROJET	9
INFORMATICIEN DES GROUPES DE SUPPORT AU DEVELOPPEMENT	5
REPRESENTANT DES UTILISATEURS	1
MEMBRE EQUIPE DE PREPARATION-PRODUCTION	1
GROUPE DBA ⁽¹⁾	2

Les chefs de projet interviewés appartiennent aux groupes de développements suivants : comptes universels, logistique, personnel et pensions.

Les informations qu'on cherchait à connaître ont trait à :

- l'organisation et le travail réalisé par une équipe de projet.
- l'importance que les développeurs attachent à la méthode SDM.
- la manière dont les projets sont documentés.
- l'avis des développeurs concernant les méthodes, modèles et outils de développement dont ils disposent.
- les relations entre développeurs, utilisateurs et groupes de support 'méthodes' et 'outils'.

La deuxième partie de l'enquête s'est faite au moyen d'un questionnaire que nous avons fait parvenir à tous les chefs de projet du département informatique :

nombre d'enquêtes envoyées : 52

nombre de réponses reçues : 31

Notons que sur les trente et une réponses, une personne a renvoyé le questionnaire vierge en déclarant ne pas appliquer la méthode SDM pour développer ses projets.

⁽¹⁾ Administration et conception des bases de données.

L'objectif de ce questionnaire était triple :

- déterminer la documentation réalisée par les développeurs lors d'un projet de développement de logiciel ainsi que les divers outils qu'ils emploient à cet effet.
- mettre en évidence les types de documentation que les développeurs estiment automatisables, c'est-à-dire pour lesquels ils souhaiteraient un outil qui les assisterait dans la réalisation de ce projet.
- savoir si ces possibilités d'automatisation de documentation sont souhaitées par les développeurs.

Au début du questionnaire se trouvent des questions plus générales permettant d'évaluer le degré de risque moyen des projets développés grâce au tableau de Mac Farlan vu précédemment. Un exemplaire de ce questionnaire se trouve en annexe 3. Les résultats de cette enquête se structurent en trois points :

- Questions générales
- Questions portant sur la réalisation de la documentation
- Questions portant sur l'automatisation de la documentation

3.2 QUESTIONS GÉNÉRALES

Détermination du degré de risque des projets selon la méthode de Mac Farlan :

	MIN	MAX	MOY
Taille du projet en mois/homme	7	200	60
Incertitude quant aux spécifications (5 : spécifications précises 0 : pas ou peu de spécifications)	2	5	3.5
Incertitude technique (5 : environnement technique connu 0 : nouvel environnement technique)	1	5	4

De ce tableau, nous pouvons déduire qu'il s'agit de projets de taille élevée, dont le niveau de la technologie utilisée est dans la plupart des cas très faible par rapport à l'expérience des développeurs et dont le niveau de structuration est moyen.

Nous pouvons donc conclure selon les résultats de Mac Farlan que les projets concernés par cette enquête comportent un degré de risque allant de faible à faible-moderé.

3.3 LA REALISATION DE LA DOCUMENTATION

Pour connaître le contenu de la documentation réalisée par les développeurs, nous avons proposé dans le questionnaire un tableau récapitulatif qui, pour chaque activité des phases de développement de SDM (phases 1 à 6), présente le type de documentation que SDM propose de réaliser. Pour chaque type de documentation, trois réponses étaient possibles : non réalisé, réalisé sommairement ou réalisé de façon complète. Ces informations vont nous permettre de classer les différents types de documentation sur base de leur fréquence de réalisation. Ainsi, il sera possible de classer les types de documentation en trois catégories :

- **la documentation jugée indispensable** : réalisée par au moins 80% des développeurs
- **la documentation jugée utile** : entre 50% et 80% des développeurs la réalise.
- **la documentation jugée superflue** : réalisée par moins de 50% des développeurs.

Un deuxième pourcentage nous permettra de distinguer parmi les développeurs qui réalisent le type de documentation cité, ceux qui le réalisent de manière **complète**. Nous tenterons d'expliquer chacun des résultats en nous basant sur les informations recueillies lors des interviews.

3.3.1 La documentation jugée indispensable

a. SYNTHÈSE

PHASE 1 : ANALYSE CONCEPTUELLE	(1)	(2)
1. Le projet cadre	100 %	42 %
2. Planning global de la suite au développement	92 %	55 %
3. Schéma global du nouveau système	83 %	70 %
PHASE 2 : ANALYSE FONCTIONNELLE		
4. Schéma logique des données commenté	92 %	100 %
5. Planning détaillé de l'analyse technique	92 %	70 %
PHASE 3 : ANALYSE TECHNIQUE		
6. Mesures de sécurité et de confidentialité	100 %	100 %
7. Schéma physique des données	92 %	100 %
8. Spécifications des programmes	83 %	100 %
PHASE 4 : PROGRAMMATION ET TESTS ISOLÉS		
9. Description détaillée des programmes	92 %	100 %
10. Cas et données de test des programmes	83 %	80 %
11. Dossier de documentation des programmes	83 %	80 %
PHASE 5 : TESTS DU SYSTÈME		
12. Le Job Control (pour chaque job)	100 %	90 %
13. Résultats du test d'intégration	100 %	80 %
14. Plan de tests (phases 5 & 6)	91 %	50 %
15. Spécification des tests (programmes et données)	90 %	78 %
16. Rapport du test d'acceptation	100 %	78 %
PHASE 6 : CONVERSION ET MISE EN PRODUCTION		
17. Planning détaillé de conversion et mise en production	90 %	77 %
18. Rapport final de la phase	90 %	89 %

(1) pourcentage de développeurs qui réalisent ce type de documentation.

(2) pourcentage de développeurs qui réalisent ce type de documentation de manière complète.

b. INTERPRETATION DES RESULTATS

Nous allons reprendre ci-dessous chaque composant du tableau et commenter les résultats obtenus.

1. Projet-cadre

Outils utilisés : traitement de texte : 50 %
aucun : 42 %
Datamanager : 17 %

Les demandes de développement sont gérées par un comité spécial du secteur d'activité appelé le "steering committee" qui décide de l'insertion de nouveaux projets à l'inventaire des projets à développer. Il détermine également les priorités de ces projets et exerce un suivi des réalisations. Pour qu'une demande de développement de projet puisse être acceptée, elle doit notamment contenir une description précise de ce qui est demandé ainsi que les raisons impératives éventuelles de la demande. Les développeurs sont donc **obligés** de réaliser un projet-cadre pour que la demande de projet ait des chances d'être acceptée. En outre, le degré de structuration du projet constitue un des facteurs qui influencent l'attribution des priorités aux projets de développement (cfr méthode de Benson page 39). Or, nous avons vu qu'en général, les spécifications des projets manquent de précision et cela se reflète également ici par le fait que plus de la moitié des développeurs interrogés affirment ne réaliser que sommairement le projet-cadre. Ils expliquent cette lacune par la difficulté de communiquer avec les utilisateurs, principaux demandeurs de projets :

- L'utilisateur a tendance à penser trop 'technique' et se montre réticent à faire la démarche d'étude conceptuelle de son problème. Il ne sait pas toujours ce qu'il veut ou ne sait pas l'exprimer de façon claire, complète et non ambiguë. De plus, les délais qu'il exige poussent les développeurs à faire des choix techniques bien avant d'avoir terminé l'étude conceptuelle et fonctionnelle.
- La méthode SDM ne précise pas avec suffisamment de précision les moments où les développeurs et les utilisateurs doivent collaborer et c'est le développeur qui les détermine, ce qui accentue le sentiment de dépendance de l'utilisateur. Ce dernier ne se montre alors guère motivé et laisse faire le développeur.
- Le contenu du projet-cadre n'est pas formalisé ni standardisé et certains développeurs avouent qu'ils se limitent souvent aux comptes-rendus des réunions avec la direction et les utilisateurs.
- Le CTI de la CGER est un client important et le nombre de projets à développer est supérieur aux capacités de développement existantes (développeurs, budget). De ce fait, les dirigeants imposent des délais très courts et privilégient le respect de ceux-ci au suivi de la méthodologie et particulièrement à la réalisation de la documentation. Or, si l'on veut

convaincre les développeurs de l'importance d'une bonne documentation, il faut que les dirigeants soient eux-mêmes convaincus et qu'ils se montrent moins tolérants face à des projets mal documentés quoique développés dans le temps imparti.

- Le fait que les clients, les utilisateurs et les développeurs soient sous le même toit n'encourage pas l'utilisateur à fixer de façon claire et détaillée ses besoins dès le début; il lui est facile de contacter le développeur à tout moment pour lui fournir des précisions et exigences supplémentaires.

2. Le planning global de la suite du développement

Outils utilisés : traitement de texte: 27 %

Superproject : 9 %

aucun : 55 %

Un projet ne peut normalement être accepté que si la demande en a été faite au Steering Committee à l'aide d'un document standardisé qui doit notamment inclure un planning global du développement et une analyse coûts-bénéfices du projet. Ces deux documents permettront d'évaluer la charge de travail du projet qui servira lorsqu'on fixera la liste annuelle des projets à développer en fonction des ressources disponibles. Les quelques développeurs qui ne réalisent pas de planning ont la charge de projets prioritaires (qui doivent être développés quel qu'en soit le prix) et qui sont donc certains d'être acceptés.

Nous remarquons que ce planning n'est pas toujours réalisé de manière suffisamment complète. Diverses raisons ont été fournies à ces imprécisions :

- Le projet-cadre n'étant pas suffisamment complet, beaucoup d'incertitudes subsistent en fin d'analyse conceptuelle et il est très difficile de réaliser un planning global et une analyse coûts-bénéfices suffisamment fiables.
- Les développeurs disposent d'un outil de planning Superproject dont ils ne sont pas satisfaits : il est trop compliqué, exige l'introduction d'un trop gros volume d'information compte tenu du temps dont ils disposent, il est irréaliste car il ne permet pas la gestion simultanée de plusieurs projets et il permet principalement d'estimer les durées alors que les développeurs sont également très intéressés par l'estimation des charges (ressources nécessaires). De plus, un tel outil nécessite une formation préalable des développeurs car il est trop souvent mal utilisé.

3. Le schéma global du nouveau système

Outils utilisés : Excelerator : 30 %
Datamanager : 20 %
traitement de texte : 20 %
aucun : 30 %

Le schéma global du système est imposé par l'équipe de support "Méthodes" et est réalisé sous la forme d'un diagramme de flux pour la partie traitement et sous la forme d'un modèle entité/association pour la partie données. Ces deux graphiques peuvent être documentés textuellement dans Datamanager mais les développeurs se contentent très souvent de la version graphique qui est utilisée lors des rencontres avec les utilisateurs. Le diagramme des flux leur fournit une vue des tâches principales du nouveau système ainsi que des flux de données qui seront nécessaires à son fonctionnement. Le modèle entité/association permet d'avoir une vue de toutes les données susceptibles d'être utilisées par le système et des relations pouvant exister entre ces données. Ces types de schémas sont jugés très utiles par l'utilisateur car ils sont généralement aisés à comprendre bien que l'utilisateur ne reçoive aucune formation concernant les modèles conceptuels. Il dispose uniquement d'un "Workshop" (recueil d'exercices d'application) pour les modèles conceptuels des traitements et des données; ce qui ne le satisfait pas. L'utilisateur se sert du schéma global du système pour juger si le nouveau système répond bien à sa demande. De plus le modèle conceptuel des données est indispensable pour la réalisation du schéma logique des données.

Etant donné que la spécification du projet n'est pas complète dès le départ, il est difficile de réaliser ces schémas avec précision et leur version finale n'est généralement obtenue qu'après plusieurs tentatives.

Un certain nombre de développeurs (17 %) déclare ne pas réaliser le schéma global et cela pour les raisons suivantes :

- la réalisation du schéma global est souvent manuelle et prend donc beaucoup de temps d'autant plus que beaucoup de modifications sont réalisées avant d'obtenir la version définitive !
- les délais imposés poussent les développeurs à arriver rapidement aux détails techniques et à ne pas trop s'attarder sur cette première esquisse du système. C'est pourquoi ils préfèrent entamer directement l'analyse fonctionnelle.
- L'outil EXCELERATOR est un outil graphique très puissant et efficace mais relativement neuf, ce qui explique le fait qu'il n'est encore utilisé que par une minorité des développeurs. Sa généralisation apporterait une amélioration certaine à la qualité de la documentation graphique ainsi qu'une standardisation du mode de représentation, ce qui serait très bien accueilli par les utilisateurs.

4. Le schéma logique des données commenté

Outils utilisés : Datamanager : 83 %

Excelerator : 17 %

Le schéma logique des données est basé sur le schéma conceptuel des données et constitue une étape obligatoire pour tout projet entraînant la création ou la modification d'une ou plusieurs bases de données. Il représente pour bon nombre de développeurs la documentation principale de l'analyse fonctionnelle. La documentation textuelle de ce schéma est standardisée dans Datamanager et sera étudiée et vérifiée par le groupe de support "méthodes" qui s'en servira pour réaliser une estimation de la charge de travail basée sur la méthode IFPA (vue au premier chapitre). C'est pourquoi, le groupe insiste fort pour que le schéma logique des données soit documenté de façon détaillée dans le dictionnaire.

Les quelques développeurs affirmant ne pas réaliser ce schéma, délèguent en fait ce travail au groupe chargé des bases de données en raison de la difficulté de ce travail.

5. Le planning détaillé de l'analyse technique

Outils utilisés : Superproject : 60 %

traitement de texte : 20 %

aucun : 10 %

La direction exerce une contrainte importante sur les développeurs en ce qui concerne le respect des délais fixés de réalisation des projets. Tout dépassement de date entraîne des coûts supplémentaires que la direction veut éviter au maximum. C'est pourquoi un des objectifs principaux des développeurs de la CGER est le respect des délais fixés de réalisation. Ainsi, ils s'efforcent de planifier au mieux les phases d'analyse et de codage. Quoique le nombre de développeurs estimant leur planning complet est plus élevé que pour le planning global, 30 % des plannings restent sommaires. Les explications fournies à cela sont :

- A la fin de l'analyse fonctionnelle, il reste toujours des incertitudes concernant les spécifications du système et il est difficile de planifier de manière très exacte le déroulement de l'analyse technique.
- La structure des différentes tâches du système constitue un input indispensable pour l'utilisation de l'outil de planning Superproject et est documentée dans Datamanager. Cependant, il n'existe pas d'intégration entre les deux outils, ce qui implique que ces données doivent être introduites deux fois et cela ne plaît guère aux développeurs.

6. Mesures de sécurité et de confidentialité

Outils utilisés : aucun

Etant donné la valeur hautement stratégique des informations traitées par les ordinateurs de la CGER, la gestion de la sécurité informatique est une tâche très importante et complexe qui est prise en charge par un groupe particulier de personnes ayant une formation en informatique et dépendant directement de la direction du département informatique. Ce groupe gère tous les aspects de la sécurité informatique, que ce soit la sécurité des bâtiments, le plan en cas de désastre ou les autorisations d'accès aux ressources sur mainframe et a élaboré pour ce faire des procédures très strictes. Celles-ci sont documentées dans un guide de l'utilisateur. En ce qui concerne les développeurs, leur tâche est grandement facilitée par l'existence de ce guide et de formulaires standards de demandes d'accès aux ressources qu'ils complètent chacun et font parvenir au groupe de sécurité.

En ce qui concerne les développements sur minis, les développeurs contactent le groupe spécialiste des minis pour obtenir la liste des routines générales de sécurité.

7. Schéma physique des données

Outils utilisés : Datamanager : 70 %
Excelerator : 10 %
traitement de texte : 10 %
aucun : 10 %

Le schéma physique des données constitue la documentation principale de l'analyse technique au niveau des données. Il représente la dernière étape indispensable avant la réalisation proprement dite des bases de données du système. La documentation textuelle de ce schéma est standardisée dans Datamanager et sera étudiée et vérifiée par le groupe chargé de la réalisation des bases de données. Cette standardisation convient parfaitement aux développeurs puisque tous estiment réaliser une documentation tout à fait complète du schéma physique. Les quelques développeurs affirmant ne pas réaliser ce schéma, sont ceux qui réalisent eux-mêmes les bases de données dont ils ont besoin et de ce fait, ils ne prennent pas le temps de documenter le schéma physique dans Datamanager.

Vu que Datamanager n'offre pas la possibilité de stocker le schéma sous forme graphique, Excelerator devrait être très utilisé en tant qu'outil graphique. Cependant ce n'est pas le cas et ceci pour diverses raisons :

- L'outil est encore mal connu et la promotion interne en est faible. De plus, son utilisation nécessite un apprentissage préalable auquel beaucoup de développeurs ont peu de temps à consacrer.

- L'intégration de Excelerator avec Datamanager est réalisée au niveau des données mais est encore très lourde car tout passage de données doit se faire via le groupe de support "méthodes" qui est à l'origine de cette intégration. Les développeurs préfèrent donc réaliser le graphique manuellement.

8. Spécifications des programmes

Outils utilisés : Datamanager : 100 %
traitement de texte : 40 %

Les spécifications des programmes font partie de la documentation minimale imposée par l'équipe de production au niveau du dossier fonctionnel (Fudos) du dictionnaire. Pour chaque application du système, elle exige de réaliser un schéma et de documenter la découpe de chaque application en sous-applications puis en programmes. Ensuite, pour chaque programme, on documente les autres programmes auxquels il fait appel, les écrans qu'il nécessite, les rapports qu'il produit ainsi que les fichiers qu'il manipule. De cette manière, on peut donc déterminer l'input et l'output de chaque programme ainsi que les relations entre les programmes. Le but visé est de documenter de manière concise la fonction logique de l'application afin de donner une idée de la succession des travaux dans l'application. Le développeur est invité à réaliser cette documentation avec suffisamment de précision sous peine de ne pouvoir transférer l'application en production. Cependant, vu les faibles capacités graphiques de Datamanager, la réalisation des schémas des applications est très laborieuse et l'équipe de production déplore le manque de précision de ces schémas. Cette documentation sera également très utile pour réaliser la maintenance ultérieure du système.

9. Descriptions détaillées des programmes

Outils utilisés : éditeur de programmes

Les commentaires au niveau des programmes sources sont très utilisés car ils permettent de documenter la découpe logique des programmes en modules et d'indiquer la fonction remplie par chaque module. Ces commentaires facilitent la lecture des programmes notamment pour les gens de la production et plus tard par d'autres programmeurs qui seront chargés d'en assurer la maintenance. Outre les commentaires, il est également possible de documenter les programmes dans Datamanager de manière textuelle par la réalisation d'ordinogrammes. Cette documentation est rarement effectuée car elle demande beaucoup de temps.

10. Cas et données de test des programmes

Outils utilisés : Datamanager 20 %
XPERT 40 %

Les développeurs sont invités à utiliser la méthode des tables de décision pour déterminer les cas de test car elle permet de vérifier si ces tests sont complets.

Pour les données, le groupe de support 'outils' met à leur disposition le logiciel XPERT orienté données et qui permet de réaliser les fichiers de données de test par extraction des fichiers existants. La liste des fichiers créés est fournie par le logiciel.

Bien que les chiffres semblent montrer que les cas de test soient bien réalisés, le groupe de support 'outils' estimant les tests réalisés par les programmeurs insuffisants, a réalisé un guide de test des programmes qui fournit notamment des indications sur la formation des cas et données de test. Au niveau de la documentation, il exige de réaliser un document "cas de test" pour chaque programme qui sera placé dans le dossier "papier" du programme et qui devra bien déterminer le but de chaque test (c'est-à-dire présenter une sélection judicieuse d'un certain nombre de fonctions du programme dont on vérifiera le travail).

Au niveau des données de test, il est demandé de documenter les fichiers utilisés au niveau de Datamanager dans une rubrique spéciale qui permet d'enregistrer les noms des fichiers utilisés ainsi que la manière dont ils ont été créés.

Datamanager n'offre guère de possibilités au niveau des tests : outre les noms des fichiers, il permet juste de garder la trace des tests des programmes en enregistrant pour tout programme présent dans le dictionnaire, la date de réalisation du test et le nom de la personne l'ayant fait.

11. Dossier de documentation des programmes

Outils utilisés : Datamanager : 56 %
traitement de texte : 22 %
aucun : 33 %

Suite aux problèmes et erreurs rencontrés au niveau des tests, l'équipe de préparation exige de mettre à jour la documentation de Datamanager relative aux différents programmes et routines (et qui est standardisée par l'équipe de production dans le dossier fonctionnel 'Fudos'). Cette documentation sera principalement utilisée par l'équipe de préparation chargée de préparer et de planifier les travaux informatiques avant leur passage en production.

A côté de cette documentation obligatoire, certains développeurs réalisent une farde de documentation par programme qui contient les schémas du programme, le programme source commenté accompagné de la documentation du dictionnaire relative à ce programme, le dossier "cas de test" du programme et les commentaires sur les tests réalisés par le programmeur. Généralement, l'équipe qui a développé le système est celle qui en assurera la maintenance et ce dossier contient les informations nécessaires qui l'aidera lors de modifications ultérieures du système. La réalisation de tels dossiers signifie donc que la documentation des programmes dans Datamanager n'est pas suffisante pour assurer une maintenance efficace du système.

12. Job Control

Outils utilisés : Datamanager : 100 %

A la CGER, les développeurs sont tenus de réaliser un Job Control par programme présenté sous forme d'un simple pseudo-code au niveau de Datamanager. Ce document est entièrement standardisé par l'équipe de production. Ces rubriques principales sont :

- une description générale de la fonction logique du programme qui donne une idée de la succession des travaux dans ce programme.
- une documentation de tous les paramètres d'exécution que le groupe de préparation devra introduire.
- une description des instructions de RESTART à suivre en cas d'interruption anormale de l'exécution et les instructions de RERUN si le programme doit être réexécuté après une première exécution réussie.
- une description des instructions à suivre si le programme est devenu inexécutable pour une raison inconnue.

L'équipe de préparation est chargée de transformer ce 'Job Control de développement' en 'Job Control de production' qui sera nécessaire quotidiennement pour l'équipe de production. L'objectif à long terme est d'automatiser cette transformation mais la standardisation actuelle laisse encore trop de place aux erreurs et imprécisions (10 % des développeurs estiment leurs 'Job Control' incomplets).

L'équipe de préparation-production travaille ardemment à l'amélioration de ses standards dont les différentes versions ne cessent de se succéder.

13. Résultats du test d'intégration

Outils utilisés : traitement de texte : 40 %
aucun : 60 %

Les tests d'intégration des différents programmes du système sont réalisés par les développeurs qui réalisent ensuite la liste détaillée des défauts constatés. Il est normal que ce document soit réalisé avec beaucoup de soin

car il est primordial de détecter et de corriger le maximum de défauts si l'on veut éviter de gros problèmes de fonctionnement lors de la mise en production.

14. Plan de tests (phases 5 & 6)

Outils utilisés : traitement de texte : 40 %
aucun : 60 %

Les tests des programmes et les tests d'intégration étant tous réalisés par les développeurs, les deux étapes ne sont pas organisées de manière indépendante (contrairement à ce que SDM propose). Aussi, le plan de tests est réalisé dès la fin de l'étape de codification des programmes (4.3) et inclut le plan de test des programmes, le plan du test d'intégration et le plan du test d'acceptation. L'objectif de ce plan est d'assurer une utilisation la plus efficace possible du temps consacré aux tests car, la date prévue de mise en production étant proche, les délais sont toujours trop courts pour pouvoir tester tous les cas possibles.

Les résultats nous montrent que seulement la moitié des développeurs sont satisfaits des plans de test. En fait, le problème se situe au niveau du plan du test d'acceptation : ce test est réalisé par des testeurs choisis parmi les futurs utilisateurs du système. Ces personnes ne reçoivent aucune formation préalable et ne sont donc guère motivées, elles se contentent souvent d'une exploration hâtive du système à la recherche des fonctionnalités qu'il offre. Il en résulte qu'actuellement, le plan du test d'acceptation est à la charge du développeur, ce qui ne devrait pas être car ce test doit être le plus indépendant possible du Développement.

Nous sommes donc confrontés à un gros problème d'organisation et pour le résoudre, il est indispensable de fixer des procédures d'organisation plus précises qui fixeraient très clairement les tâches respectives des testeurs et des développeurs. Il est également nécessaire de développer des cours destinés aux testeurs qui leur enseigneraient la démarche à suivre pour réaliser un plan de test ainsi que des cas de test. Dans cette optique, un nouvel environnement appelé "Quality Assurance" a été créé récemment dans le département informatique. Son rôle sera de veiller à la qualité des systèmes développés en contrôlant la réalisation des tests d'intégration et d'acceptation et en assurant l'écologie des utilisateurs chargés des tests.

15. Spécification des tests d'intégration et d'acceptation

Les spécifications des tests d'intégration sont réalisés par les développeurs en même temps que celles des tests des programmes. En ce qui concerne les tests d'acceptation, les problèmes ont déjà été envisagés au point précédent, lors de la réalisation du plan de test.

16. Rapport du test d'acceptation

Outils utilisés : traitement de texte : 60 %
aucun : 40 %

Le rapport du test d'acceptation est réalisé par les utilisateurs qui ont testé le système. Il contient la liste de tous les défauts et erreurs constatés au niveau des fonctionnalités du système ainsi que toutes les remarques concernant la facilité d'apprentissage du système, l'enchaînement des commandes, la rapidité d'exécution, la présentation des écrans et le contenu de l'output.

Ce rapport constitue un instrument de contrôle et de décision pour la direction et les utilisateurs en ce qui concerne la qualité du système et les dispositions à prendre pour les éventuelles améliorations à apporter.

17. Planning détaillé de conversion et mise en production

Outils utilisés : Superproject : 22 %
traitement de texte : 45 %
aucun : 33 %

Les développeurs tentent de réaliser un plan des différentes activités de conversion et de mise en production de manière à pouvoir contrôler efficacement l'avancement du travail afin de respecter le plus précisément possible la date prévue de démarrage du projet. Toutefois, ils regrettent le manque d'efficacité de Superproject qui ne leur permet pas de réaliser des plannings suffisamment précis en raison du volume excessif des données à fournir en entrée et de la complexité de l'outil.

18. Rapport final de la phase de conversion et mise en production

Outils utilisés : traitement de texte : 67 %
aucun : 33 %

Les développeurs estiment indispensable de réaliser un rapport final de la phase de conversion et de mise en production car il permet de spécifier tous les problèmes rencontrés durant cette étape et par conséquent, de justifier un éventuel retard par rapport à la date prévue de mise en production.

3.3.2 La documentation jugée utile

a. SYNTHÈSE

PHASE 1 : analyse conceptuelle	(1)	(2)
1. Points non résolus et hypothèses de base	75 %	44 %
2. Solutions alternatives et sélection	75 %	44 %
3. Moyens & coûts pour conversion et mise en prod.	75 %	55 %
PHASE 2 : analyse fonctionnelle		
4. Schéma des sous-systèmes et descriptions	75 %	89 %
5. Rapport final de l'analyse fonctionnelle	67 %	37 %
6. Précision des exigences et du cadre de fonctionnement du système	58 %	43 %
PHASE 3 : analyse technique		
7. Plan détaillé de programmation et de tests	67 %	63 %
8. Input et output de l'ordinateur	50 %	67 %
9. Rapport final de l'analyse technique	50 %	67 %
PHASE 4 : programmation et tests isolés		
PHASE 5 : tests du système		
10. Récapitulatif des résultats des tests	60 %	33 %
PHASE 6 : conversion et mise en production		

(1) pourcentage de développeurs qui réalisent ce type de documentation

(2) pourcentage de développeurs qui réalisent ce type de documentation de manière complète.

b. INTERPRETATION DES RESULTATS

1. Points non résolus et hypothèses de base

Outils utilisés : traitement de texte : 44 %
aucun : 56 %

La majorité des développeurs trouve cette liste utile au niveau du suivi du projet : en effet, non seulement elle leur permet de garder en vue les problèmes à résoudre mais permet aussi d'indiquer les mesures qui sont prises pour y remédier. Ces informations temporelles permettent donc de suivre la résolution des problèmes. Quant à la minorité, elle prétend que les points à résoudre la préoccupent suffisamment pour ne pas les oublier et ne juge pas utile de les documenter. En ce qui concerne les hypothèses de base, les développeurs ne voient guère de quoi il s'agit.

2. Solutions alternatives et sélection

Outils utilisés : traitement de texte : 60 %
aucun : 40 %

Ce document est jugé important car il est soumis aux utilisateurs et à la direction qui vont décider si le projet peut être poursuivi ou si la solution retenue ne satisfait pas les utilisateurs car elle ne répond pas suffisamment à leurs exigences ou encore ne satisfait pas le management car elle nécessite des investissements non souhaités. Seuls 44 % des développeurs réalisent ce document complètement car la technique utilisée est en général bien connue et le nombre de solutions alternatives, très restreint. De plus, les développeurs avouent que pour des raisons de délais, ils se mettent déjà d'accord avec l'utilisateur sur un type de solution dès l'élaboration du projet-cadre et précisent déjà certains aspects techniques.

3. Moyens & coûts pour conversion et mise en production

Outils utilisés : aucun

Etant donné que la plupart des projets de développement utilisent les technologies existantes, les développeurs réalisent dès l'analyse conceptuelle, une première estimation des ressources qui devront être allouées au nouveau système. Au niveau de la conversion, il s'agira d'évaluer la place à réserver au niveau des supports physiques des données et pour la mise en production, les ressources de la salle machine à réserver. Ces estimations sont très sommaires et seront détaillées tout au long du développement. Toutefois, lorsque les conversions à effectuer sont très complexes, les développeurs jugent préférable d'en faire un projet particulier afin de faciliter le développement et en diminuer le risque.

4. Schéma des sous-systèmes et descriptions

Outils utilisés : Datamanager : 90 %

Excelerator : 13 %

Les développeurs réalisent généralement un premier diagramme des flux lors de la première phase, qui présente la découpe du système en plusieurs sous-systèmes et montre ainsi les différentes activités à élaborer. Durant cette deuxième phase, ils reprennent ce premier diagramme de contexte et le détaillent par la technique d'explosion des diagrammes de flux jusqu'au niveau qu'ils jugent élémentaire. On peut donc regrouper ici les activités suivantes :

- Division en sous-systèmes (2.3).
- Définition des entrées et sorties de chaque sous-système ainsi que des interfaces entre eux (2.4).
- Réalisation des diagrammes des traitements (2.5/6).

Les diagrammes de flux permettent de distinguer les entrées et sorties, les interfaces entre les différents sous-systèmes, les accès aux bases de données ainsi que les interfaces homme/machine. Ils sont réalisés par tous les développeurs car ils sont riches en information et sont bien compris par les utilisateurs. La documentation de ces diagrammes est possible sous forme textuelle dans Datamanager.

En ce qui concerne la description des écrans, la CGER dispose d'un outil de prototyping qui permet de construire des écrans élémentaires pour des transactions interactives. Cependant les résultats de l'enquête ne mentionnent aucune utilisation de cet outil.

Les descriptions des traitements réalisées dans Datamanager permettent de décrire chaque sous-système en un certain nombre d'activités elles-mêmes décomposées en tâches (cfr. la structure hiérarchique des activités et des tâches, point 1.2.5). Cette documentation constitue le dossier fonctionnel du projet. Quant à la documentation graphique, elle est généralement faite manuellement et est rarement complète car elle exige beaucoup de temps. On constate que l'outil Excelerator est très peu utilisé pour la réalisation de la documentation graphique, ce qui montre une fois de plus qu'il est encore très mal connu des développeurs.

Il est regrettable de constater que les développeurs ne réalisent pas cette documentation avec plus de rigueur car il s'agit tout de même des spécifications fonctionnelles du projet qui serviront de base et de référence pour réaliser l'analyse technique et la programmation.

De plus, elles constituent la documentation du futur système qui sera soumise à l'approbation des utilisateurs. Plus elles sont complètes et précises, plus elles auront de chance d'être acceptées et plus grande est la probabilité qu'il n'y aura plus de modifications ultérieures importantes de la logique du

système. Or, ces modifications tardives coûtent cher car elles engendrent des feedback importants et donc des pertes de temps.

Elles servent également de base pour l'évaluation de la charge de travail pour le projet grâce à la méthode nouvellement introduite par le groupe 'Méthode': la méthode d'analyse par points de fonction qui permet, à partir du modèle logique des données et des descriptions des tâches d'évaluer le temps de développement des systèmes informatiques.

La description des tâches est également indispensable pour exécuter le logiciel PALOMA développé par le groupe 'méthodes', qui permet d'évaluer la charge de chaque tâche sur base des accès aux données qu'elle nécessite. Ces estimations permettront de quantifier le modèle conceptuel et, par la suite, réaliser le modèle logique des données.

Pour résoudre ce problème nous proposons au groupe "Méthodes" de réaliser un modèle de réalisation des spécifications fonctionnelles bien défini qui réponde aux besoins des utilisateurs tout en étant utile pour le développeur. Idéalement ce modèle s'accompagnerait d'exemples concrets d'utilisation afin de pouvoir juger de l'utilité de chaque concept. Ce modèle pourra ensuite être implémenté dans l'outil Excelerator pour permettre la réalisation de graphiques et la création de la documentation du modèle au niveau du dictionnaire local. Ensuite, dès que l'intégration entre Excelerator et Datamanager sera entièrement terminée, il sera possible de documenter automatiquement les spécifications fonctionnelles dans Datamanager par simple transfert d'information.

5. Rapport final de l'analyse fonctionnelle

Outils utilisés : Datamanager : 60 %
traitement de texte : 20 %
aucun : 37 %

Il est clair que le rapport final de l'analyse fonctionnelle existe pour tout projet dans le dictionnaire central et est intitulé 'dossier fonctionnel'. Le document dont il est question dans ce tableau consiste en un rapport récapitulatif sur papier qui est rédigé par le chef de projet et qui reprend l'ensemble des résultats de la phase. Le fait que seulement 67 % des développeurs le réalisent s'explique par diverses raisons :

- Beaucoup se contentent d'imprimer le dossier fonctionnel du dictionnaire et d'y ajouter les divers graphiques qu'ils auront réalisés (schéma logique des données, diagrammes de flux, schémas de la découpe hiérarchique des sous-systèmes, ...)
- Les différents résultats de la phase ne peuvent être intégrés par un outil unique qui permettrait de générer automatiquement le rapport final. C'est pourquoi les développeurs considèrent la réalisation de ce rapport comme une tâche redondante et y consacrent un minimum de temps.

6. Précision des exigences et du cadre de fonctionnement du système

Outils utilisés : Datamanager : 57 %
traitement de texte : 43 %

La précision des exigences et du cadre de fonctionnement consiste à modifier le projet-cadre réalisé durant l'analyse conceptuelle suite à un changement de contexte. Nous constatons que pour plus de la moitié des développements, des changements ont lieu et ceux-ci s'expliquent par le fait que les exigences des utilisateurs manquent généralement de précision au départ. Comme les utilisateurs sont sous le même toit que les développeurs, il leur est facile de contacter ces derniers à tout moment du développement pour des précisions supplémentaires ou des modifications des exigences du système. Nous regrettons le fait que la majorité des modifications apportées ne donnent lieu qu' à des mises à jour sommaires du projet-cadre car cela signifie qu'il subsiste des incohérences entre les objectifs définis dans le projet-cadre et le système développé.

7. Plan détaillé de programmation et de tests

Outils utilisés : superproject : 63 %-
aucun : 37 %

Les développeurs se basent sur le planning global réalisé en fin d'analyse fonctionnelle et précisent ce dernier en fonction des différents programmes à coder (désormais connus) et de leur complexité ainsi que du nombre de développeurs et de leur expérience. Leur objectif est de répartir au mieux le temps disponible (toujours trop court aux yeux des programmeurs) afin que les programmeurs aient suffisamment de temps pour réaliser des programmes clairs, bien structurés et pour y introduire tous les commentaires nécessaires. Cependant, les développeurs déclarent ne pas disposer d'un outil suffisamment efficace pour réaliser un plan bien concis.

Au niveau du plan des tests, nous avons vu que les développeurs intègrent en une seule phase, les tests des programmes, d'intégration et d'acceptation et par conséquent, ils sont planifiés entièrement dès la fin de l'étape de codage (ce plan de test a été vu au point 6 de la documentation indispensable).

8. Input et output de l'ordinateur

Outils utilisés : Datamanager : 50 %

Il s'agit de documenter l'aspect physique de toutes les entrées et sorties de l'ordinateur qui ont été décrites au niveau des spécifications fonctionnelles. Cette documentation est standardisée au niveau de Datamanager dans le dossier "tedos" de l'analyse technique qui, pour chaque rapport, écran et

fichier permet de documenter les données qu'ils contiennent en termes de records, groupes et champs qui les composent.

Le fait que seulement 50 % des développeurs réalisent cette documentation s'explique par la durée que prend cette tâche si elle se veut complète et par le fait que les programmeurs considèrent la documentation comme une perte de temps. Ils donnent la priorité au codage et ne documentent que le strict minimum.

9. Rapport final de l'analyse technique

Outils utilisés : Datamanager : 84 %
aucun : 16 %

Ce rapport devrait servir de base pour le programmeur or, vu l'absence d'outil permettant d'intégrer tous les types de documentation, les programmeurs situés à proximité du chef de projet communiquent beaucoup de manière informelle et se contentent des spécifications fonctionnelles réalisées.

Quant aux développeurs qui déclarent réaliser ce dossier, ils se contentent généralement d'imprimer le dossier d'analyse technique du dictionnaire Datamanager et d'y ajouter le graphique du schéma physique des données. Ce dossier est alors transmis à la direction et aux utilisateurs pour qu'ils puissent juger de l'évolution du développement.

10. Récapitulatif des résultats des tests

Outils utilisés : traitement de texte : 100 %

La correction des erreurs détectées durant les différents tests peuvent entraîner des modifications importantes de la documentation du système au niveau de l'analyse technique (documentation des programmes, de la structure de stockage) et peut-être même au niveau de l'analyse fonctionnelle si des erreurs importantes ont été commises au niveau de la logique des programmes. C'est pourquoi les développeurs soucieux de la cohérence de leur documentation, réalisent ce document récapitulatif des résultats de tests qui décrit toutes les modifications apportées au système et qui leur permettra par la suite d'effectuer les mises à jour de la documentation. Vu l'importance que nous attachons à la cohérence globale de la documentation, nous regrettons que ce document ne soit pas réalisé par tous les développeurs et qu'il soit aussi sommaire.

Le problème vient du fait que la majorité des développeurs refusent toujours de considérer la réalisation de la documentation comme une activité indispensable du développement d'un logiciel. Pour preuve, elle n'est que très rarement prise en compte dans les plannings si ce n'est pour permettre aux développeurs de se munir d'une marge de sécurité pour le respect des dates de fin de projet.

3.3.3 La documentation jugée superflue

a. SYNTHÈSE

PHASE 1 : analyse conceptuelle		
1. Rapport final de l'analyse conceptuelle	42 %	40 %
PHASE 2 : analyse fonctionnelle		
2. Exigences détaillées en matière de sécurité	20 %	50 %
3. Cahier des charges complet	42 %	60 %
PHASE 3 : analyse technique		
4. Manuel d'utilisation	42 %	60 %
PHASE 4 : programmation et test isolés		
1. Description des tâches manuelles	0 %	
PHASE 5 : tests du système		
PHASE 6 : conversion et mise en production		

- (1) pourcentage de développeurs qui réalisent ce type de documentation
- (2) pourcentage de développeurs qui réalisent ce type de documentation de manière complète.

b. INTERPRÉTATION DES RESULTATS

1. Rapport final de l'analyse conceptuelle

Outils utilisés : Datamanager : 48 %
traitement de texte : 52 %

Bien que le rapport final de la première phase constitue un document très important puisqu'il sert de base pour évaluer si le projet doit être poursuivi ou non, il n'est réalisé que par une minorité.

Tout d'abord, ce rapport n'étant qu'un résumé de la phase d'analyse conceptuelle, les développeurs avouent qu'ils intègrent souvent pour des raisons de temps, la phase conceptuelle et la phase fonctionnelle, ce qui les amène à ne pas tenir compte du point de contrôle situé entre les deux phases. C'est pourquoi ils négligent la réalisation de ce rapport.

Deuxièmement, les développeurs ne disposent pas d'outil permettant d'intégrer en un seul rapport les documents de planning, les schémas, les données du dictionnaire ainsi que les documents de traitement de texte. La réalisation de ce rapport est alors considérée comme une tâche redondante et c'est pourquoi ils se contentent donc souvent de réunir les copies des différents résultats dans une farde.

2. Exigences détaillées en matière de sécurité

Outils utilisés : aucun

Nous avons vu que la gestion de la sécurité est entièrement prise en charge par un groupe particulier qui a élaboré des procédures très précises à cet effet. Les développeurs ne réalisent donc aucune documentation particulière si ce n'est un document destiné aux futurs utilisateurs qui décrit les mesures de sécurité qui les concernent directement et dont ils devront tenir compte pour organiser leur travail.

En ce qui concerne la réalisation proprement dite de ces mesures, nous avons vu que le développeur dispose de formulaires de demandes standards qu'il remplit et fait parvenir au groupe de sécurité durant la phase d'analyse technique (activité 3.4).

En ce qui concerne la réalisation d'un plan d'urgence, elle est prise en charge par les utilisateurs avec l'aide du groupe de sécurité.

3. Cahier des charges complet

Outils utilisés : aucun

Le cahier des charges contient la liste de tous les achats de matériel que nécessite le fonctionnement du nouveau système. Très peu de développeurs réalisent ce cahier des charges car la plupart des développements concernés par cette enquête utilisent l'environnement technique existant et par conséquent, ils n'entraînent pas de gros achats.

4. Manuels d'utilisation

Outils utilisés : traitement de texte : 100 %

Peu de développeurs réalisent un manuel d'utilisation du système car leur souhait est que les utilisateurs ainsi que les testeurs réalisent eux-mêmes ces manuels car il s'agit d'une tâche qui exige beaucoup de temps et qui ne plaît généralement pas aux développeurs. De plus, ils réalisent souvent des manuels trop complets, trop volumineux et très théoriques qui sont très utiles pour la formation mais qui deviennent vite trop complets pour être efficaces. Les utilisateurs ont besoin de manuels plus condensés pouvant être utilisés tous les jours.

5. Description des tâches manuelles

Les développeurs ne sont pas concernés par la description des tâches manuelles que devront accomplir les utilisateurs du système. Il s'agit d'une activité qui incombe totalement au groupe des utilisateurs (cette tâche est davantage un problème d'organisation des utilisateurs eux-mêmes qu'un problème informatique). En général, les utilisateurs se contentent de formations "sur le tas" et ont toujours la possibilité de contacter les développeurs pour d'éventuelles questions.

3.3.4. Résumé de la situation actuelle de la documentation.

Dans ce quatrième et dernier point, nous allons utiliser la classification de la documentation de SDM (vue p 28) et proposer une synthèse de la documentation réalisée actuellement à la CGER ainsi que des principaux problèmes rencontrés.

A. LA DOCUMENTATION DU PROJET

Documentation de gestion de projet

La documentation de gestion de projet est sous la responsabilité du chef de projet et fait l'objet d'un dossier particulier. Il s'agit d'une documentation sur papier dont les principaux composants sont : la demande formelle de développement, la définition du groupe de travail, les plannings, les comptes-rendus des réunions, le courrier ainsi que les feuilles de facturation qui récapitulent les prestations des membres de l'équipe. Il s'agit d'une documentation très personnelle où la correspondance occupe une grande place. C'est pourquoi plusieurs chefs de projet estiment qu'il serait très utile de disposer d'un outil de suivi et de catalogue de courrier. Nous pensons que ce type d'outil est relativement standard et son acquisition pourrait faire gagner du temps aux chefs de projet.

La difficulté majeure de cette documentation concerne la réalisation de plannings précis et suffisamment fiables :

- Les chefs de projet ne sont pas satisfaits de l'outil Superproject et voudraient disposer d'un outil plus simple qui permettrait de gérer plusieurs projets simultanément et qui ne serait pas seulement un gestionnaire de temps mais également de ressources.
- De plus, les chefs de projet seraient très intéressés par une formation en matière de gestion de projet qui permettrait d'augmenter la qualité et l'efficacité de leur travail.

documentation des résultats des activités effectuées

Les développeurs documentent le plus possible dans le dictionnaire Datamanager qui constitue ainsi la source d'information principale concernant les projets informatiques de la CGER. Cet outil (pour rappel) propose trois dossiers principaux :

- FUDOS qui permet de documenter les résultats des phases d'analyse conceptuelle et fonctionnelle.
- TEDOS pour les phases d'analyse technique et de programmation
- PRODOS pour la phase de mise en production

Datamanager assure la centralisation, la standardisation et la communication automatique des données puisqu'il est utilisé par les environnements de développement, de quality-assurance, de préparation et production ainsi que par les représentants des utilisateurs encore appelés coordinateurs informatiques.

Le défaut majeur de Datamanager est qu'il ne permet pas de stocker de documentation sous forme graphique (modèles conceptuels des données et des traitements, schémas logique et physique des données, découpe hiérarchique des sous-systèmes, schémas des traitements, ...)

Les graphiques sont souvent réalisés manuellement et placés dans une farde. C'est une des raisons pour lesquelles, le groupe "méthodes" a acquis l'outil graphique Excelerator qui permet de faciliter et de standardiser la réalisation de la documentation graphique. Son intégration totale avec Datamanager permettra de définir les objets des différents graphiques dans le dictionnaire local de Excelerator et de transférer automatiquement ces définitions dans Datamanager.

Il est certain que les développeurs ne verront en Excelerator qu'un simple outil graphique tant que son intégration avec Datamanager ne sera pas terminée car ils sont très réticents à l'idée qu'une même information doit être introduite deux fois dans des outils différents.

B. LA DOCUMENTATION DU SYSTEME

Documentation technique du système

La documentation technique du système est utilisée pour réaliser la maintenance du système. A la CGER, un projet de maintenance doit se limiter à des modifications des résultats des phases de programmation et d'analyse technique. Si les modifications à réaliser remettent en cause les résultats de l'analyse fonctionnelle, elles feront l'objet d'un nouveau projet de développement.

L'environnement bancaire étant très dynamique, les projets de maintenance y sont nombreux et sont généralement réalisés par l'équipe qui a développé le système. Si la documentation technique du système est trop volumineuse, elle sera pénalisante lors de la réalisation de la maintenance car elle nécessitera un gros travail de mise à jour. Au contraire, si elle est trop faible, la maintenance sera plus difficile. Il s'agit donc de trouver le juste milieu.

La documentation technique du système se compose des principaux résultats de l'analyse technique. Au niveau des données, on trouve le schéma physique des données sous forme textuelle dans Datamanager et sous forme graphique, généralement réalisé manuellement sur papier. On dispose également des descriptions physiques de l'input et de l'output de chaque traitement dans la mesure où celles-ci sont réalisées et nous avons vu que ce n'est pas le cas pour la moitié des développements.

Au niveau des traitements, on dispose des spécifications des programmes. Il ressort des interviews que si les spécifications des programmes standardisées et imposées dans Datamanager par le groupe de production sont relativement bien élaborées, elles ne sont pas suffisantes pour pouvoir réaliser une maintenance efficace du système. Beaucoup admettent même l'existence d'un 'flou' au niveau documentation entre la fin de l'analyse technique et la mise en production. Ils expliquent cela par le fait qu'au niveau de l'analyse technique, il n'existe pas d'exigences suffisamment précises de la part des groupes de support au développement qui permettraient de savoir exactement ce qui doit être documenté ainsi que la manière de faire. Ils souhaiteraient des modèles de documentation précis pour Datamanager.

Ce flou se traduit par une très forte diversification de la documentation technique des projets d'un groupe de développement à l'autre. Certains réalisent des descriptions en pseudo-code de chaque programme ainsi que des descriptions très précises des entrées et sorties de ces programmes, d'autres entament la programmation directement à partir des spécifications fonctionnelles et du schéma physique des données. Nous pensons qu'il y a lieu d'uniformiser et de standardiser cette documentation.

Documentation de l'utilisateur

La documentation de l'utilisateur réalisée par les développeurs est de plus en plus sommaire ou même inexistante car ces derniers souhaitent que les utilisateurs assument eux-mêmes cette tâche, éventuellement à l'aide des testeurs (qui sont eux-mêmes des utilisateurs). Depuis quelques années est apparue à la CGER une nouvelle fonction

qui est celle de 'coordinateur informatique'. Actuellement au nombre de quatre, ces personnes issues du groupe des utilisateurs assurent le rôle de coordinateur entre les utilisateurs et les informaticiens. Le nombre d'utilisateurs des systèmes informatiques étant beaucoup trop élevé pour permettre une communication directe entre eux et les informaticiens, les coordinateurs informatiques se chargent de représenter les utilisateurs aux différentes réunions. Ce sont également eux qui sont chargés d'organiser le travail de l'équipe de testeurs lors des tests d'acceptation.

Au niveau de la documentation, ils aident les utilisateurs à réaliser des manuels d'utilisation du système et restent à leur disposition pour tout renseignement.

Le développement de cette fonction devrait permettre aux développeurs de se libérer de toute la documentation relative aux utilisateurs. Du côté des utilisateurs, ils disposent de représentants suffisamment formés et informés pour défendre leurs exigences devant les informaticiens et pour leur apporter l'assistance dont ils ont besoin.

Documentation de production

Compte tenu du nombre sans cesse croissant des applications informatiques développées à la CGER, il est devenu absolument nécessaire que les applications se présentent de façon uniforme et standardisée pour le groupe de production afin de permettre qu'au niveau de la production même, on puisse assurer une gestion automatique des aspects tels que le planning, l'allocation des espaces-disque, la gestion des cassettes et bandes magnétiques, ... Dans le but d'améliorer la qualité de ses services et de pouvoir garantir ce niveau de qualité pour l'avenir, le groupe de production a élaboré un document intitulé "contrôle du développement et standards de production" qui définit de manière très précise la documentation minimale d'une application. Chaque développeur est tenu de respecter ces standards sous peine de voir refuser son application en production. Cette documentation se situe principalement dans le dictionnaire Datamanager.

C. LA DOCUMENTATION HISTORIQUE

La documentation historique conservée par les développeurs est constituée de la farde contenant la documentation de gestion du projet.

3.4 L'AUTOMATISATION DE LA DOCUMENTATION

Dans ce quatrième point, nous vous présentons les types de documentation dont l'automatisation est jugée possible et est souhaitée par la majorité des développeurs (50 % et plus). Ensuite, nous verrons dans quelle mesure les outils existant à la CGER peuvent répondre à ces souhaits d'automatisation. Finalement, nous déduirons de ces résultats les objectifs prioritaires du groupe de support au développement.

3.4.1 Souhaits d'automatisation

PHASE 1 : Analyse conceptuelle	
Act 1.5 : Points non résolus et hypothèses de base	
Document : Liste des problèmes rencontrés	52 %
Act 1.10 : Planning et analyse coût-bénéfice	
Document : Planning global et analyse coût-bénéfice	63 %
PHASE 2 : Analyse fonctionnelle	
Act 2.3 à 2.6 : Division en sous-systèmes, entrées et sorties des sous-systèmes, description des traitements	
Document : Schéma de la découpe en sous-systèmes et schémas des traitements	81 %
Act 2.9 : Réaliser la structure logique des données	
Document : Schéma logique des données	75 %
Act 2.13 : Planning de la suite du développement	
Document : Plannings détaillés des phases suivantes	63 %
PHASE 3 : Analyse technique	
Act 3.3 : Réaliser la structure de stockage	
Document : Schéma physique des données	77 %
PHASE 5 : Tests	
Act 5.1 : Réaliser un plan de test	
Document : Plan de test	63 %

INTERPRETATION DES RESULTATS

Nous pouvons regrouper les résultats de ce tableau en trois grandes catégories : la documentation de gestion de projet, la description fonctionnelle du système et la documentation des bases de données.

a. La documentation de gestion de projet :

Dans cette catégorie, nous pouvons regrouper la liste des problèmes rencontrés, les différents plannings et analyses coût-bénéfice réalisés tout au long du développement ainsi que le plan de test. Ces souhaits d'automatisation traduisent principalement l'insatisfaction des chefs de projet à l'égard de l'outil de planning Superproject.

b. La description fonctionnelle du système :

Les spécifications fonctionnelles du système sont documentées textuellement dans Datamanager. Etant donné que cette documentation doit être utilisée et étudiée par les utilisateurs, on reproche à Datamanager son manque de convivialité. Les rapports réalisés par extraction des données du dictionnaire ne contiennent aucun graphique et sont difficiles à appréhender par les utilisateurs.

Partant du principe que "un dessin vaut mieux que 1000 mots", les développeurs souhaitent un outil graphique que permettrait de réaliser, de modifier et d'imprimer aisément les schémas suivants :

- les schémas de la découpe hiérarchique du système en sous-systèmes, traitements et tâches.
- les schémas de présentation des entrées et sorties de chaque sous-système ainsi que des interfaces entre sous-systèmes.
- les schémas des traitements.

Cet outil devrait également permettre d'effectuer des contrôles de cohérence entre les différents schémas et de montrer clairement les liens entre les différents schémas.

De plus, cet outil devra être intégré avec Datamanager car les développeurs ne sont pas favorables à une double documentation des spécifications fonctionnelles, la première sous forme graphique et la seconde sous forme textuelle.

c. La documentation des bases de données :

Les étapes logique et physique de conception des bases de données sont très liées puisque le schéma physique des données résulte de la transformation du schéma logique des données selon des règles connues. Il est de plus en plus question actuellement d'outils qui permettent d'automati-

ser cette transformation et les développeurs de la CGER sont favorables à un tel outil.

3.4.2 Demandes réalisables avec les outils existants

ACTIVITE	DOCUMENT	OUTIL
1. 1.5	Liste des problèmes rencontrés	Excelerator
2. 2.3 à 2.6	Schémas des spécifications fonctionnelles	Excelerator
3. 2.9	Schéma logique des données	Excelerator
4. 3.3	Schéma physique des données	(Bachman/DBA)

1. LISTE DES PROBLEMES RENCONTRES

a. Possibilités de Excelerator

Excelerator en tant qu'outil de gestion de projet permet de réaliser une base de données des problèmes rencontrés durant les phases d'analyse, que se soit au niveau des besoins d'information, des spécifications conceptuelles ou fonctionnelles, des procédures à suivre, ...

Ces problèmes sont enregistrés dans le dictionnaire de l'outil (au moyen d'un type d'objet particulier) et la base de données ainsi constituée peut être utilisée tout au long du développement pour y accumuler tous les problèmes rencontrés mais surtout pour estimer leur date de résolution et indiquer les mesures qui sont prises pour y remédier.

De plus, à chaque problème décrit il est possible de relier les objets du dictionnaire qui se situent à la base du problème.

La description d'un problème se structure de la manière suivante :

- La personne responsable de la résolution du problème
- Le degré de priorité du problème
- Le statut du problème : découvert, solution connue, planifié, testé, résolu.
- La date d'apparition, les dates planifiée, espérée et actuelle de résolution
- La description du problème en texte libre
- Les actions à entreprendre en vue d'une résolution.

A tout moment, le gestionnaire peut faire appel à des procédures d'analyse et de génération de rapport qui réalisent des sélections dans la base de données des problèmes sur base de leur statut.

Ainsi, il suffit au gestionnaire de spécifier le statut non "résolu" pour obtenir un rapport présentant tous les problèmes non encore résolus ainsi que la liste des objets reliés à ce problème.

b. Raisons évoquées à la non-utilisation de Excelerator

- Les possibilités de Excelerator en tant que gestionnaire de projet ne sont pas connues.
- La gestion des problèmes exige d'utiliser Excelerator dès le tout début du projet. Or bon nombre de projets actuels étaient déjà commencés lorsque Excelerator a été proposé aux développeurs.

2. SCHEMAS DES SPECIFICATIONS FONCTIONNELLES

De l'enquête, il ressort que Excelerator n'est utilisé que pour une très petite minorité de développeurs pour la réalisation des schémas faisant partie de l'analyse fonctionnelle. De plus, ceux-ci l'utilisent comme un simple outil de "dessin" alors qu'il permet de définir de documenter et de vérifier l'architecture du système. Cette sous-utilisation est due à une méconnaissance de l'outil.

a. Possibilités de Excelerator

Pour modéliser les traitements, Excelerator utilise les diagrammes de flux de données.

Un premier diagramme de flux permet de représenter un processus donné comme un ensemble d'autres processus qui remplissent chacun une certaine mission par la transformation de données. Chaque processus peut recevoir des données d'un autre, accéder à des collections de données et transformer les données reçues pour fournir un nouveau flux de données en sortie. On représente également des entités externes qui interagissent avec le système pour lui fournir ou recevoir des flux de données : il peut s'agir d'utilisateurs, de machines ou d'autres systèmes.

Un diagramme de flux comprend donc quatre types d'objet différents :

- des processus
- des collections de données
- des flux de données
- des entités externes

Excelerator offre deux techniques possibles de représentation des diagrammes de flux (Gane & Sarson ou Yourdon) qui ne diffèrent que par la forme des objets de représentation.

Il est possible de spécifier complètement le fonctionnement du système en créant un ensemble hiérarchisé de diagrammes de flux :

Un premier diagramme de flux est réalisé; il ne contient qu'un seul processus représentant l'entière du système. Ensuite, par le mécanisme d'explosion, on "explose" ce processus unique en un diagramme de flux plus détaillé. Il reste à continuer ces "explosions" jusqu'à ce qu'on ait atteint un niveau de détail suffisant. Ainsi, en partant du niveau du système, on descend au niveau des activités puis finalement au niveau des tâches. Chaque processus, collection de données et flux de données défini doit être soumis au mécanisme d'explosion :

- chaque processus explose vers un diagramme de flux de niveau inférieur ou vers les spécifications d'un processus de niveau élémentaire.
- chaque collection de données explose vers un certain nombre de groupes de données ou vers un diagramme du modèle des données.
- chaque flux de données explose vers un certain nombre de groupe d'éléments ou d'éléments.

Lorsque les diagrammes sont terminés et tous les objets sont décrits dans le dictionnaire interne, il est alors possible de générer automatiquement un rapport d'explosion de graphes qui documentera ainsi la structure hiérarchique du système.

De plus, Excelerator permet à tout moment d'effectuer des procédures de vérification qui, par exemple, permettront de contrôler :

- le respect des règles de représentation d'un diagramme de flux.
- la présence d'une description dans le dictionnaire pour chaque objet présent dans les diagrammes.

Chaque procédure de contrôle se termine par la génération d'un rapport indiquant toutes les anomalies et erreurs détectées.

b. Raisons évoquées à la non-utilisation de Excelerator

- La promotion de l'outil est trop faible et les possibilités en sont mal connues.
- L'outil est très puissant et exige un apprentissage auquel les développeurs n'ont que très peu de temps à consacrer.
- La récupération des définitions des traitements de Excelerator est impossible dans Datamanager. C'est pourquoi bon nombre de développeurs n'utilisent pas Excelerator afin d'éviter des introductions multiples.

- Les développeurs souhaiteraient disposer d'un exemple complet de documentation avec Excelerator.

3. SCHEMA LOGIQUE DES DONNEES

a. Possibilités de Excelerator

Excelerator permet de définir le schéma logique des données à partir du schéma conceptuel des données réalisé durant l'analyse conceptuelle.

Le schéma conceptuel des données réalisé à l'aide des diagrammes entité/relation permet de décrire les données et les relations existant entre elles au moyen des objets "types d'entité" et "types d'association". Chaque objet du schéma réalisé est ensuite défini dans le dictionnaire local.

Le schéma logique des données doit contenir des agrégats (records), des champs (elements), des clefs d'accès ainsi que des domaines de valeur et doit être normalisé.

La conversion du schéma conceptuel en schéma logique se réalise en deux phases : la définition du modèle des données puis la normalisation du modèle des données :

- Définition du modèle des données

On définit le modèle des données par l'application du mécanisme d'explosion au niveau des définitions présentes dans le dictionnaire de Excelerator.

A partir des définitions des types d'entité et des types d'association, on expose vers des définitions d'agrégats et de champs auxquels sont associées des définitions de clés et de domaines de valeur.

Une fois les définitions terminées, on fera appel aux procédures d'analyse et de contrôle de cohérence qui fourniront automatiquement des rapports qui permettront entre autres de vérifier que tous les types d'entité du graphe conceptuel sont définis dans le dictionnaire, que tous les types d'entité explosent bien vers la description d'un agrégat, que tout agrégat contient au moins un élément et qu'il n'existe pas d'agrégats identiques (redondants), ...

- Normalisation du modèle des données

Après avoir vérifié que les description du modèle des données sont complètes, cohérentes et non redondantes, il reste à normaliser le modèle des données, c'est-à-dire définir l'organisation logique des agrégats du modèle des données conformément à une forme normale.

Excelerator propose des procédures de vérifications qui détectent toute violation par rapport aux trois premières formes normales (pour plus de détail, consulter HAINAUT (8)) et qui génèrent des rapports qui indiquent toutes ces violations.

Il est très important de voir que Excelerator ne fait qu'aider à détecter les endroits de violation des formes normales et n'effectue aucune réorganisation des agrégats du modèle.

b. Raisons évoquées à la non-utilisation de Excelerator

- Excelerator est intégré à Datamanager au niveau de la description des données mais le transfert est difficile, long à effectuer et doit être obligatoirement réalisé par l'équipe de support "Méthodes" qui est à l'origine de l'intégration. Ce passage obligatoire est une contrainte pour le développeur mais permet à l'équipe de support d'exercer un contrôle sur les informations transférées.
- Excelerator n'est qu'un outil d'aide à la réalisation de la documentation. Un outil vraiment intelligent devrait automatiser la normalisation du modèle des données sans aucune intervention humaine. Ce type d'outil existe sur le marché.

4. SCHEMA PHYSIQUE DES DONNEES

Excelerator est un Upper Case tool, ce qui signifie qu'il est principalement utilisé durant l'analyse conceptuelle et l'analyse fonctionnelle. Au niveau de l'analyse technique, aucun outil n'est présent actuellement.

Pour faciliter la conception physique des bases de données relationnelles (DB2), le groupe d'administration des bases de données étudie actuellement un Lower Case tool d'IBM appelé Bachman/DBA (Database Administrator). Il s'agit d'un ensemble d'outils très puissant et intelligent capable de prendre en charge la réalisation des bases de données, du niveau conceptuel jusqu'à la génération des tables DB2. De plus, il offre des possibilités de reverse engineering très intéressantes pour réaliser des modifications des bases de données existantes.

Aux niveaux conceptuel et fonctionnel, cet outil est en concurrence avec Excelerator, ce qui représente un gros obstacle à son éventuelle acquisition compte-tenu du coût d'un tel outil.

Les résultats actuels de l'étude en cours à la CGER ainsi que de la documentation concernant l'outil ne nous permettent pas de fournir des informations plus détaillées sur cet outil.

3.4.3 Objectifs prioritaires du groupe de support

Les résultats de l'enquête montrent qu'une grande partie des souhaits des développeurs sont déjà réalisables avec l'outil Excelerator. C'est pourquoi nous pouvons dire que les objectifs du groupe de support consistent davantage à promouvoir les possibilités de Excelerator plutôt qu'à rechercher de nouvelles possibilités pour cet outil.

Suite à cette enquête, nous sommes en mesure de définir trois objectifs prioritaires pour le groupe de support au développement :

1. REALISER UNE PROMOTION "COMMERCIALE" DE EXCELERATOR
2. REALISER UNE INTEGRATION SIMPLE ET COMPLETE ENTRE EXCELERATOR ET DATAMANAGER
3. ENVISAGER LES POSSIBILITES D'UN NOUVEL OUTIL DE PLANNING

1. REALISER UNE PROMOTION "COMMERCIALE DE EXCELERATOR

Les développeurs du département informatique n'ayant pas pris part à la décision d'acquérir Excelerator, réagissent négativement. Certains se désintéressent de l'outil, d'autres se montrent très perplexes quant à son utilité. L'outil accompagné de brochures explicatives est mis à leur disposition et chaque développeur tente de déterminer en quoi l'outil va pouvoir l'aider.

Pour résoudre ce problème, le groupe de support doit adopter une attitude plus commerciale pour réaliser la promotion de l'outil. L'objectif à atteindre n'est pas simplement de mettre l'outil à la disposition du développeur mais bien de **donner envie** à ce dernier d'utiliser l'outil. Pour atteindre cet objectif, nous proposons ci-dessous une série de recommandations à suivre :

- Les démonstrations de Excelerator doivent se centraliser davantage sur les besoins des utilisateurs que sur l'outil lui-même car une simple présentation des fonctionnalités offertes par l'outil ne permet pas au développeur de déduire immédiatement les gains de temps et des améliorations qualitatives qui résulteront de son utilisation.
- Excelerator n'est pas un outil isolé et est destiné à faire partie à long terme d'un ensemble intégré d'outils qui couvriront l'entièreté du développement. Cet objectif à long terme est trop souvent considéré comme un rêve, une utopie par les développeurs. Il est donc nécessaire de les informer davantage et plus concrètement pour les convaincre que cela deviendra tôt ou tard une réalité. Cet objectif d'environnement de développement automatisé et intégré fera l'objet du chapitre suivant.
- L'utilisation de Excelerator doit être située par rapport aux activités de la méthode SDM et tout type de document réalisable doit être justifié du point de vue de son utilité et de ses destinataires : le temps des développeurs est trop précieux pour qu'ils décident de documenter avec Excelerator uniquement pour la beauté du résultat.

2. REALISER UNE INTEGRATION SIMPLE ET COMPLETE ENTRE EXCELERATOR ET DATAMANAGER

La documentation du système développé doit être centralisée dans le dictionnaire Datamanager. Aussi, les modèles conceptuels et les spécifications fonctionnelles réalisées à l'aide de Excelerator doivent également se trouver dans Datamanager. L'intégration entre les deux outils qui est prise en charge par le groupe de support au développement, est terminée pour la partie "données" et est en projet pour la partie "traitements". Tant que cette intégration ne sera pas complète, il subsistera des introductions multiples d'informations qui serviront d'argument de poids à la sous-utilisation de Excelerator. En ce qui concerne le transfert de la partie "données" désormais possible, les développeurs se plaignent de la lourdeur de la procédure et espèrent une simplification à long terme. Si cela n'est pas possible ou non souhaité, il est très important de les en avertir.

3. ENVISAGER LES POSSIBILITES D'UN NOUVEL OUTIL DE PLANNING

Il ressort de l'enquête que Superproject est inadapté aux besoins des développeurs. Compte-tenu de l'importance qu'ils accordent à la réalisation de plannings fiables nous pensons que l'acquisition d'un nouvel outil de planning est à envisager. Nous vous présentons sur base des informations recueillies, les principales caractéristiques que devrait avoir un bon outil de planning :

- **simple et efficace** : il doit être simple à apprendre et à utiliser et doit avoir un rendement maximum (niveau de précision des résultats) pour un investissement minimum (volume et niveau de détail des données à introduire). Sans cette dernière condition, il est très difficile aux chefs de projet de réaliser des plannings fiables car ils estiment ne pas disposer de suffisamment de temps pour introduire toute l'information nécessaire.
- **multiprojet** : il n'est pas rare que certains chefs de projet assurent la gestion simultanée de plusieurs projets et cela doit pouvoir se faire avec un seul outil, ce qui n'est pas le cas actuellement.
- **gestionnaire de ressources** : il doit permettre de réaliser la répartition des tâches entre les différents membres de l'équipe et être compatible avec le système de facturation existant afin que les informations concernant l'emploi du temps des développeurs ne soient introduites qu'une seule fois.
- **intégré au dictionnaire Datamanager** : plus les informations fournies au départ sont précises, plus les estimations seront fiables. Or, une partie de ces informations sont contenues dans Datamanager : la description des différentes tâches du système à développer. Il est donc souhaitable que l'outil de planning soit intégré à Datamanager afin d'éviter l'introduction multiple des mêmes informations.

CHAPITRE 4

**Les Case Tools comme
générateurs de documentation**

4.1 L'AD/CYCLE: L'INFORMATISATION DE L'INFORMATIQUE.

Nous avons vu que comme la plupart des entreprises, la CGER cherche depuis plus de dix ans à simplifier, rendre plus efficace et plus productif le processus de développement de systèmes informatiques. Cette recherche est axée sur les technologies et outils CASE (Computer-Aided Software Engineering) dont le nombre ne cesse d'augmenter sur les marchés.

En 1989, IBM a proposé une solution, l'AD/cycle (Application Development Cycle) dont la structure correspond parfaitement à la solution qui a été choisie à la CGER. L'objectif de l'AD/cycle est de promouvoir au maximum l'automatisation du développement.

4.1.1 Définition de l'AD/CYCLE ⁽¹⁾

L' AD/cycle peut être défini comme un cadre destiné à contenir un ensemble de Case Tools qui couvriront l'entièreté du cycle de développement d'une application informatique, en transformant la définition des exigences et des besoins en une application informatique sûre, efficace et appropriée, tout en consommant un minimum de ressources.

Une présentation plus précise de la technologie de l'AD/cycle est proposée en annexe 8. La figure ci-dessous représente le cadre AD/cycle :

⁽¹⁾ Cette définition est tirée de l'ouvrage "L'AD/cycle", réalisé à la CGER par Christian GRAAS (13), informaticien.

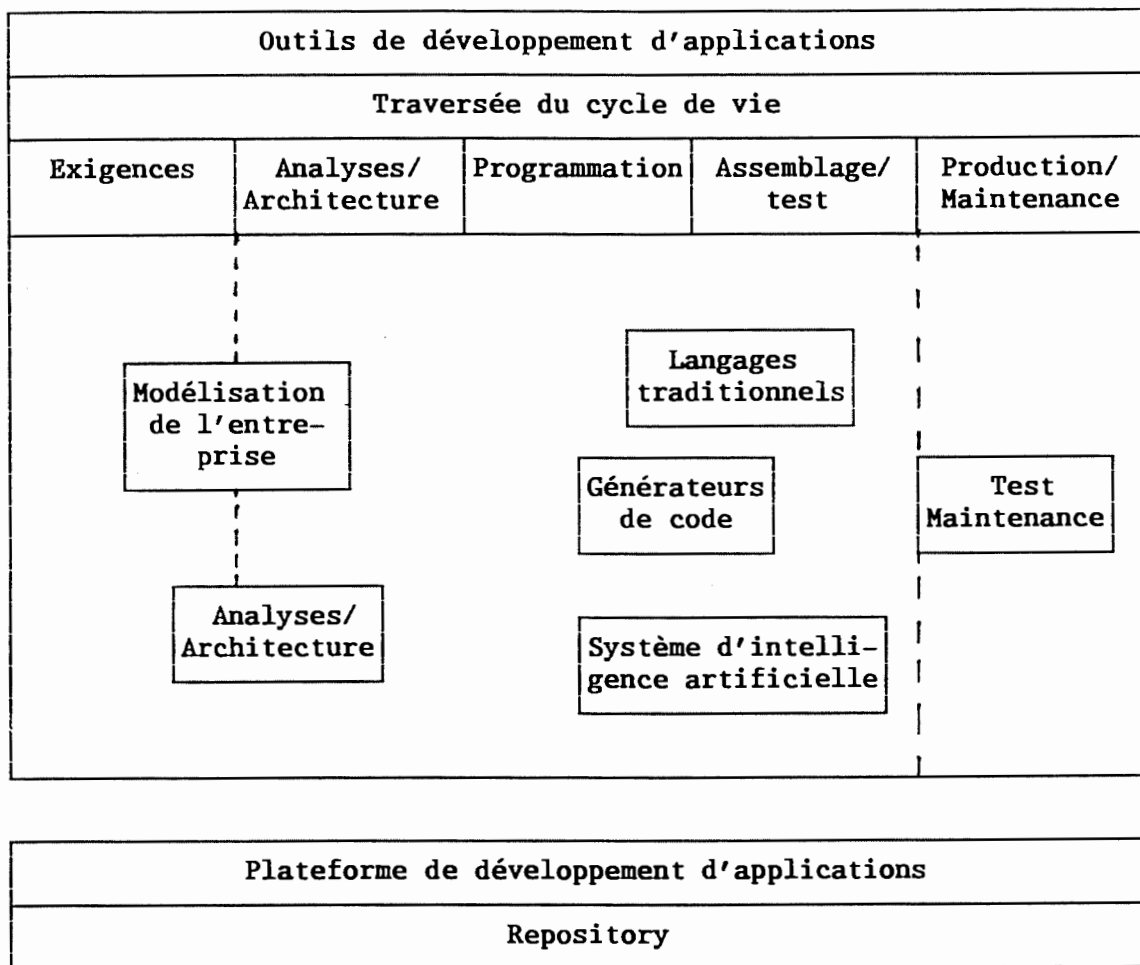


Figure 9 : Schéma général d'AD/cycle

Comme le montre le schéma ci-dessus, AD/cycle se décompose en deux parties principales :

- **les outils de développement** qui prennent en charge une ou deux phases du cycle de vie. Il est très important de constater qu'aucun des outils ne prend en charge les activités de gestion de projet telles la planification, les estimations des coûts ou le suivi du développement. Etant donné l'importance de cette activité il s'agit là d'une importante lacune.
- **la plateforme de développement** qui permet d'intégrer les différents outils. Elle est constituée du Repository ainsi que les interfaces nécessaires (interfaces standards entre les outils, interface unique pour l'utilisateur pour palier le problème des différents interfaces propres à chaque outil).

Le **Repository** est un élément fondamental de l'AD/cycle: il s'agit d'un dictionnaire de données destiné à contenir toutes les méta-informations stratégiques de l'entreprise. De cette manière, il constitue le point central de contrôle pour toutes les étapes de développement. Cependant, nous avons pu constater lors de la lecture d'ouvrages consacrés à l'AD/cycle que la définition du Repository est encore trop vague pour pouvoir en déduire très précisément les fonctionnalités.

4.1.2 Impacts au niveau de la documentation

Nous pouvons affirmer que l'AD/cycle représente une véritable révolution au niveau de la documentation :

Actuellement, les développeurs mettent l'accent sur la documentation technique du système dans le but de préparer au mieux la phase de programmation.

Quant à AD/cycle, il concentre la majeure partie de la documentation au niveau de la spécification des besoins et des phases d'analyse conceptuelle et fonctionnelle de ces besoins tandis que l'automatisation de la phase de programmation permet de décharger le développeur des détails techniques.

En ce qui concerne la documentation de chacune des cinq phases du cycle de vie, elle sera créée et générée par les outils de support respectifs. Les informations produites par un outil et qui constitueront un input pour l'outil de la phase suivante seront échangées entre les deux outils via le Repository afin d'assurer la centralisation et la cohérence des informations pour tout l'ensemble. De cette manière, AD/cycle assure une gestion et un partage centralisés des informations propres au développement d'une application.

4.2 AD/CYCLE A LA CGER

La création d'un environnement automatisé et intégré de développement d'applications sur mainframe dans un environnement IBM (IMS ou DB2), représente le défi à long terme du groupe de support au développement de la CGER et a été envisagée dès 1980, lorsque Datamanager fut acheté.

Dans ce second point, nous allons examiner la situation actuelle ainsi que les perspectives d'avenir concernant la réalisation de l'AD/CYCLE à la CGER d'une part et d'autre part, l'impact de cette réalisation sur la documentation.

4.2.1 Situation actuelle

a. Réalisation de l'AD/cycle

Au niveau de la plateforme de développement, la CGER dispose du dictionnaire Datamanager qui assure actuellement la fonction de Repository. De plus, la méthode SDM présente une découpe en phases du développement totalement

compatible avec celle de l'AD/cycle.

Au niveau des outils, le groupe de support a fixé une politique de choix qui consiste à retenir le meilleur outil dans chacune des phases de l'AD/cycle.

Actuellement, la CGER dispose déjà de quatre outils :

- Excelerator (de la firme Index Technology) acquis en 1989 et qui supporte la phase d'analyse et d'architecture.
- APS (développé par SAGE) acquis en 1991 et qui constitue un générateur de code Cobol. Il est actuellement en phase de test et est destiné à supporter la phase de programmation. L'avantage d'un générateur de code est qu'il exige des spécifications moins détaillées qu'un langage procédural. Il permet à des programmeurs moins qualifiés de produire des applications de qualité.
- Cobol Interactive Debugger (IBM) acquis en 1990 et qui permet de tester et de corriger les programmes Cobol.
- Les outils XPERT (développés par ESC) acquis en 1990 et permettant de créer des fichiers de données de test par extraction au niveau des données existantes.

Au niveau de l'intégration, la CGER se base sur les piliers suivants :

1. Pont Excelerator <-> Datamanager pour la consolidation des modèles de données (depuis 1990).
2. Pont Excelerator <-> Datamanager pour la consolidation des traitements (à développer 1992-1993).
3. Générateur d'application APS (mise en oeuvre prévue en 1992).
4. Outil de test : générateur de données de tests XPERT.

Un élément qui fait encore défaut actuellement concerne l'enchaînement des différents outils sur une base plus automatique.

Entre Excelerator et APS, il n'existe aucune intégration actuellement. Cependant, les constructeurs de ces outils se sont récemment associés pour entreprendre cette intégration. Toutefois, l'intégration qu'ils envisagent est en contradiction avec l'AD/cycle en ce qui concerne le **partage centralisé** des données entre les outils. En effet, le projet d'intégration dont il est question prévoit des échanges **directs** d'informations entre les deux outils, sans devoir passer par un dictionnaire central (Repository).

b. La documentation

Les résultats de l'enquête présentés au chapitre précédent ont montré que la majeure partie de la documentation du projet est réalisée dans le dictionnaire central.

D'autre part, ils révèlent que l'outil Excelerator est sous-utilisé : les développeurs exploitent principalement ses possibilités graphiques et négligent les possibilités de spécification et de validation des modèles des données et des traitements.

En observant la situation actuelle, on remarque que l'utilisation de l'outil APS pour la génération des applications suit directement celle de Excelerator. Cela signifie qu'une partie de la documentation du projet fournie par Excelerator est susceptible d'être utilisée par APS.

Ce transfert d'information entre les deux outils devrait dès maintenant constituer un argument de taille en faveur de l'utilisation de Excelerator. Et cet argument devrait, selon nous, être employé par le groupe de support pour promouvoir Excelerator. Le fait de pouvoir dire aux développeurs que s'ils réalisent un certain type de documentation à l'aide de Excelerator, celui-ci pourra être directement utilisé par le générateur d'application leur apportera une motivation supplémentaire incontestable.

Et même si ce transfert n'est pas encore possible actuellement, il peut être envisagé à relativement court terme dès que l'intégration entre Excelerator et Datamanager sera terminée et que l'outil APS sera suffisamment connu pour être également intégré à Datamanager car, pour rester conforme à l'AD/cycle, le transfert devra se faire via Datamanager.

4.2.2 Perspectives d'avenir

a. Réalisation de l'AD/cycle

L'objectif principal à moyen terme du groupe de support est de réaliser une intégration complète et efficace des outils existants au niveau du dictionnaire Datamanager.

Parallèlement à cela, il s'agit pour eux de suivre de près l'évolution des définitions du Repository proposées par IBM afin d'être prêt à entamer l'adaptation de Datamanager dès que les modèles du Repository seront suffisamment définis.

Finalement, le groupe de support désirerait compléter l'ensemble des outils par les acquisitions suivantes :

- Un **générateur de scénarios de test** qui viendrait compléter le générateur de données de test XPERT. Il n'existe actuellement aucune offre intéressante d'un tel outil sur le marché.

- Un outil de "reverse engineering" permettant de prendre en charge la maintenance des applications.

L'outil actuellement envisagé est BACHMAN/DBA dont nous avons déjà parlé au chapitre précédent et qui offre des possibilités de "reverse engineering" très intéressantes au niveau des bases de données. Cet outil fait actuellement l'objet d'une étude approfondie de la part du groupe de support au développement et du groupe chargé de l'administration des bases de données. Un problème majeur de cet outil est qu'au niveau du "forward engineering" une importante partie des fonctionnalités qu'il offre est redondante avec celles de Excelerator, ce qui n'est pas à négliger compte-tenu du coût d'acquisition d'un tel outil.

b. Documentation

Dans le cadre de l'AD/cycle, nous avons vu que la documentation du système partagée par les différents outils est centralisée au niveau du Repository. De cette manière, le Repository constituera l'unique source d'information concernant les développements d'applications, ce qui simplifiera considérablement la gestion de cette information et qui améliorera sa qualité. En effet, le partage de l'information via le Repository réduira la propagation d'informations redondantes, incomplètes ou incohérentes et permettra un contrôle accru de l'information. De plus, lors de projet de maintenance il sera plus aisé de prévoir les impacts des modifications sur l'information stockée dans le dictionnaire.

Quant à la documentation détaillée des résultats de chacune des phases du développement, elle est standardisée pour chaque phase et générée par l'outil de support de cette phase.

Ces perspectives en matière de documentation sont jugées **insatisfaisantes** par les développeurs qui estiment que les générateurs de rapports de documentation qui accompagnent les outils sont **inadéquats** :

Ils génèrent des rapports qui ne sont bien souvent que de simples impressions du contenu du dictionnaire local de l'outil.

Le grand souhait des développeurs, c'est de disposer d'un véritable **outil "bureautique" global**, qui supporterait toutes les phases du développement et qui serait capable de produire de **véritables** dossiers de documentation et non seulement des notes et schémas divers. Ces dossiers contiendraient une table des matières, des titres et des index qui en assureraient la clareté et la cohérence.

Nous pensons qu'une telle proposition constitue un domaine de réflexion très intéressant car il nous paraît tout à fait possible d'envisager un tel outil dans le cadre de l'AD/Cycle et plus particulièrement au niveau de la plateforme. Ainsi présenté, cet outil constituerait un **pilote** pour le développeur en ce qui concerne la réalisation de la documentation.

CONCLUSIONS

La standardisation et l'automatisation du processus de développement de systèmes d'information sont bien présentes à la CGER.

De même, la réalisation de la documentation des systèmes développés est en voie de subir le même sort. L'évolution d'une documentation manuelle et personnalisée vers une documentation automatisée et standardisée se réalise progressivement et semble devenir inévitable.

L'évolution est tout d'abord possible par la détermination d'un modèle de documentation de projet standard et généralisé par des outils d'aide au développement capables de supporter ce modèle.

Le stade ultérieur consistera en un outil de support totalement intégré et couvrant la totalité du cycle de vie d'un projet. Quoique cet outil n'existe pas encore et soit qualifié d'utopique par bon nombre de développeurs, nous avons toutes les raisons de croire qu'il constituera le quotidien de leurs successeurs.

Le département informatique de la CGER subit actuellement cette évolution dont les promoteurs sont la direction et les équipes de support au développement "Méthodes" et "Outils".

Nous avons vu que la méthode SDM telle qu'elle ne propose pas de standardisation de la documentation suffisante. Nous avons donc étudié et critiqué cette méthode afin d'en dégager les composants principaux concernant la réalisation complète d'une documentation de projet de développement.

Nous avons ensuite défini et commenté la documentation réalisée à la CGER afin de fixer l'état actuel de la documentation des projets en cours de développement.

Sur base des besoins actuels en documentation, nous avons mis en évidence les souhaits d'automatisation exprimés par les développeurs et qui constitueront pour le groupe de support au développement des objectifs prioritaires.

Finalement, nous avons étudié en quoi consiste le stade final de standardisation et d'automatisation des développements de la CGER tel qu'il est envisagé par le groupe de support au développement et nous avons vu qu'il ne correspond pas totalement au souhait des développeurs.

La réalisation de ce mémoire nous a permis de mettre en évidence un problème de communication des informations dû à la structure fortement spécialisée horizontalement du département informatique de la CGER.

Les équipes de développement bénéficient d'une liberté d'action très importante de la part de la direction.

Principalement confrontées à des problèmes concrets et journaliers de respect des exigences et des délais imposés par les utilisateurs, les équipes de développement ne se montrent guère disponibles à se préoccuper des résultats des recherches de standardisation et

d'automatisation du groupe de support méthodologique .

Conscients de cette divergence des préoccupations, les membres du groupe de support n'ont pourtant pas suffisamment essayé de se rapprocher des problèmes des développeurs et se sont progressivement isolés. Actuellement, ils poursuivent leurs recherches tout en laissant aux développeurs le choix d'accepter ou non les fruits de ces recherches.

Nos premiers et principaux contacts avec la CGER se sont fait via le groupe de support au développement au sein duquel nous avons été placés. Nous avons consulté de nombreux documents réalisés par ce groupe et la vision que nous avions de la CGER reposait sur celle du groupe de support.

Lors de la réalisation des interviews et enquêtes, nous nous sommes rendus compte que la communication entre développeurs et groupe de support était difficile. Cependant, il était trop tard pour sortir de cette enceinte et considérer les résultats de ce mémoire d'un oeil vraiment externe et donc plus critique.

Nous sommes donc arrivés à la conclusion que le département informatique de la CGER souffre d'un problème de structure dont la résolution incombe à la direction car, c'est son rôle de veiller à la coopération harmonieuse des différents groupes de travail.

BIBLIOGRAPHIE

- (MARCELIS 1) : D. MARCELIS, *Implantation de méthodes de conception et de développement de SI, étude de cas*, mémoire (Institut d'Informatique), 1990.
- (ROLLAND 2) : C. ROLLAND, "Introduction à la conception des systèmes d'information et panorama des méthodes disponibles", *Revue Génie Logiciel*, n° 4, pp. 6-11, 1985.
- (MAC FARLAN 3) : J. MAC FARLAN, *Corporate information systems management : text and cases*, Richard IRWIN, 1988.
- (HEILERS 4) : H.B. HEILERS, *Systeem-ontwikkeling volgens SDM*, document CGER-DSI/M, 1979.
- (ROLLAND 5) : C. ROLLAND, *La conception des systèmes d'information : état de l'art et nouvelles perspectives*, document de la chaire Franqui, 1990.
- (BOEHM 6) : W.B. BOEHM, "A spiral model of software development and enhancement", *Computer Revue*, pp. 61-72, mai 1988.
- (BODART 7) : F. BODART, Y. PIGNEUR, *Conception assistée des systèmes d'information*, Masson, 1989.
- (HAINAUT 8) : J.L. HAINAUT, *Conception assistée des applications informatiques*, Masson, n°2, 1986.
- (GRAAS 9) : C. GRAAS, *Paloma2*, document CGER-DSI/M, 1989.
- (CLAESSENS 10) : M. CLAESSENS, *Case Tools, meer in het bijzonder : Excelerator*, document CGER DSI/M.
- (RETTIG 11) : M. RETTIG, "Nobody reads documentation", *Communications of the ACM review*, vol. 34, n° 7, pp. 19-21, juillet 1991.
- (BOEHM 12) : B.W. BOEHM, "Software engineering economics", *IEEE Transactions on Computers*, Prentice-Hall, 1981.
- (GRAAS 13) : C. GRAAS, *L'AD-Cycle et l'AD-Cycle à la CGER*, document CGER-DSI/M, 1991.

Facultés Universitaires Notre-Dame de la Paix, Namur - Institut d'Informatique
Année Académique 1991-1992

Etude et génération d'une documentation
minimale pour accompagner la méthode
d'analyse SDM, étude de cas

ANNEXES

Muriel DETAILLE

Mémoire présenté en vue de l'obtention du grade de licencié et maître en informatique.

TABLE DES MATIERES

Annexe 1	: Résumé de la méthode SDM.	7
Annexe 2	: Synthèse de la méthode SDM.	12
Annexe 3	: Questionnaire distribué aux informaticiens de la CGER.	45
Annexe 4	: Exemple simplifié d'application de IFPA.	51
Annexe 5	: Description du dictionnaire de données Datamanager.	61
Annexe 6	: Description de l'outil de planning Superproject.	78
Annexe 7	: Description de l'Upper Case Tool Excelsior.	87
Annexe 8	: Présentation de la technologie de l'AD/Cycle.	102

ANNEXE 1

Résumé de la méthode SDM

INDICATIONS DESTINEES A L'UTILISATION DE CE MANUEL

Le premier chapitre donne un résumé schématique des différentes phases et activités de la méthode SDM en précisant les spécificités de son application à la CGER.

Les diverses rubriques de ce schéma ont les significations suivantes :

1. **PHASES** : énumération, en français, des différentes phases et activités.
2. **REMARKS** : remarques préalables éventuelles.
3. **CONTACTS** : les personnes, groupes ou services que l'on peut - et que l'on doit parfois - contacter, suivant le type de projet.
4. **TOOLS** : les outils ou produits pouvant être mis en oeuvre pour l'activité correspondante.
5. **TECHNIQUES** : les techniques ou principes que l'on peut ou que l'on doit appliquer.
6. **DATA DICTIONARY** : l'utilisation d'un dictionnaire de données destiné au stockage d'informations et de documentation relatives au projet, de structures et de descriptions de données ainsi que de règles de production. Le "data dictionary" utilisé actuellement est "Data manager".
7. **SECURITY** : dispositifs, mesures ou règles qui contribuent à accroître la sécurité du système d'information.
8. **HARDWARE** : le cas échéant, nouvelle infrastructure ou nouveau matériel (informatique) nécessaire.
9. **SOFTWARE** : le cas échéant, nouveaux logiciels d'exploitation et/ou logiciels d'application.
10. **TEST** : planning, mesures et exécution en ce qui concerne le test de programmes et de (sous-)systèmes ainsi que le test global (test d'acceptation).
11. **CONVERSION** : planning, mesures et exécution en ce qui concerne la conversion de données vers la nouvelle structure.
12. **FORM** : l'utilisation de formulaires appropriés.
13. **IMPLIED PERSONS** : les personnes qui sont spécialement associées aux phases et aux activités respectives.
14. **COURSES** : les cours recommandés.
15. **ORGANISATION** : les aspects humains dans un système d'information ainsi que les modifications possibles de l'environnement.

16. **REPORT** : la rédaction d'un rapport sur lequel la direction et les demandeurs de projet se baseront pour prendre des décisions concernant le développement ultérieur du projet.
17. **PHASES** : énumération, en néerlandais, des différentes phases et activités.

Le deuxième chapitre donne un résumé descriptif de toutes les activités de la méthode SDM.

Pour chaque activité, on développe au moins les deux points suivants :

- une introduction comportant des remarques préalables
- une méthode possible pour effectuer cette activité.

Ces deux points se basent sur l'ouvrage de H.B. EILERS relatif à la méthode SDM ("Systeemontwikkeling volgens SDM")

- édition : Academic Service - La Haye
- PSBN 90 6233 1777).

En outre, vous trouverez pour presque chaque activité un point où figurent des indications pratiques concernant son application à la CGER. Il y a lieu de considérer ces indications comme une notice explicative du résumé schématique (voir chapitre I).

Le troisième chapitre comporte un certain nombre de formulaires qu'il faut utiliser au cours du développement de systèmes d'information à la CGER et auxquels il est fait référence dans le résumé schématique (voir chapitre 2).

Le quatrième chapitre contient enfin une liste explicative d'abréviations et de termes utilisés dans les chapitres précédents.

PHASES S.D.M.	REMARKS	CONTACTS			TOOLS PRO- DUCTS	TECH- NIQUES	DICT. (D.M.R.) CASE	S E C.	H. W.	S. W.	T E S T	C O N V.	FORM.	CON- CERNED PER- SONS	COURSE	O R G.	R E P.	S.D.M. FAZEN
		PERSONS	TEL.	GROUP														
0. ETUDE PRELIMINAIRE		C. GRAAS	7037	DSI-M									Develop. Request	Project Leader/ Promo- tor			X	0. VOORSTUDIE
		Documentalist											SMI					
		B. BREMS	6292	PREP.									Applic. Code Assign. Form					
		J. MATTHIJS	7247	AUDIT														
		J-F DUCHENNE	7342	Costs- centre														
1. ETUDE DE L'EXISTANT ET ANALYSE CONCEPTUELLE																		1. STUDIE VAN DE BEST/ TOESTAND EN CONCE- TUELE ANALYSE
1.1 Définir le problème et le champ de l'étude																		1.1 Het probleem en de o- van de studie bepalen
1.2 Rassembler les données sur la situation existante						Interv. Tech. D.F.D. H.I.P.O. T.U.D. Dec. Tables H.S.A.T.		X							SPEC-AN MCM			1.2 De gegevens over de bestaande toestand verzamenen
1.3 Analyser et évaluer les données						Interv. Tech. D.F.D. H.I.P.O. T.U.D. Dec. Tables H.S.A.T.		X							SPEC-AN MCM			1.3 Die gegevens analyser evalueren
1.4 Déterminer les objectifs et exigences du nouveau système							SY-	X										1.4 De doelstellingen en de vereisten van het nieuw systeem bepalen
1.5 Arrêter les points qui restent à résoudre et les hypothèses de base									X	X						X		1.5 De nog op te lossen p- en de gemaakte veron- stellingen vastleggen
1.6 Faire un schéma du système		C. GRAAS	7037	DSI-M		CDM/CPM H.I.P.O. Dec. Tables	SY- AC- TK- OB- AG- EL-								S E R V I C E S	MCM		1.6 Een systeemschets ma-
1.7 Déterminer les outils et les solutions possibles		R. MATHAR	6267	RSI/ IOTA					X	X						X		1.7 De instrumenten en i- lijke oplossingen bepa-
		D. ABRASSART	6128	PTS														
1.8 Evaluer les solutions et opérer une sélection		J. MATTHIJS	7247	AUDIT												(X)	X	1.8 De oplossingen evaluer een selectie doorvoeren
1.9 Déterminer les problèmes de conversion et de mise en production; déterminer les critères d'acceptation		R. MATHAR	6267	RSI/ IOTA					X	X	X	X				X		1.9 De problemen met be- king tot de conversie de invoering vaststelle de acceptatiecriteria v- leggen
	Operability Requirements	B. BREMS	6292	PREP.														
		J-P REMORY	8386	SYS- TEMS														
		J. VAN DE WATER	8715	PROD.														
1.10 Elaborer un planning global et une vue d'ensemble des					Super- Project													1.10 Een globale planning opstellen en con anal

↑

EXCEL

FUDOS

↑

INFORMATION

PILASES S.D.M.	REMARKS	CONTACTS			TOOLS PRO- DUCTS	TECH- NIQUES	DICT. (D.M.R.) CASE	S E C.	H. W.	S. W.	T E S T	C O N V.	FORM.	CON- CERNED PER- SONS	COURSE	O R G.	R E P.	S.D.M. FAZEN
		PERSONS	TEL.	GROUP														
2. ANALYSE FONCTIONNELLE																		2. FUNCTIONELE ANALY
2.1 Spécifier les exigences du système, y compris futures		R. MATHAR	6267	RSI/ IOTA			SY- AC- TK- OB- AG- EL-	X	X	X								2.1 De systeemeisen speci- inclusief de toekomst
2.2 Déterminer le cadre dans lequel le nouveau système devra fonctionner		L. VAN DAMME	7337	RSI/ IOTA				X	X	X						X		2.2 Het kader vastleggen waarbinnen het nieuw systeem moet functione
		J-P REMORY	8386	SYS- TEMS														
		J. VAN DE WATER	8715	PROD.														
		B. HANZEN	8978	PC														
		E. HOEDEMAKERS	8555	PC														
		D. ABRASSART	6128	PTS														
2.3 Diviser le système en sous-systèmes et les décrire						D.F.D. H.L.P.O. H.S.A.T.	SY- AC- TK-	X										2.3 Het systeem in subsyste- opdelen en deze beschrijf
2.4 Définir l'input et l'output par sous-système et les interfaces					H.L.P.O. Proto- typing	OB-												2.4 Per subsysteem, de inp- de output bepalen, als interfaces
2.5 Elaborer les diagrammes et les descriptions des traitements						D.F.D. H.L.P.O. Dec. Tables	TK- AG- EL-											2.5 De processchema's en /6 procesbeschrijvingen m
2.7 Spécifier les exigences en sécurité et confidentialité		EDI-Security (D. HARDY)	8503	SYS- TEMS				X								X		2.7 De eisen op het gebied de beveiliging en de ver- wijking specificeren
2.8 Déterminer les problèmes humains et les solutions																X		2.8 De menselijke problem- vaststellen en oplossen
2.9 Concevoir la structure logique des données		C. GRAAS	7037	DSI-M	PALOMA	Normalisation	OB- AC- EL-									MCM		2.9 De logische gegevens- structuur ontwerpen
	Systemchoice: IMS, DB2, and files.	DBA's	7348 8860	IMS/DB2												IMS		
	Turbo Image (HIP), AS/400	C. DE WIT	7055	MINIS														
2.10 Spécifier les facilités requises /12 en communication de données, en hardware et en software	Production Standards Naming Con- ventions	B. BREMS	6292	PREP				X								X		2.10 De vereiste faciliteiten r /12 betrekking tot data-com- municatie, apparatuur en r grammatuur specificeren
		D. ABRASSART	6128	PTS														
	Design Rev. Walk Through	R. MATHAR	6267	AKA/IA														
		C. DE WIT	7055	MINIS														
		J-P MELEBECK	6903	PROD.	PELIKAN (File Tr.)													
2.13 Elaborer un plan pour la poursuite du développement et la mise en production	START DATUM	B. BREMS	6292	PREP.	Super- Project											X		2.13 Een plan opstellen voor verdere ontwikkeling en invulling
		C. DESIRON	8874	TOOLS	TIMS/ HELP													
	HIP	C. DE WIT	7055	MINIS														
2.14 Rédaction du rapport		M. COPPIETERS	6220	neg. itrol	A.S.F.												X	2.14 Een rapport samenstellen

F U D O S

I N F O R M A T I C I A N

U S E R S

[illegible]

[illegible]

PHASES S.D.M.	REMARKS	CONTACTS			TOOLS PRO- DUCTS	TECH- NIQUES	DICT. (D.M.R.) CASE	S E C.	H. W.	S. W.	T E S T	C O N V.	FORM.	CON- CERNED PER- SONS	COURSE	O R G.	R E P.	S.D.M. FAZEN
		PERSONS	TEL.	GROUP														
5. TESTS														^				5. TESTEN
5.1 Elaboration d'un plan de test détaillé					Super- Project			X			X							5.1 Uitwerking van een gedetailleerd testplan
5.2 Installation du matériel, du software et préparation de l'environnement de travail		P. HERMAN	6452	Techn. Inst.					X	X						X		5.2 Installatie van de apparaten van de software; voorbereiding van de werk omgeving
	HP AS/400	C. DE WIT	7055	MINIS														
	PC's	E. HOEDEMAKERS	8555	PC														
5.3 Définition des unités de traitement et de leur ordre de succession																		5.3 Bepaling van de verwerkeenheden en van de omlingsvolgorde ervan
5.4 Test des formations, utilitaires et procédures	J.P. MILLECAM		Forma- tion													X		5.4 Testen van de opleidingshulpmiddelen en de procedures
				Vorming														
5.5 Constitution des données de test de système et d'acceptation						Dec. Tables		X			X					SPEC-AN		5.5 Samenstelling van systeem en acceptatietestgegevens
		L. VERVAET	9118	TOOLS	(XPERT)													
5.6 Test des sous-systèmes		R. MATHAR	6267	RSI/ IOTA				X			X							5.6 Testen van de subsysteemen
		L. VERVAET	9118	TOOLS	(XPERT)													
5.7 Exécution du test d'acceptation	Q. A.	DBA's	7348 8860	IMS		Dec. Tables		X	X		X							5.7 Uitvoeren van de acceptatietest
	Q. A.	F. DE BRABANTER	7751	DB2														
	Q. A.	E. VAN DER SYPE	6684	SYS- TEMS	T.P.N.S. SMF RMF													
	Capacity Management & Planning	J-P REMORY	8386	SYS- TEMS	NETVIEW IMF IMS-LOG DB2 Trace Facility GTF Trace MICS LOOK													
	Special Software	R. MATHAR	6267	RSI/ IOTA														
		C. DE WIT	7055	MINIS	LASER RX (HP)													
		E. HOEDEMAKERS	8555	PC														
		L. VERVAET	9118	TOOLS	(XPERT)													
5.8 Rédaction du rapport 'Résultats de test'					A.S.F.												X	5.8 Opstellen van het rapport 'testresultaten'

INFORMATION
 USERS

[illegible]

PHASES S.D.M.	REMARKS	CONTACTS			TOOLS PRO- DUCTS	TECH- NIQUES	DICT. (D.M.R.) CASE	S E C.	H. W.	S. W.	T E S T	C O N V.	FORM.	CON- CERNED PER- SONS	COURSE	O R G.	R E P.	S.D.M. FAZEN
		PERSONS	TEL.	GROUP														
7. GESTION ET MAINTENANCE DU SYSTEME							^							^				7. ONDERHOUD EN BEHEER VAN HET SYSTEEM
7.1 Elaboration et utilisation du système de signalement des erreurs		B. BREMS	6292	PREP.	BETA91													7.1 Bouwen en in gebruik nemen van een systeem fouten signaleert
7.2 Planification périodique de maintenance		H. MARIEN	7348	IMS														7.2 Plannen van de periodieke onderhoud
		P. FROIDCOEUR	7368	PROD.														
		F. DE BRABANTER	7751	DB2														
		P. DEVIGNE	7368	PROD.														
		C. DE WIT	7055	MINIS	BACK UP 3000													
7.3 Planification du traitement informatique		B. BREMS	6292	PREP.	CA-7 CA-11													7.3 Plannen van de computerverwerkingen
7.4 Prévention et restauration des fautes et perturbations		P. DECLERCQ	8875	TOOLS	ADF													7.4 Voorkomen en herstell van fouten en storingen
		M. DRUEZ	7759	TOOLS	DMR													
		T. COURARD	8857	TOOLS	SPITAB													
		H. MARIEN	7348	IMS														
		P. FROIDCOEUR	7368	PROD.														
		F. DE BRABANTER	7751	DB2														
		P. DEVIGNE	7368	PROD.														
		Dispatching	8833(F) 8822(N)	RSI/ IOTA														
		W. DOOREMONT	7197	RSI/ IOTA														
		P. HERMAN	6452	Techn. Inst.	NETMAN													
		C. DE WIT	7055	MINIS														
		E. VAN DER SYPE	6684	SYS- TEMS	NETVIEW IMF													
		J-P REMORY	8386	SYS- TEMS	IMS-LOG GTF DB2 Trace Facility MICS Trace RMF LOOK													
7.5 Surveillance des dispositions de sécurité	Disaster Recovery	EDP-Security (D. HARDY)	8503	SYS- TEMS	R.A.C.F			X										7.5 Toezicht op de beveiligingsvoorziening
		DBA's	7348 8860	IMS														
		F. DE BRABANTER	7751	DB2	D.B.M.S. D.B.R.C.													
7.6 Modifications et mise à jour de la documentation. Réponse aux besoins en information ad-hoc		R. PARIZEL	6264	INFO- CENTRE	QUERY Lang. ICI	Documentation Standards									QUERY Lang.			7.6 Wijzingen en bijwerken van de documentatie. Voorzien in de ad-hoc informatiebehoeften
7.7 S'occuper des formations complémentaires																X		7.7 Verzorgen van de aanvullende opleidingen
7.8 Gestion des données et des fichiers		DBA's	7348 8860	IMS	LOAD LIN.													7.8 Beheren van de gegevensbanken en de

ANNEXE 2

Synthèse de la méthode SDM

SYNTHESE DE LA METHODE SDM

Lefèbvre Christian
Philippot Jean Noël
Claessens Martine
Graas Christian

DSI-M
version 1.0, septembre '89

Table des Matières

INTRODUCTION

PARTIE 1 : SYNTHESE DE LA METHODE SDM

1	INTRODUCTION	
2	PHASE 1 : ETUDE DE L'EXISTANT ET ANALYSE CONCEPTUELLE	
2.1	description	
2.2	planning possible des activités	
2.3	description des activités et justification du planning	
2.4	activités importantes	
3	PHASE 2 : ANALYSE FONCTIONNELLE	
3.1	description	
3.2	planning possible des activités	
3.3	description des activités et justification du planning	
3.4	activités importantes	
4	PHASE 3 : ANALYSE TECHNIQUE	
4.1	description	
4.2	planning possible des activités	
4.3	description des activités et justification du planning	
4.4	activités importantes	
5	PHASE 4 : REALISATION DES PROGRAMMES	
5.1	description	
5.2	planning possible des activités	
5.3	description des activités et justification du planning	
5.4	activités importantes	
6	PHASE 5 : TESTS	
6.1	description	
6.2	planning possible des activités	
6.3	description des activités et justification du planning	
6.4	activités importantes	
7	PHASE 6 : CONVERSION ET MISE EN PRODUCTION	
7.1	description	
7.2	planning possible des activités	
7.3	description des activités et justification du planning	
7.4	activités importantes	
8	PHASE 7 : UTILISATION ET GESTION DU SYSTEME	
8.1	description	
8.2	planning possible des activités	
8.3	description des activités et justification du planning	
8.4	activités importantes	
9	SYNTHESE	
9.1	pour les données	
9.2	pour les traitements	
9.3	pour la sécurité	

1 INTRODUCTION

Cette synthèse a pour but de reprendre les différents jalons de la méthode et d'essayer d'en retirer les plus importants.

La méthode S.D.M. traite du cycle de vie complet d'un système d'information. Elle s'occupe de l'analyse, de la réalisation et de la maintenance d'un projet. C'est un outil permettant de gérer, de manière uniforme, les projets se présentant dans une entreprise; de constituer la documentation d'un système; de développer et gérer correctement des systèmes d'information².

La méthode analyse le cycle de vie d'un système en le décomposant en phases et pour chacune d'elles en décrivant des activités. La méthodologie phasée permet de voir où retourner et quelles activités refaire en cas de modification.

La méthode est indépendante des techniques. En effet, S.D.M. décrit le QUOI faire et non le COMMENT faire. En complément, des outils, des modèles sont proposés pour supporter différentes activités car décrire uniquement le QUOI n'est pas suffisant.

Les activités citées constituent une "checklist" universelle. Cette liste étant universelle, certaines activités peuvent être omises pour certains projets. Toutefois, il est souhaitable d'en justifier la raison. Cette liste peut servir à la planification du projet.

La méthode offre une guidance pour des rapports réguliers qui offrent un moyen de contrôler la qualité, la progression, le coût, la planification d'un projet. Tout en développant le système, on constitue la documentation du projet. S.D.M. se veut un outil utile de communication pour la direction (points de décision, indications sur l'avancement du projet), les utilisateurs (vérification de conformité, d'avancement) et les collaborateurs d'un projet.

La méthode se veut orientée vers les utilisateurs. De plus, pour le bon déroulement du projet, une personne unique devrait servir d'intermédiaire entre les utilisateurs et les concepteurs. Cette personne aura une connaissance approfondie de l'organisation pour être reconnu par les utilisateurs comme un interlocuteur fiable et une connaissance étendue de l'automatisable (possibilité et faisabilité). Dans la réalité, cette personne, étant difficile à trouver, est remplacée par un ensemble de personnes.

2. La méthode permet de développer tous les types de système d'information, mais elle est principalement orientée vers les systèmes d'information informatiques. En effet, de nombreuses activités concernant uniquement le matériel informatique. Pour les systèmes non-informatiques, on omet ces activités.

Dans S.D.M., les activités sont classées en 7 phases :

0. Etude préliminaire
1. Etude de l'existant et Analyse conceptuelle
2. Analyse fonctionnelle
3. Analyse technique
4. Réalisation des programmes
5. Tests
6. Conversion et Mise en production
7. Utilisation et Gestion du système

Remarques :

- La phase 0, l'étude préliminaire, ne fait pas partie intégrante de la méthode. Cette phase est souvent terminée lorsqu'un projet commence. Elle comprend les opérations préalables à la phase 1. Les activités de cette phase consistent à cerner les problèmes qui se présentent sur le plan de la production d'informations. Son but essentiel est de donner le feu vert pour le démarrage de la phase 1 lorsqu'un problème est apparu.
- Les phases 1 et 2 de S.D.M., c-à-d l'étude de l'existant, l'analyse conceptuelle et l'analyse fonctionnelle, sont réalisées pour le système dans son ensemble et s'occupent de la conception du système.
- Les phases 2 à 7 peuvent se dérouler pour le système entier ou par sous-systèmes et s'occupent de l'aspect technique. Il est possible de développer les sous-systèmes individuellement.
- Pour de petits projets, certaines phases sont combinées.

Le découpage en phases sert à déterminer les différentes étapes du cycle de vie d'un système et à diviser le développement du système en unités faciles à manier et à appréhender. (Remarque : certaines phases peuvent se chevaucher)

Cela rend possible :

- une diminution des risques de retour en arrière,
- un juste emploi des compétences du personnel prévu pour le projet informatique,
- une planification des phases,
- une prise de décision sur base du rapport fourni après chaque phase .

S.D.M. constitue donc un support pour la conduite de projet et pour la documentation. Cela est possible car :

- le contenu des activités est clair et évaluable;
- des standards et des normes peuvent être prévus;
- la documentation est intégrée à chaque activité.

2 PHASE 1 : ETUDE DE L'EXISTANT ET ANALYSE CONCEPTUELLE

2.1 description

Son but est de comprendre le domaine à étudier et d'en fixer les limites. Au cours de cette phase, la définition du problème est précisée, les processus existants sont étudiés.

Il convient :

- de fixer les objectifs justifiant la poursuite du développement,
- d'envisager différentes conceptions possibles du système,
- d'effectuer une analyse bénéfices/coûts globale complète.

La documentation résultante décrit les contraintes et les critères fonctionnels auxquels devra satisfaire le système. Elle sert de base pour la poursuite du développement. Ce rapport donne aux réalisateurs une mine de renseignements pour la suite du projet et la maintenance. Il permet aux décideurs de choisir en fonction de l'approche bénéfices/coûts. Il peut aussi servir de base pour un cahier des charges.

En résumé, on peut définir cette phase en 3 points :

1. Quel est le problème ?
 - a. La demande de développement pour cerner le problème.
 - b. L'étude de l'existant : recueil de renseignements sur l'organisation, la production, les informations et évaluation.
2. Quelle est la solution ?
 - a. Exigences et limites du système.
 - b. Schéma du système.
3. Comment concrétiser la solution ?
 - a. Choix d'une solution et Conséquences.

2.2 planning possible des activités

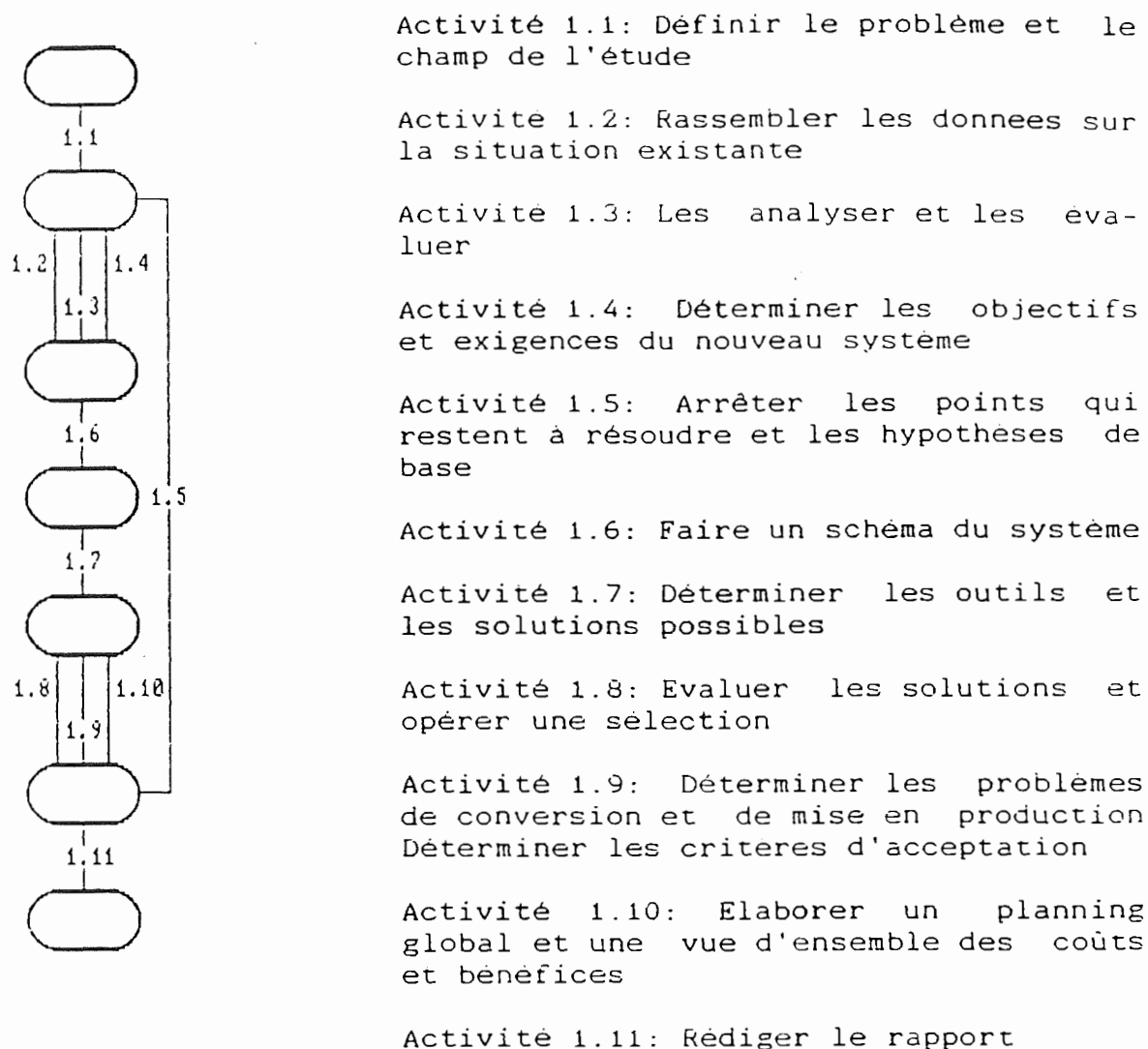


Figure 1: Planning de la phase 1

2.3 description des activités et justification du planning

Activité 1.1: Définir le problème et le champ de l'étude (objectifs, limites, exigences, ...) de manière globale. C'est ici que le groupe de travail est constitué.

Informations utiles pour son exécution : le rapport de l'étude préliminaire, les manuels, les directives, les normes, la politique sociale de l'entreprise, ...

Outil : un questionnaire.

Activité 1.2: Rassembler les données sur la situation existante. Grâce à des interviews et des études de documents, on obtient un inventaire global sur l'organisation, la production et l'information.

Informations utiles : 1.1 - la définition du projet; les rapports d'analyses précédentes; des schémas et descriptions du système actuel; les données se trouvant dans le dictionnaire.

Outils possibles : un questionnaire, des techniques d'interview, la statique des traitements (schémas IPÖ (input-process-output)), des diagrammes de flux, des tables de décision, un dictionnaire de données.

Activité 1.3: Les analyser et les évaluer. On a une vue d'ensemble sur le système existant ainsi que ses points forts, ses points faibles et des quantifications. On entrevoit ici aussi les problèmes de sécurité et de confidentialité.

Informations utiles : Cfr activité 1.2.

Outils possibles : Cfr activité 1.2.

Activité 1.4: Déterminer les objectifs et exigences du nouveau système, les critères de prestation (temps de réponse), les contraintes organisationnelles (formation) et d'intégrité (sécurité). Les exigences sont classifiées.

Informations utiles : 1.1 - la définition du problème, le champ de l'étude; 1.3 - l'analyse de l'existant, les souhaits et exigences, les suggestions pour améliorer le système; des informations techniques issues de la littérature scientifique ou de l'expérience d'une autre entreprise; ...

Activité 1.5: Arrêter les points qui restent à résoudre et les hypothèses de base, les moyens pour développer le système, les postulats.

Informations utiles : 1.3 - l'analyse de l'existant; 1.4 - les objectifs et exigences du système; les manuels internes; la politique sociale de l'entreprise; la législation; ...

Activité 1.6: Faire un schéma du nouveau système. On détermine, de manière globale, les entrées et les sorties, les données, les traitements principaux, les fonctions du nouveau système, les besoins en information.

Informations utiles : 1.2 - la situation existante; 1.3 - son analyse; 1.4 - les objectifs et exigences du système, les critères de prestation; 1.5 - les limites et hypothèses.

Outils possibles : la modélisation conceptuelle des données, la structuration et la statique des traitements (SHAT - IPO), les diagrammes de flux de données, le dictionnaire de données, la normalisation.

Activité 1.7: Déterminer les outils (estimation du hardware et du software nécessaires) et les solutions possibles avec avantages et inconvénients (Développement d'un prototype, package sur le marché, micro - mini) .

Informations utiles : 1.4 - les objectifs du S.I., les critères de prestation; 1.5 - les limites et utilitaires; 1.6 - les fonctions du système, la description des I/O; d'autres systèmes et expériences de développement; des packages d'application disponibles sur le marché, ...

Activité 1.8: Evaluer les solutions et opérer une sélection . On décrit une procédure d'évaluation en fonction des coûts, de la flexibilité, de la confidentialité, de l'ergonomie, de la continuité, des conséquences organisationnelles, ...

Informations utiles : 1.1 - les buts et limites de l'étude; 1.3 - l'analyse de l'existant; 1.4 - les objectifs et exigences du système; 1.5 - les limites, hypothèses et utilitaires; 1.6 - la description des I/O et les fonctions du nouveau système; 1.7 - les solutions alternatives.

Activité 1.9: Déterminer les problèmes de conversion et de mise en production (acquisition du hardware et du software supplémentaires, problèmes de formation). Déterminer les critères d'acceptation.

Informations utiles : 1.1 - la description du problème, les références; 1.3 - l'analyse de l'existant; 1.4 - les objectifs du système; 1.5 - la description des I/O, les limites, les utilitaires; 1.8 - la solution sélectionnée; les possibilités de formation.

Activité 1.10: Elaborer un planning global (pour toutes les phases) et une vue d'ensemble des coûts et bénéfices (retour possible en 1.8).

Informations utiles : les résultats des différentes activités; la politique budgétaire; l'analyse bénéfices/coûts.

Activité 1.11: Rédiger le rapport. (Remarque : il peut être rédigé en parallèle avec les autres activités).

Informations utiles : les résultats des activités.

Outil : un dictionnaire de données.

2.4 activités importantes

Remarque préliminaire : Le rapport produit après chaque phase n'est pas repris dans les activités importantes car il constitue une sorte de synthèse de la phase. Il est évident, cependant, que la rédaction de ces différents rapports est très importante. En effet, ces rapports servent de base pour la prise de décisions importantes (par exemple : la poursuite du développement, un retour en arrière, ...).

- L'analyse de l'existant et son évaluation.
- Le développement de l'analyse de base en fonction des limites et objectifs du système.
- L'évaluation et le choix d'une solution.

3 PHASE 2 : ANALYSE FONCTIONNELLE

3.1 description

Il s'agit d'une esquisse du système complet. Toutes les fonctions que le système devra remplir sont spécifiées. Ces spécifications consistent en une analyse fonctionnelle de l'ensemble du système, y compris le matériel et le logiciel nécessaires, jusqu'au niveau où les programmes et les procédures manuelles peuvent être conçues et où le matériel peut éventuellement être commandé.

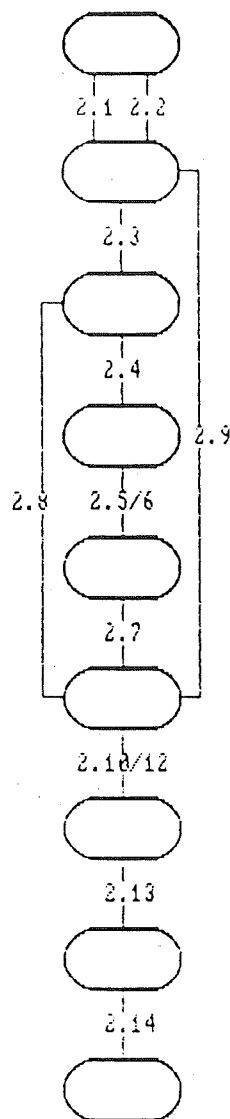
Tous les critères relatifs aux données à utiliser doivent être étudiés afin de définir la structure logique des données et de déterminer les "clés" par lesquelles on souhaite accéder à ces données. Les critères auxquels doivent répondre le logiciel et le matériel seront étudiés en détail et en corrélation avec ceux du système.

C'est au cours de cette phase que les sous-systèmes éventuels sont définis. L'analyse fonctionnelle peut se dérouler en deux passes. Lors de la première, le système est développé en gros jusqu'à un niveau uniforme pour garder la cohérence du système. Lors de la seconde, on développe en détail les sous-systèmes. Les interrelations entre sous-systèmes sont clairement définies. Une architecture de développement est élaborée.

Il peut s'écouler un grand laps de temps entre la phase d'analyse conceptuelle et celle-ci. Il peut se produire des événements modifiant la solution choisie. Il est donc possible de retravailler l'analyse conceptuelle lors des premières activités de l'analyse fonctionnelle.

A la fin de la phase, on a une conception claire et "définitive" du système global ainsi que les conditions auxquelles les sous-systèmes devront répondre. L'analyse fonctionnelle doit être détaillée suffisamment sans toutefois verser dans la technicité de sorte que l'utilisateur comprenne clairement les potentialités du système.

3.2 planning possible des activités



Activité 2.1: Spécifier les exigences du système y compris les exigences futures

Activité 2.2: Déterminer le cadre dans lequel le nouveau système devra fonctionner

Activité 2.3: Diviser le système en sous-systèmes et les décrire

Activité 2.4: Définir l'input et l'output par sous-système et les interfaces

Activité 2.5/6: Elaborer les diagrammes des traitements et les descriptions des traitements

Activité 2.7: Spécifier les exigences en sécurité et confidentialité

Activité 2.8: Déterminer les problèmes humains et des solutions

Activité 2.9: Concevoir la structure logique des données

Activité 2.10/12: Spécifier les facilités requises en communication de données, en hardware et en software

Activité 2.13: Elaborer un plan pour la poursuite du développement et la mise en production

Activité 2.14: Rédiger le rapport d'analyse fonctionnelle

Figure 2: Planning de la phase 2

3.3 description des activités et justification du planning

Activité 2.1: Spécifier les exigences du système, y compris les exigences futures (croissance du système) car les choses ont pu changer, ainsi que les critères de prestation.

Informations utiles : 1.4 - les plans et exigences des utilisateurs; 1.10 - le rapport de la 1^{re} phase et commentaires; les développements politique et économique; les standards et normes;

...

Activité 2.2: Déterminer le cadre dans lequel le nouveau système devra fonctionner (procédures de travail, environnement software et hardware, cadre organisationnel, conditions d'intégrité, ...).

Informations utiles : 1.4 - les objectifs et exigences du système; 1.8 - la solution sélectionnée; 2.1 - les exigences détaillées et futures .

Activité 2.3: Diviser le système en sous-systèmes et tâches et les décrire (spécifier les interfaces entre sous-systèmes). La subdivision en sous-systèmes de la phase conceptuelle est validée, éventuellement modifiée, et définitivement fixée.

Informations utiles : 1.3 - l'analyse de l'existant; 1.4 - les critères de prestation; 1.4 - 2.1 - les exigences du système; 1.6 - les fonctions du nouveau système; 2.2 - le cadre du nouveau système; 1.11 - le rapport de la 1^{re} phase.

Outils : la structuration et la statique des traitements (SHAT - schémas IPO), SADT (Structured Analysis and Design Technique), les diagrammes de flux.

Activité 2.4: Définir l'input et l'output par sous-système et par tâches et les interfaces (contenu des écrans et formulaires, dialogues, ...) et description des données.

Informations utiles : 1.2 - 1.3 - les informations sur l'existant; 1.6 - la description des I/O et des fonctions du nouveau système; 1.4 - 2.1 - les exigences du système, les besoins en information; 2.2 - le cadre du système; 2.3 - la division en sous-systèmes.

Outils : le prototyping, la statique des traitements, le dictionnaire de données.

Activité 2.5/6: Elaborer les diagrammes des traitements et les descriptions des activités et des tâches.

Informations utiles : 1.4 - 2.1 - les exigences du système; 1.8 - la solution sélectionnée; 2.2 - l'environnement du système; 2.3 - la division en sous-systèmes et leur description; 2.4 - la spécification des I/O.

Outils possibles : les diagrammes de flux de données, la statique et la dynamique des traitements, les tables de décision, le prototyping, un dictionnaire de données, SADT, ...

Activité 2.7: Spécifier les exigences en sécurité, confidentialité, contrôle, correction, ... (contre les personnes, le système; rétablir le système; accès aux données, aux transactions; manipulation de données, erreur de programmation, d'appareil, de transmission de données, du réseau).

Informations utiles : 1.3 - l'analyse de l'existant; 1.5 - les limites du système; 1.4 - 2.1 - les exigences du système; 1.8 - 2.2 - le cadre du nouveau système; 2.3 - la description des sous-systèmes; 2.5/6 - la description des traitements; la législation; les techniques de contrôle; les prescriptions incendie; le dictionnaire de donnée; ...

Outil : un gestionnaire de sécurité, un logiciel de contrôle d'accès.

Activité 2.8: Déterminer les problèmes humains et des solutions (résistance au changement, problèmes d'ergonomie, utilisation de formulaires, dialogue Homme/Machine, ...).

Informations utiles : 1.5 - les limites et utilitaires; 1.4 - 2.1 - les exigences et objectifs du système; 2.3 - la description des sous-systèmes; 2.4 - les I/O du système et le dictionnaire de donnée; 2.6 - la description des traitements; les normes ergonomiques, sociales, organisationnelles; la législation sur le travail.

Activité 2.9: Concevoir la structure logique des données et les chemins d'accès requis (volume, accès, fréquence, ...).

Informations utiles : 2.4 - les I/O du système et le dictionnaire de données; 2.6 - la description des traitements.

Outils possibles : la normalisation, un dictionnaire de données, la modélisation conceptuelle des données.

Activité 2.10/12: Spécifier les facilités requises en communications de données (temps de réponse), en hardware et en software. Cela fera partie du cahier des charges.

Informations utiles : 1.4 - 2.1 - les exigences et objectifs du système ainsi que les critères de prestation; 1.8 - 2.2 - le cadre du nouveau système; 2.3 - la description des sous-systèmes; 2.4 - les I/O exigées; 2.5/6 - la description des traitements; 2.7 - la sécurité souhaitée; 2.9 la structure des données, le volume des données, la fréquence d'accès.

Activité 2.13: Elaborer un plan pour la poursuite du développement et la conversion et la mise en production (conversion, formation, installation, ...); révision de l'analyse bénéfices/coûts.

Informations Utiles : Le rapport de la 1^{re} phase; les résultats des différentes activités de la phase 2.

Activité 2.14: Rédiger le rapport d'analyse fonctionnelle. Il sert de base pour prendre la décision de poursuivre, de recommencer ou d'arrêter.

Informations utiles : Tous les résultats obtenus jusqu'ici.

Outil : le dictionnaire de données.

3.4 activités importantes

- Définir l'étendue du système (repréciser - évaluer).
- Approche intégrale de la description du système pour garder sa cohérence (les interfaces).
- Diviser éventuellement le système global en sous-systèmes.
- Décrire le système ou chaque sous-système (structure des données, interfaces, tâches, activités).

4 PHASE 3 : ANALYSE TECHNIQUE

4.1 description

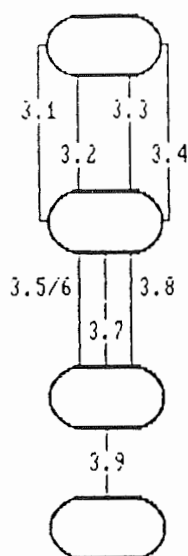
L'analyse fonctionnelle est détaillée - par sous-système éventuellement - en vue de définir les spécifications relatives à la programmation. C'est ici que l'on définit les procédures de traitement, la structure et l'organisation physique des bases de données et des fichiers, les tâches, les critères relatifs au matériel et au logiciel, la découpe en programmes. L'analyse technique reste aussi indépendante que possible d'outils de réalisation.

Il est préférable de traiter la partie de l'analyse technique qui concerne l'homme (tâche - interface) avant la partie machine (programme). Bien que cela puisse en principe se faire simultanément. En effet, il faut adapter la machine à l'homme et non l'inverse.

Dans les systèmes de moyenne et de grande importance, la distinction entre analyse fonctionnelle et technique est bien marquée car à partir de l'analyse technique, les sous-systèmes peuvent suivre une évolution distincte (remarque : la réalisation des programmes d'un sous-système ne peut commencer avant que l'analyse technique de ce sous-système ne soit complètement terminée). Pour les systèmes plus petits, suivant leur complexité, suivant les réalisateurs de l'analyse fonctionnelle, les deux phases peuvent être fusionnées.

A la fin de cette phase, tout doit être spécifié pour que le reste du développement se déroule au mieux.

4.2 planning possible des activités



Activité 3.1: Conception des procédures manuelles

Activité 3.2: Conception des formulaires et de tous les input et output de l'ordinateur

Activité 3.3: Conception de la structure de stockage

Activité 3.4: Conception des mesures de sécurité

Activité 3.5/6: Elaboration des descriptions de programmes et schémas

Activité 3.7: Description des programmes standards à utiliser

Activité 3.8: Elaboration d'un plan détaillé de programmation et de test

Activité 3.9: Rédaction du rapport d'analyse technique

Figure 3: Planning de la phase 3

4.3 description des activités et justification du planning

Activité 3.1: Conception des procédures manuelles, des manuels d'aide ainsi que les moyens d'aide pour la formation des utilisateurs.

Informations utiles : 2.4 - les spécifications des I/O, le dictionnaire de données; 2.5/6 - la description des traitements, les dialogues; 2.8 - les problèmes humains possibles et leurs solutions; 3.2 - les formulaires et les I/O de l'ordinateur.

Activité 3.2: Conception des formulaires et de tous les inputs et outputs de l'ordinateur (design des écrans, listes, ...).

Informations utiles : 2.4 - les spécifications des I/O, le dictionnaire de données; 2.5/6 - la description des traitements; 2.8 - les aspects ergonomiques.

Activité 3.3: Conception de la structure de stockage (BD physique, fichiers).

Informations utiles : 1.2/3 - recueil de données sur l'existant; 2.1 - les exigences du système; 2.2 - le cadre du système; 2.4 - les spécifications des I/O, le dictionnaire de données; 2.5 - la description des traitements; 2.7 - les exigences en sécurité; 2.9 - la structure des données.

Activité 3.4: Conception des mesures de sécurité (contrôle, sécurité, confidentialité, solutions de remplacement, reconstitution au démarrage, ...).

Informations utiles : 2.7 - les exigences en sécurité.

Activité 3.5/6: Elaboration des descriptions des programmes, des modules et schemas (partage des écrans, fichiers temporaires, ...).

Informations utiles : 2.4 - le dictionnaire de données; 2.5 - la description des traitements; 3.1 - les procédures; 3.2 - la classification des I/O; 3.3 - la structure de stockage; 3.4 - les mesures de sécurité.

Outils possibles : le pseudo-code, les tables de décision, la programmation structurée (J.S.P.), la statique des traitements.

Activité 3.7: Description des programmes standard à utiliser.

Informations utiles : 2.5 - la description des traitements; 2.12 - les spécifications des softwares; 3.5 - les spécifications des programmes.

Activité 3.8: Elaboration d'un plan détaillé de programmation et tests des programmes (données et environnement de test).

Informations utiles : Tous les résultats de la phase.

Activité 3.9: Rédaction du rapport d'analyse technique.

Informations utiles : Tous les résultats obtenus jusqu'ici.

4.4 activités importantes

- Description détaillée de la nouvelle organisation.
- Description complète du système.

5 PHASE 4 : REALISATION DES PROGRAMMES

5.1 description

Les programmes sont conçus en vue de l'essai du (sous-)système, sur base des spécifications définies au cours de l'analyse technique. C'est ici que l'on écrit les programmes, essaye ceux-ci et que l'on décrit les tâches en fonction des procédures et vérifie leur exactitude dans la pratique.

Cette phase consiste à vérifier les spécifications imposées aux programmes, à coder, à compiler *à l'aide d'un éditeur* les programmes et à tester leur logique.

Remarque : C'est le dernier moment où l'on peut agir sur la description des tâches et les exigences du système.

5.2 planning possible des activités

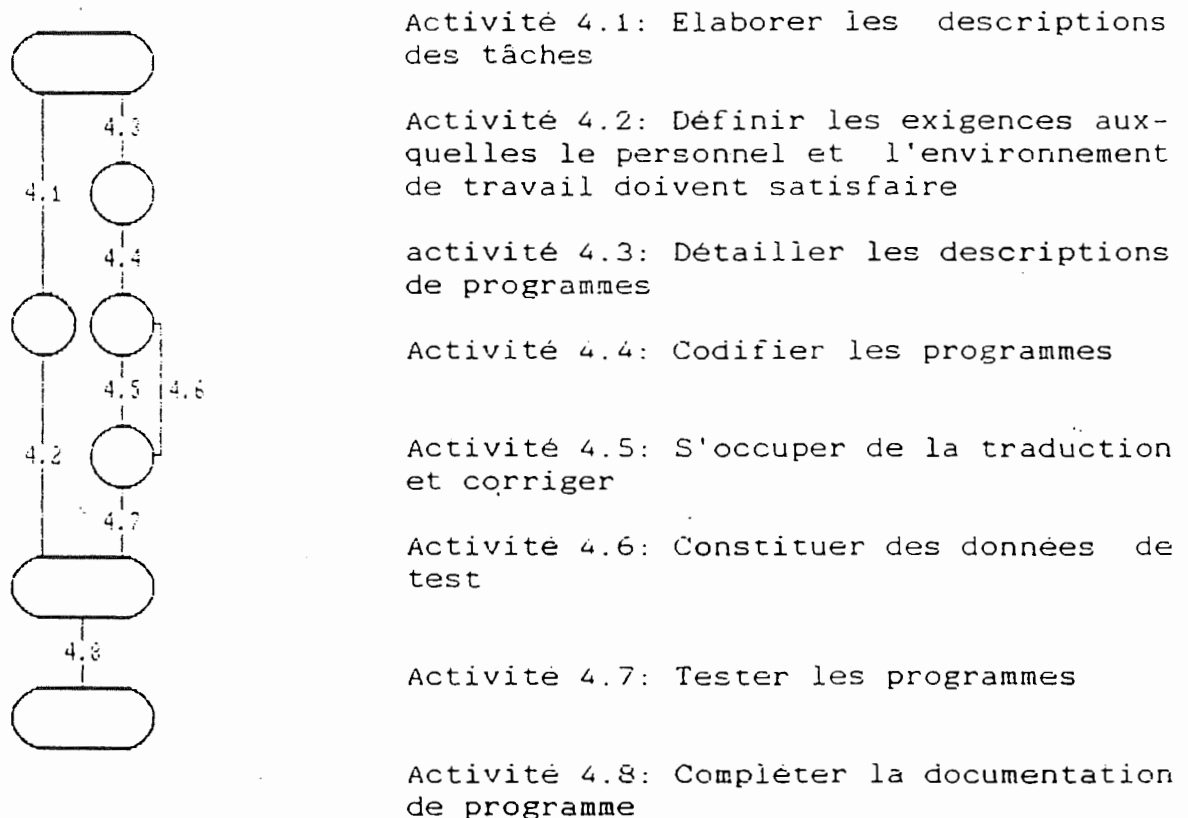


Figure 4: Planning de la phase 4

5.3 description des activités et justification du planning

Activité 4.1: Elaborer les descriptions des tâches et les normes d'exécution.

Informations utiles : 2.8 - les facteurs ergonomiques; 3.1 - les descriptions des procédures et des guides-utilisateurs; 3.2 - les formulaires à utiliser et la description des I/O.

Activité 4.2: Définir les exigences auxquelles le personnel et l'environnement doivent satisfaire (expérience requise, formation, engagement de personnel, exigences techniques, ...).

Informations utiles : 2.8 - les facteurs ergonomiques; 3.1 - les guides-utilisateurs; 3.2 - les formulaires, les utilitaires à utiliser, la description des I/O; 4.1 - la description des tâches; la législation du travail; les formations disponibles.

Activité 4.3: Détailler les descriptions de programmes (structure).

Informations utiles : 3.2 - la division des écrans, des sorties; 3.3 - la structure de stockage; 3.5/6 - les spécifications des programmes.

Outils possibles : le pseudo-code, la statique des traitements pour les modules, la programmation structurée (J.S.P. - le langage structuré), les tables de décision.

Activité 4.4: Codifier les programmes.

Informations utiles : 2.4 - les données du dictionnaire; 3.2 - la division des écrans, des imprimés; 3.3 - la structure de stockage; 3.4 - les mesures pour le redémarrage; 4.3 - les spécifications des programmes; les manuels de programmation; les normes et standards.

Outils possibles : les langages de programmation (cobol, ...).

Activité 4.5: compiler et corriger (programmes-source, programmes-objet).

Informations utiles : 4.3 - les spécifications des programmes; 4.4 - les programmes codés; les standards des JCL.

Outils possibles : les manuels des différents langages, les debuggers.

Activité 4.6: Constituer des données de test.

Informations utiles : 3.2 - la division en enregistrements ; 3.4 - la structure de stockage; 4.3 - la structure des programmes; 4.5 - les programmes-source.

Outils possibles : les tables de décision, les générateurs de données de test.

Activité 4.7: Tester la logique des programmes et résultats obtenus.

Informations utiles : 4.3 - les spécifications des programmes; 4.5 - les programmes-source et les programmes-objet; 4.6 - les données de test des programmes.

Activité 4.8: Compléter la documentation de programme.

Informations utiles : Toute la documentation sur les programmes.

Outil : un dictionnaire de données.

5.4 activités importantes

- La codification des programmes.
- Le test des programmes.

6 PHASE 5 : TESTS

6.1 description

Remarque : un contrôle utilisateur s'effectue lors de toutes les phases a la production du rapport.

Les programmes ayant été testés sur le plan de leur logique, ils font à présent l'objet de tests d'intégration dans l'ensemble du système. Il faut définir des procédures rigoureuses de test. (Remarque : on peut distinguer différentes sortes de test : de programmes, de système, de masse, d'environnement, d'acceptation)

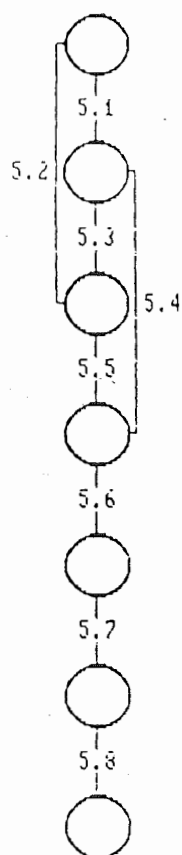
La responsabilité du test des procédures incombe aux concepteurs du système (les utilisateurs s'occupent des procédures manuelles et les informaticiens s'occupent des procédures automatiques).

Le test de la cohérence entre les programmes et les procédures est de la compétence des concepteurs mais est souvent réalisé par les programmeurs.

Les informaticiens regardent l'intégration du projet avec les autres applications et les demandeurs du projet réalisent le test d'acceptation.

Enfin, on procède, normalement, à un test d'acceptation formelle (exemple : pour les package).

6.2 planning possible des activités



Activité 5.1: Elaboration d'un plan de test détaillé

Activité 5.2: Installation du matériel, du software et préparation de l'environnement de travail

Activité 5.3: Définition des unités de traitement et leur ordre de succession

Activité 5.4: Test des formations, utilitaires et procédures

Activité 5.5: Constitution des données de test de système et d'acceptation

Activité 5.6: Test du (sous-)système

Activité 5.7: Exécution du test d'acceptation

Activité 5.8: Rédaction du rapport "Résultats de test"

Figure 5: Planning de la phase 5

6.3 description des activités et justification du planning

Activité 5.1: Elaboration d'un plan de test détaillé et de procédures de test.

Informations utiles : 1.9 - les critères d'acceptation; 1.4 - 2.1 - les critères de prestation et les exigences du système; 3.1 - les procédures; 2.7 - 3.4 - les mesures et exigences en sécurité; 3.8 - le plan de test; 3.9 - l'analyse technique; 4.6 - la documentation des programmes.

Activité 5.2: Installation du matériel, du software et préparation de l'environnement de travail et test (en général, cela est réalisé beaucoup plus tôt).

Informations utiles : 1.4 - les critères de prestation; 2.1 - les exigences du système; 2.10/12 - les facilités requises en communication de données, en hardware et en software; 2.13 - le plan d'installation; la littérature spécialisée; l'expérience d'autres utilisateurs.

Activité 5.3: Définition des unités de traitement (jobs) et leur ordre de succession (job stream).

Informations utiles : 4.7 - les programmes testés; les standards et les normes de job control; ...

Activité 5.4: Test des formations (pour la conversion et la mise en production), des utilitaires et des procédures sur un public averti (prérequis pour le test d'acceptation).

Informations utiles : 3.1 - les procédures et les manuels-utilisateurs; 3.2 - les formulaires, la description des I/O; 4.1 - la description des tâches; 4.2 - la formation, les utilitaires.

Activité 5.5: Elaborer des données de test de système et d'acceptation.

Informations utiles : 1.9 - les critères d'acceptation; 1.4 - 2.1 - les critères de prestation et les exigences du système; 3.3 - la structure de stockage; 3.4 - les mesures de sécurité; 3.6 - les spécifications des programmes; 3.7 - les standard de programmation à utiliser; 5.1 - le plan de test.

Outils possibles : les tables de décision.

Activité 5.6: Test du (sous-)système avec éventuellement les modifications apportées.

Informations utiles : 1.4 - 2.1 - les critères de prestation et les exigences du système; 2.7 - 3.4 - les mesures et exigences en sécurité; 4.1 - la description des tâches; 4.7 - les programmes testés; 5.4 - les procédures testées; 5.5 - les données de test; 5.7 - le plan de test.

Activité 5.7: Exécution du test d'acceptation avec la liste des défauts constatés.

Informations utiles : 1.4 - 2.1 - les critères de prestation et les exigences et objectifs du système; 1.9 - les critères d'acceptation; 2.7 - 3.4 - les mesures et exigences en sécurité; 5.5 - les données du test d'acceptation; la documentation du système.

Outils possibles : les tables de décision.

Activité 5.8: Rédaction du rapport "Résultats de test". Le (sous-)système est accepté ou refusé.

Informations utiles : Tous les résultats des tests.

6.4 activités importantes

- Installation et test du matériel et du software de base.
- Test du système et des formations avec un public-test.
- Test d'acceptation.

7.3 description des activités et justification du planning

Activité 6.1: Elaborer un plan détaillé de conversion et de mise en production.

Informations utiles : 2.13 - le plan de conversion et de mise en production ainsi que le plan pour l'installation des facilites; 5.8 - les resultats des tests; les possibilites de formation.

Activité 6.2: Formation du personnel informatique et évaluation.

Informations utiles : 3.4 - les procédures de reprise; 5.2 - le résultat de l'installation et du test du hardware et du software; 5.3 - les instructions pour les opérateurs; 5.4 - la formation; les normes et standards du centre de calcul.

Activité 6.3: Elaboration d'instructions de conversion et de mise en production et contrôle de la documentation.

Informations utiles : 1.2 - les données sur l'existant; 2.8 - les formulaires; 2.9 - 3.3 - les données et la structure de stockage du nouveau système; 3.1 - les procédures; 4.1 - la description des tâches; les manuels utilisateurs; les utilitaires disponibles pour la conversion.

Activité 6.4: Conversion des données et compte rendu des fautes et problèmes constatés.

Informations utiles : 2.9 - la structure des données; 3.3 - la structure de stockage, la division en enregistrements, les utilitaires pour la conversion; 4.1 - la description des tâches; 6.1 - le plan de conversion et de mise en production; 6.3 - les instructions pour la conversion.

Activité 6.5: Donner des renseignements sur le nouveau système (réunion et évaluation).

Informations utiles : 1.4 - les critères de prestation, les objectifs et exigences du système; 1.8 - les concepts du système; 1.11 - le rapport de la 1^{re} phase; 2.3 - 2.6 - la description des sous-systèmes et des traitements; 2.13 - le rapport de la 2^{re} phase; 3.1 - les procédures; 4.1 - la description des tâches; 6.1 - le planning pour la conversion et la mise en production; 6.3 - les manuels utilisateurs.

Activité 6.6: Formation des utilisateurs et évaluation.

Informations utiles : 3.1 - les procédures ; 4.1 - la description des tâches; 5.4 - la formation et les utilitaires; 5.4 - 6.1 - le plan de formation; 6.3 - les manuels utilisateurs; des cours donnés par l'entreprise ou par les fournisseurs.

Activité 6.7: Formation du groupe de maintenance et évaluation.

Informations utiles : 5.4 - le plan de formation; 5.5 - les données de test; 5.7 - la documentation du système; 6.3 - les manuels d'entretien; les standards, les normes, les manuels; les cours des fournisseurs, de sociétés de software.

Activité 6.8: Mise en production du nouveau système, son transfert et évaluation. La documentation des (sous-)systèmes est définitive et la documentation du projet est cloturée.

Informations utiles : la documentation du système et du projet.

Outil : un dictionnaire de donnée.

7.4 activités importantes

- Le passage de l'ancien système vers le nouveau.
- La formation de toutes les personnes concernées.

8 PHASE 7 : UTILISATION ET GESTION DU SYSTEME

8.1 description

Cette phase se prolonge tant que le système est utilisé. C'est ici que l'on verra comment il fonctionne dans la pratique.

Il faut être particulièrement attentif aux procédures de réparation, à la sécurité et aux catastrophes prévisibles. Les préparatifs de la sécurité doivent être envisagés dès l'analyse conceptuelle.

Au cours de cette phase, on vérifie périodiquement que le système répond toujours bien à la demande des utilisateurs. Dans la négative, cela peut donner lieu à de nouvelles activités (modification du système).

Bien qu'une importante partie des frais entraînés par un système d'informations informatique soient absorbés par cette phase, l'exploitation et la gestion du système sont soumises à très peu de méthodes et de techniques. On peut s'attendre à ce qu'à l'avenir, lorsque les personnes responsables de la méthode auront acquis l'expérience de très nombreux systèmes, ils puissent généraliser des techniques standard pour mener cette phase à bien.

Cette phase termine le cycle de vie du système.

8.2 planning possible des activités

Activité 7.1: Elaboration et utilisation du système de signalement des erreurs

Activité 7.2: Planification périodique de maintenance

Activité 7.3: Planification du traitement informatique

Activité 7.4: Prévention et restauration des fautes et perturbations

Activité 7.5: Surveillance des dispositions de sécurité

Activité 7.6: Modifications et mise à jour de la documentation. Réponse aux besoins en information ad hoc

Activité 7.7: S'occuper des formations complémentaires

Activité 7.8: Gestion des données et des fichiers

Activité 7.9: Evaluation du système et plans d'action

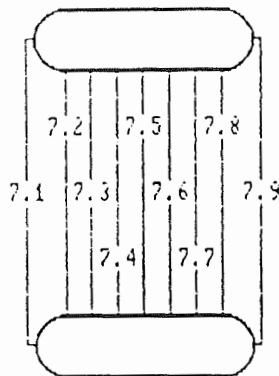


Figure 7: Planning de la phase 7

8.3 description des activités et justification du planning

Activité 7.1: Elaboration et utilisation du système de signalement des erreurs via un canal unique de communication.

Informations utiles : les exigences du système, les critères de prestation; les normes, les standard; rapport d'utilisation, nombre de transactions, ...

Activité 7.2: Planification périodique de maintenance (activités à réaliser).

Informations utiles : 7.1 - le rapport sur le signalement des erreurs; 7.6 - les propositions de modifications; 7.7 - les formations complémentaires souhaitées; 7.9 - le rapport d'évaluation; la documentation du système.

Activite 7.3: Planification du traitement informatique.

Informations utiles : les normes, les estimations du travail
actuel et futur; les manuels de production; ...

Activité 7.4: Prévention et restauration des fautes et perturbations .

Informations utiles : les rapports sur les problèmes survenus; les procédures de reconstruction et de redémarrage; les manuels du centre de calcul.

Activité 7.5: Surveillance des dispositions de sécurité et les tester (plan catastrophe, accès au centre de calcul).

Informations utiles : 2.7 - les exigences en sécurité; 3.4 - les mesures de sécurité; 7.4 - les rapports sur fautes et les mesures prises; la documentation du système; les programmes et techniques de contrôle.

Activité 7.6: Modifications et mise à jour de la documentation. Réponse aux besoins en information ad hoc.

Informations utiles : 5.5 - les données de test; 7.9 - le rapport d'évaluation; les souhaits des utilisateurs; les possibilités pour corriger; de nouveaux appareils.

Activité 7.7: S'occuper des formations complémentaires (exemple : une application pratique).

Informations utiles : 3.1 - les procédures; 4.1 - la description des tâches; 5.4 - les formations et utilitaires disponibles; les manuels utilisateurs.

Activité 7.8: Gestion des données (base de données) et des fichiers (structure des données et du stockage). Le dictionnaire de données est mis à jour.

Informations utiles : 7.1 - le système de signalement; la documentation du système.

Outil possible : un dictionnaire de données.

Activité 7.9: Evaluation du système et plans d'action.

Informations utiles : 1.10 - l'analyse bénéfices/coûts; 2.1 - les exigences et objectifs du système; les normes et standards; la documentation du système.

8.4 activités importantes

- La maintenance du système.
- Surveillance du système.

9 SYNTHESE

9.1 pour les données

- 1.2 : Rassembler des données sur l'existant
- 1.3 : Evaluer et analyser ces données
- 1.6 : Schema du nouveau système (MCD)
- 2.4 : Description des données
- 2.9 : Structure logique des données et chemins d'accès (quantification - PALOMA - normalisation)
- 3.2 : Conception des formulaires, des I/O de l'ordinateur
- 3.3 : Conception de la structure de stockage (DB physique)

9.2 pour les traitements

- 1.2 : Rassembler des données sur l'existant (tâches)
- 1.3 : Evaluer et analyser ces données
- 1.6 : Schéma du nouveau système (SHAT - DFD - IPO)
- 2.3 : Division en sous-systèmes et tâches et les décrire
- 2.5/6 : Diagramme des traitements, description des activités et des tâches
- 3.5/6 : Elaboration des descriptions de programmes
- 3.7 : Description des programmes standards à utiliser
- 4.1 : Elaborer les descriptions de tâches

9.3 pour la sécurité

- 1.3 : Analyse et évaluation des données de l'existant (confidentialité de certaines tâches, clés d'accès)
- 1.4 : Contraintes du système (recovery - privacy - solution de remplacement)
- 2.2 : Cadre du nouveau système (contraintes d'intégrité : fall-back, recovery, solution de remplacement)
- 2.7 : Spécifier les exigences en sécurité, confidentialité
- 3.4 : Conception des mesures de sécurité
- 7.4 : Prévention et restauration des fautes et perturbations
- 7.5 : Surveillance des dispositions de sécurité

BIBLIOGRAPHIE

- [BO-PY] CONCEPTION ASSISTEE DES APPLICATIONS INFORMATIQUES - "1. ETUDE D'OPPORTUNITE ET ANALYSE CONCEPTUELLE."
F. BODART, Y. PIGNEUR - MASSON
- [DEF-AC] DEFINITION UTILISEES LORS DE L'ANALYSE CONCEPTUELLE - V1.0
C. GRAAS - CGER - Avril 1988
- [DEM-DEV] DEMANDE DE DEVELOPPEMENT - V1.6
C. GRAAS, R. VAN'T DACK - CGER - Juillet 1988
- [DMR] NOTES SUR LE DICTIONNAIRE DE DONNEES DATAMANAGER
- [IMS] GUIDANCE POUR LE DEVELOPPEMENT DE PROJETS EN IMS
B. DELVAUX - CGER - Septembre 1988
- [MCM] MODELISATION CONCEPTUELLE (MCM) - V1.0
C. GRAAS, A. SPILTOIR, R. VAN'T DACK - CGER - Septembre 1988
- [SDM-ANG] SDM - SYSTEM DEVELOPMENT METHODOLOGY
W.S. TURNER, R.P. LANGERHORST, G.F. HICE, H.B. EILERS, A.A. UIJTENBROEK - Elsevier science publishers B.V. - 1987
- [SDM-DOLMEN] COURS SDM (SYSTEM DEVELOPMENT METHODOLOGY - CGER)
Société DOLMEN - Octobre 1988
- [SDM-NEER] SYSTEEM ONTWIKKELING VOLGENS SDM
H.B. EILERS - Academic service - 1987
- [SPEC-AN] ANALYSE DES SPECIFICATIONS
C. GRAAS, R. VAN'T DACK - CGER - Septembre 1988

ANNEXE 3

**Questionnaire distribué aux
informaticiens de la CGER**



Muriel DETAILLE

16 décembre 1991

**Note à Mesdames et Messieurs les informaticiens des Services
DSI-1, DSI-2, DSI-M et DSI-3 (groupes 44, 55, 65, 68)**

Copie à MM. JACHOWICZ, Premier conseiller
LAPERCHE, info D
LEMAIRE, info D
HAVET, info C

Je suis étudiante aux Facultés Universitaires Notre Dame de la Paix de Namur et je termine une licence et maîtrise en informatique.

J'accomplis actuellement un stage au DSI-M pour réaliser un mémoire de fin d'études traitant de la documentation des développements informatiques avec la méthode SDM.

C'est pourquoi je me permets de vous proposer le questionnaire en annexe afin de mettre en évidence les éléments de documentation proposés par SDM et effectivement utilisés à la CGER.

Ce questionnaire comporte une grille de réponse et reprend la documentation relative à chaque activité d'SDM. Je vous saurais gré de répondre aux questions suivantes :

- ° Produisez-vous ce type de documentation (oui, sommairement ou non) ?
- ° Si oui, la produisez-vous à l'aide d'un outil logiciel et dans ce cas, lequel ?
- ° Si non, pensez-vous que cette documentation soit automatisable et dans ce cas la produiriez-vous, si vous disposiez des outils informatiques nécessaires ?

Afin de terminer à temps mon travail, j'aimerais recevoir votre réponse avant le 10 janvier prochain. Il va de soi que je serai heureuse de communiquer les résultats de mon enquête à tous ceux qui auront bien voulu y participer ou qui en feront la demande.

Je compte beaucoup sur votre collaboration pour mener au mieux ma tâche et je tiens à vous exprimer d'avance mes vifs remerciements.

Muriel DETAILLE

c:mdet9101-nj

Informations générales.

1. Données personnelles (facultatives).

Nom:

Prénom:

Service:

2. Classification de vos projets.

2.1 Taille en mois/homme:

2.2 Incertitude quant aux spécifications ⁽¹⁾

Cote: (5: spécifications précises
0: pas ou peu de spécifications)

2.3 Incertitude technique ⁽¹⁾

Cote: (5: environnement technique connu
0: nouvel environnement technique)

(1) Pour la détermination de cette cote, vous pouvez vous référer à la note de Mr Jachowicz concernant l'évaluation des priorités des projets informatiques. (du 15.04.91)

PHASE 1: Etude de définition et analyse conceptuelle.

Activités:

1. Définition du problème.
- 2/3. Analyse de la situation actuelle.
4. Objectifs et exigences du système.
5. Points non résolus et hypothèses de base.
6. Schémas du nouveau système.
7. Outils et solutions possibles.
8. Evaluation et sélection des solutions.
9. Problèmes de conversion et mise en prod.
10. Planning et analyse coût-bénéfice.
11. Rédaction du rapport final de la phase.

Documentation:

- Projet-cadre.
- Liste des problèmes rencontrés.
Schémas commentés du système.
Cahier des charges sommaire.
Description de la solution.
Estimation des moyens et des coûts.
Planning global + analyse coût-bénéfice.
Rapport final de l'analyse conceptuelle.

Réalisé ⁽¹⁾ O/N/S	Outil ⁽²⁾	Automatisable ⁽³⁾	
		O/N	Souhaité

PHASE 2: Analyse fonctionnelle.

Activités:

1. Exigences du système.
2. Cadre de fonctionnement.
3. Division en sous-systèmes.
4. Entrées et sorties de chaque sous-système + interfaces.
- 5/6. Diagrammes des traitements.
7. Exigences de sécurité et de confidentialité.
8. Pratiques humaines et solutions.
9. La structure logique des données.
- 10/12. Moyens en software, hardware, ...
13. Planning des étapes ultérieures.
14. Rédaction du rapport final de la phase.

Documentation:

- (document 1.4 détaillé)
(document 1.8 détaillé)
- Schéma de la découpe en sous-systèmes et descriptions.
Schémas et descriptions des traitements.
(document 2.1 détaillé)
- Schéma logique des données commenté.
Cahier des charges complet.
Plannings détaillés.
Rapport final de l'analyse fonctionnelle.

Réalisé ⁽¹⁾ O/N/S	Outil ⁽²⁾	Automatisable ⁽³⁾	
		O/N	Souhaité

PHASE 3: Analyse technique.

Activités:

1. Conception des procédures manuelles.
2. Conception des formulaires et des input/output.
3. Structure de stockage.
4. Conception des mesures de sécurité.
- 5/6. Description des programmes, et schémas.
7. Programmes standards à utiliser.
8. Plan de programmation et de test.
9. Rédaction du rapport final de la phase.

Documentation:

- Manuels d'utilisation et des procédures.
Document 2.4 auquel on a ajouté l'aspect physique des éléments.
Schéma physique des données.
Description des mesures de sécurité.
Schémas des programmes.
- Planning détaillé des phases 4 et 5.
Rapport final de l'analyse technique = cahier des charges du programmeur.

Réalisé ⁽¹⁾ O/N/S	Outil ⁽²⁾	Automatisable ⁽³⁾	
		O/N	Souhaité

PHASE 4: Programmation et tests isolés.

Activités:

1. (Description des tâches).
2. Exigences au niveau du personnel et de l'environnement de travail.
3. Descriptions détaillées des programmes.
4. Codification des programmes.
5. Traduction et correction des programmes.
6. Constitution des données de test.
7. Test des programmes.
8. Compléter la doc. des programmes.

Documentation:

- Programmes sources commentés.
- Liste des fichiers créés et plan de test.
Liste des problèmes importants rencontrés.
Documentation des programmes māj.

Réalisé ⁽¹⁾ O/N/S	Outil ⁽²⁾	Automatisable ⁽³⁾	
		O/N	Souhaité

Grille de réponse (1).

⁽¹⁾ O = oui ; N = non ; S = sommairement.

⁽²⁾ Répondre soit N pour non, soit préciser le nom de l'outil utilisé.

⁽³⁾ Préciser O = oui ou N = non en première sous-colonne ; si oui, préciser dans la deuxième sous-colonne si vous souhaiteriez disposer de cet outil.

PHASE 5: Réalisation des tests.

Activités:

Documentation:

1. Plan de test.
2. Installation du matériel, des logiciels,...
3. Définition des unités de traitement.
4. Test des formations, des procédures,...
5. Constitution des données de test.

- Planning 3.8 détaille.
- Réalisation de JCL.
- Document de description des programmes de test et des fichiers à utiliser.
- Liste des problèmes rencontrés.
- Liste des défauts constatés.
- Rapport proposant les résultats des tests commentés.

Réalisé ⁽¹⁾ O/N/S	Outil ⁽²⁾	Automatisable ⁽³⁾	
		O/N	Souhaité

PHASE 6: Conversion et mise en production.

Activités:

Documentation:

1. Plan de l'étape.
2. Formation du personnel informatique.
3. Instructions de conversion et de mise en production.
4. Conversion des données.
5. Renseignements sur le nouveau système.
6. Formation des utilisateurs.
7. Formation du personnel de maintenance.
8. Mise en production et transfert du nouveau système.

- Planning 2.13 détaille.

- Liste des problèmes rencontrés.

Réalisé ⁽¹⁾ O/N/S	Outil ⁽²⁾	Automatisable ⁽³⁾	
		O/N	Souhaité

Grille de réponse (2).

⁽¹⁾ O = oui ; N = non ; S = sommairement.

⁽²⁾ Répondre soit N pour non, soit préciser le nom de l'outil utilisé.

⁽³⁾ Préciser O = oui ou N = non en première sous-colonne ; si oui, préciser dans la deuxième sous-colonne si vous souhaitez disposer de cet outil.

Remarques éventuelles.

ANNEXE 4

**Exemple simplifié d'application de
IFPA**

CONTENU:

1. Généralités
2. Principes de la documentation en datamanager.
3. Calcul et documentation des points de fonction
 - 3.1 Définition d'une tâche
 - 3.2 Tâches et IFPA
 - 3.2.1 Types de tâches
 - 3.2.2 Comment trouver les tâches ?
 - 3.3 Exemples
 - 3.3.1 Tâches d'introduction
 - 3.3.2 Tâches d'extraction
 - 3.3.3 Tâches d'interrogation
 - 3.3.4 Tâches de couplage
4. Projet et IFPA
5. Gestion des facteurs de productivité

SOURCE : "IFPA Handbook", (1991), manuel réalisé à la CGER par Monsieur Christian GRAAS, Informaticien.

1. GENERALITES

La méthode d'analyse par points de fonction permet d'évaluer la charge et partant, le temps de développement des systèmes informatiques.

Le principe général en est d'attribuer des points de fonction aux fonctionnalités d'un système requises par l'utilisateur, puis de convertir ce nombre de points de fonction en heures-homme au moyen de coefficients de productivité.

La CGER utilise la méthode IFPA et a réalisé les adaptations suivantes : l'implémentation de IFPA dans le dictionnaire Datamanager et les coefficients de productivité propres à son environnement.

Par conséquent, les matrices permettant d'attribuer des points de fonction aux tâches et aux données du système sont définies une fois pour toutes, et donc non modifiables.

Seuls les facteurs de productivité qui permettent de convertir les points de fonction en effort peuvent être modifiés, soit suite à un ou plusieurs changements dans l'environnement soit suite à une expérience accrue dans un environnement donné.

IFPA sera donc un outil de prévision aussi précis que les chefs de projet le voudront : plus on mettra de soin à définir les tâches et les données et à rapporter les heures prestées, plus les prévisions des systèmes ultérieurs seront fiables.

IFPA est documentée dans le dictionnaire central DATAMANAGER. On y documentera un membre de type PROJECT pour chaque projet pour lequel une estimation IFPA aura été faite.

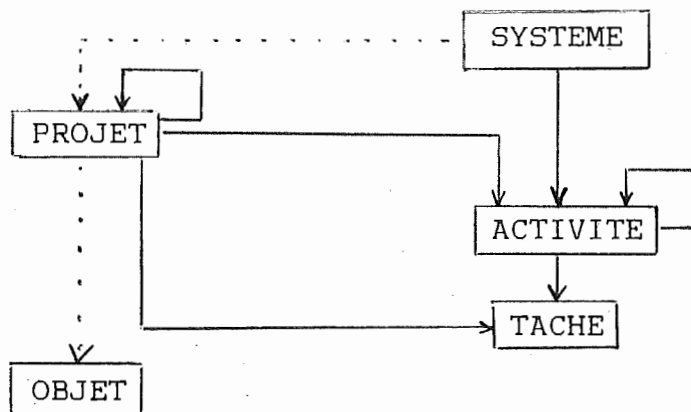
2. PRINCIPES DE LA DOCUMENTATION EN DATAMANAGER

Dans une première étape, Datamanager sera utilisé pour documenter IFPA.

Dans ce but :

- Une rubrique FUNCTION-POINT sera ajoutée aux membres de type TACHE, ACTIVITE et OBJET.
- Un nouveau type de membre PROJECT sera créé.
Ce membre devra toujours être rempli, même si le projet n'est pas documenté en DMR.
- Un membre particulier contiendra la table des facteurs de productivité.

La figure suivante montre les liens entre les types de membres cités ci-dessus :



----> contains

.....> see

3. CALCUL ET DOCUMENTATION DES POINTS DE FONCTION

3.1 DEFINITION D'UNE TACHE

Une tâche est un ensemble organisé d'opérations élémentaires, perçues et désignées globalement, manuelles et/ou automatisées, répétitives pour chaque échange d'information et transformant selon des règles prédéfinies des informations d'entrée et informations de sortie.

Elle constitue une unité de traitement aux yeux de l'utilisateur. Une tâche est donc caractérisée par son unicité de temps, de lieu et d'action.

Des accès de création, suppression ou modification de deux objets peuvent faire partie de la même tâche si et seulement si ces accès sont couplés, c'est-à-dire si l'un entraîne l'autre et réciproquement.

3.2 TACHES ET IFPA

3.2.1 Types de tâches

L'analyse des points de fonction distingue quatre types de tâches, qui sont:

- les tâches d'introduction,
- les tâches d'extraction,
- les tâches d'interrogation.
- les tâches de couplage.

Chaque tâche peut ensuite être évaluée à l'aide de matrices de pondération.

Remarque:

Quoiqu'implémentant des conversions techniques, les couplages entre systèmes sont considérés comme des tâches, pour des raisons d'uniformité.

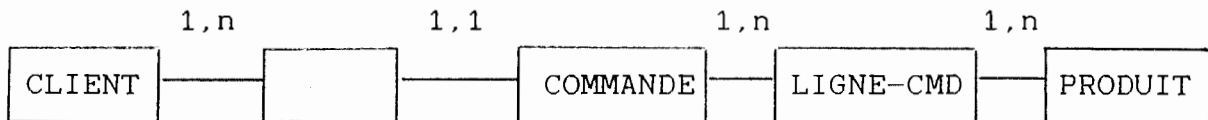
3.2.2 Comment trouver les tâches ?

En principe, pour tout objet porteur du modèle conceptuel des données propre au projet à développer, on aura au moins une tâche de création, une de modification et une de suppression ou d'archivage.

Très souvent on trouvera aussi pour chacun de ces objets une tâche d'extraction (output) et une tâche d'interrogation (query).

3.3 EXEMPLES

Les exemples suivants s'appliquent au modèle ci-dessous :



Commentaires:

°La création d'une commande et celle des lignes de commande correspondantes font partie de la même tâche car la création d'une commande implique celle de ligne(s) de commande, et la création d'une ligne de commande implique celle d'une commande.

°La création d'une commande et la création d'un client constituent au contraire deux tâches différentes car la création d'une commande **peut** impliquer la création d'un client, tandis que la création d'un client peut avoir lieu sans création d'une commande.

°Comme la tâche de création de la commande appelle dans certains cas la tâche de création d'un client, on considérera que la tâche "création d'une commande" utilise 3 objets : l'entité commande, l'association ligne de commande et la tâche création d'un client.

°L'enregistrement d'une commande et la mise à jour du stock des produits se feront dans des tâches séparées. En effet, si

l'enregistrement de la commande implique la mise à jour du stock des produits commandés, l'inverse n'est pas vrai, car on met également à jour le stock après réception d'une livraison d'un fournisseur. La mise à jour du stock d'un produit constituera donc une tâche séparée car elle peut être réutilisée dans deux contextes différents.

3.3.1 Tâches d'introduction

Des objets doivent être maintenus. Il est par conséquent indispensable que le système offre certaines facilités à l'utilisateur. Ces facilités sont des tâches à l'aide desquelles des données d'un objet peuvent être ajoutées, modifiées et supprimées. ces trois types de tâches sont appelées des tâches d'introduction.

Une tâche d'introduction est caractérisée par le fait qu'elle modifie le contenu d'un ou plusieurs objets du modèle conceptuel des données parce que les tâches d'introduction ajoutent (augmentent le nombre), modifient et suppriment des données ou des occurrences.

a. Matrice de pondération des tâches d'introduction.

Nombre d'attributs

		1-4	5-15	>15
Nombre d'objets utilisés	1	3	3	4
	2	3	4	6
	>2	4	6	6

b. Exemple

L' utilisateur modifie les données de commande à l'aide d'un écran d'introduction par lequel au maximum 10 attributs peuvent être changés de valeur; par exemple trois attributs de l'objet ENTETE DE COMMANDE et sept attributs de LIGNE DE COMMANDE.

Type de tâche :

le contenu d'objets est modifié, c'est donc une tâche d'introduction.

Comptage :

deux objets utilisés (ENTETE DE COMMANDE et LIGNE DE

COMMANDE), dix éléments.

Valorisation :

4 points de fonction.

3.3.2 Tâches d'extraction

Une tâche d'extraction d'un système fournit de l'information provenant des objets du modèle conceptuel des données de façon telle que le volume de l'extraction est indéterminé. Cela veut dire que le volume de l'extraction comporte 0 à n occurrences.

Ceci contrairement à la fonction de consultation dans laquelle le résultat de la fonction est bien déterminé.

a. Matrice de pondération des tâches d'extraction.

Nombre d'attributs

		1-5	6-19	>19
Nombre d'objets utilisés	1	4	4	5
	2-3	4	5	7
	>3	5	7	7

b. Exemple1

L'utilisateur veut un rapport écrit comportant les données suivantes de ses clients : NOM , ADRESSE , CODE POSTAL et DOMICILE (dans cet ordre), classées alphabétiquement sur NOM

Type de tâche :

présentation dont le volume de l'extraction est indéterminé, donc une tâche d'extraction.

Comptage :

nombre d'objets utilisés : un (CLIENT) nombre d'éléments : quatre (NOM, ADRESSE, CODE POSTAL, DOMICILE).

Valorisation :

4 points de fonction.

c. Exemple2

L' utilisateur veut une deuxième extraction présentée de la même façon mais triée par CODE POSTAL. Conformément aux règles cette tâche d'extraction n'est pas comptée séparément.

3.3.3 Tâches d'interrogation

Le volume d'une tâche d'interrogation est toujours déterminé d'avance car la consultation est guidée par l'introduction de caractéristiques identifiantes uniques.

Dans ce contexte on entend par "déterminé" le fait que les éléments ou les groupes d'éléments présentés ne sont pas répétés.

a. Matrice de pondération des tâches d'interrogation.

Nombre d'attributs

		1-5	6-19	>19
Nombre d'objets utilisés	1	3	3	4
	2-3	3	4	6
	>3	4	6	6

b. Exemple

L' utilisateur veut consulter toutes les données disponibles concernant l'entreprise 'VITE'. Le modèle de données indique 2 objets concernés par cette tâche. La présentation des spécifications fonctionnelles sur écran ou sur une liste indique que 35 éléments seront présentés, répartis sur 2 pages.

Type de tâche :

les résultats de la transaction sont déterminés d'avance, notamment toutes les données concernant une entreprise, c'est donc une tâche d'interrogation.

Comptage :

le nombre d'objets impliqués s'élève à 2 et le nombre d'éléments à présenter à 35.

Remarque :

le nombre de pages n'a pas d'importance.

Valorisation :

6 points de fonction.

3.3.4 Tâches de couplage

De plus en plus, les systèmes sont reliés entre eux afin de pouvoir employer plus efficacement les données qui y sont stockées. Lors de la transmission des données une conversion doit être effectuée.

Dans ce contexte, le concept de conversion est défini comme la modification des caractéristiques des objets, par exemple la modification du nom, de la longueur, du format d'un élément ou du format d'un fichier.

Un couplage est la tâche qui effectue cette conversion. Si le système en question est responsable de ce traitement, ce dernier doit être compté.

a. Matrice de pondération des tâches d'extraction.

Nombre d'attributs	1-10	11-25	>25
Valorisation	5	7	10

4. PROJET ET IFPA

Le calcul du nombre de points de fonction fait partie du point de passage de l'analyse fonctionnelle à l'analyse technique.

De préférence deux développeurs du projet, dont le chef de projet, procéderont à l'analyse. Les résultats seront ensuite discutés avec un membre du groupe de support au développement "Méthodes".

Le nombre d'heures à prendre en compte pour la réalisation concerne les phases 1 à 6 incluse de SDM.

On établira cependant deux sous-totaux :

° l'un pour les phases de spécification:

0: étude préalable,

1: étude de définition et analyse conceptuelle,

2: analyse fonctionnelle.

- ° et l'autre pour les phases de réalisation
- 3: analyse technique,
 - 4: programmation,
 - 5: test,
 - 6: conversion et mise en production.

5. GESTION DES FACTEURS DE PRODUCTIVITE

La conversion points de fonction en heures-homme se fait au moyen de facteurs de productivité.

Ceux-ci dépendent bien entendu de l'expérience acquise avec IFPA et de l'environnement de développement, deux choses qui évoluent dans le temps. Il est donc nécessaire de prévoir périodiquement des réunions de review IFPA entre le groupe de support "Méthodes" et toutes les personnes ayant mis le projet en production au cours de la dernière période. Ces réunions servent à suivre l'application de la méthode IFPA en répondant aux questions suivantes :

1. Il y a-t-il des différences entre les prévisions et les réalisations?
2. Dans l'affirmative, faut-il revoir les facteurs de productivité existants ou créer d'autres environnements?
3. Peut-on affiner les facteurs de productivité existants.
(Par exemple distinguer les phases de spécification, celles de réalisation voire celle de test?)

* *

*

ANNEXE 5

**Description du dictionnaire de
données Datamanager**

CONTENU:

1. DEFINITION D'UN DICTIONNAIRE DE DONNEES
2. REPARTITION LOGIQUE DE LA DOCUMENTATION DANS DMR
 - 2.1 ... SELON LES PHASES DE DEVELOPPEMENT
 - 2.2 HIERARCHIE DE LA DOCUMENTATION
 - 2.3 ETENDUE DE LA DOCUMENTATION
 - 2.4 TYPES DE MEMBRES
 - 2.5 ENSEMBLE DES 'MODELS' DE DOCUMENTATION DISPONIBLES
 - 2.6 SCHEMA HIERARCHIQUE DES TYPES DE MEMBRES
 - 2.7 QUELQUES RUBRIQUES ESSENTIELLES
 - 2.7.1 Le member name
 - 2.7.2 Alias
 - 2.7.3 Catalog
 - 2.7.4 Contains - calls
 - 2.7.5 See
3. PARTITION DU DICTIONNAIRE - LES STATUS
 - 3.1 GENERALITES
 - 3.2 STATUS WORK
 - 3.3 STATUS DEV
 - 3.4 STATUS QA
 - 3.5 STATUS PRODUCTION
4. LES COMMANDES D'INTERROGATION SIMPLE
 - 4.1 GENERALITES
 - 4.2 LISTE ET BUT DES COMMANDES

- 5. LES COMMANDES D'INTERROGATION COMPLEXE
 - 5.1 GENERALITES
 - 5.2 LISTE ET BUT DES COMMANDES
- 6. LES COMMANDES DE MANIPULATION
 - 6.1 GENERALITES
 - 6.2 LISTE ET BUT DES COMMANDES
- 7. LES COMMANDES DE CREATION DE LISTES 'KEPT DATA'
 - 7.1 GENERALITES
 - 7.2 LISTE ET BUT DES COMMANDES

SOURCE : " Manuel d'utilisation de Datamanager, commandes de base " (1991), manuel réalisé à la CGER par Monsieur Alain SPIGTOIR.

1. DEFINITION D'UN DICTIONNAIRE DE DONNEES

Un dictionnaire de données tel que le DATAMANAGER est une collection organisée et automatisée de définitions. C'est un ensemble d'informations sur les informations intervenant dans un développement.

Le DMR est un dictionnaire de méta-informations.

Par exemple:

un 'Client' a un nom en 15 positions alphanumériques :
c'est une méta-information;
le nom d'un client 'DUPONT' :
c'est une information.

Le DMR rassemble ainsi des définitions standardisées. Son unicité et sa position centrale assurent la mise à disposition d'une documentation à jour et cohérente.

Le dictionnaire est destiné à contenir la documentation constituée tout au long d'un cycle de vie de développement de projet. Il permet la production de rapports à tout moment.

Le principal intérêt du dictionnaire est de retrouver aisément les informations à traiter par un processus, quel qu'il soit.

2. REPARTITION LOGIQUE DE LA DOCUMENTATION DANS DMR

2.1 ... SELON LES PHASES DE DEVELOPPEMENT

Un projet et son développement se déroulent en plusieurs phases décrites dans la méthode SDM.

Le dictionnaire DMR permet de documenter le développement d'un projet en 3 domaines bien distincts correspondant aux phases essentielles de cette méthode :

- l'analyse fonctionnelle est documentée en FUDOS :
documentation sur la 'tâche', l''objet', l''élément',...
- l'analyse technique est documentée en TEDOS :
documentation sur le 'programme', le 'record', le
'field',...
- la phase de mise en production est documentée en PRODOS :
documentation sur le 'job', le 'dataset',...

2.2 HIERARCHIE DE LA DOCUMENTATION

La documentation est aussi organisée d'une façon hiérarchique

Exemples :

- un FILE contient des RECORDS, composés eux-mêmes de GROUPS, et ceux-ci de FIELDS;
- un PROGRAM fait appel à des ROUTINES qui utilisent des PARAMETERS

2.3 ETENDUE DE LA DOCUMENTATION

La documentation concerne tout autant les 'PROCESS' que les 'DATA'

2.4 TYPES DE MEMBRES

La documentation se répartit sur un ensemble de types de membres, chacun ayant un rôle prédéfini.

Chaque type de membre est structuré d'une façon standard en diverses rubriques, avec pour chacune, une syntaxe à respecter.

La documentation ne peut se trouver que dans les différentes rubriques prévues; par contre, certaines d'entre elles pourront être supprimées.

Ainsi, chaque type de membre est défini par un 'model'.

Exemple: 'model' du type de membre RECORD

```
*****
RECORD
*****
ALIAS
  NEDERLANDS
  "
  ,FRANCAIS
  "
  ,GEBRUIKER
  "
  ,UTILISATEUR
  "
  ,IMS
  "
*****
CATALOGUE
  "TEST"
  ,".other.."
*****
```

```

DESCRIPTION
"NEDERLANDS : "
"
"
"
"FRANCAIS : "
"
"
"
*****
NOTE
"NEDERLANDS : "
"
"
"
"FRANCAIS : "
"
"
"
*****
CONTAINS
member-name-1 cversion!
  c ELSE
  c member-name-x cversion!
  c IF member-name EQ/NE/GT/LT/...member-name or literal
,member-name-m cversion!
,member-name-z cversion!
,(integer) member-name cversion!
*****
KEYS
  member-name-1 UNIQUE or DUPLICATED  ASCENDING or DESCENDING
  member-name-2 UNIQUE or DUPLICATED  ASCENDING or DESCENDING
  member-name-n UNIQUE or DUPLICATED  ASCENDING or DESCENDING
*****
DEVELOPMENT-DATA
"NEDERLANDS : "
"
"
"FRANCAIS : "
"
"
*** END OF MODEL *****

```

Exemple: type de membre RECORD 'model' documenté :

```

*****
REPORT OF RECORD FCRC0012/TRANSFERT COMMANDES HP/
EDITION 4 ENCODED BY DE GROOTE JAN AT 08.44.42 ON 05 SEP 1989
STATUS PCH
PROTECTION: NONE
ALIAS
  GEBRUIKER TRANS-ORDR-HP

```

NEDERLANDS TRANSFERT BESTELLINGEN HP
 IMS FCHPORDR
 FRANCAIS TRANSFERT COMMANDES HP
 UTILISATEUR TRANS-ORDR-HP
 CATALOGUED AS
 TEST
 DESCRIPTION
 NEDERLANDS:
 Record beschrijving, die gebruikt wordt om de bestellingen
 van de agentschappen et transfereren naar het systeem
 CRESUS (HP)
 THIS MEMBER IS DIRECTLY REFERRED TO 2 TIMES
 THIS MEMBER CONTAINS 8 DIRECT REFERENCES
 RECORD FCRC0012/TRANSFERT COMMANDES HP/
 FORM DEFAULTED-AS
 CONTAINS
 AGGREGATE FCAG0020/NUMERO UNIQUE POUR UNE
 FIELD FCFL0003/CODE ALPHABETIQUE POUR
 FIELD FCFL0026/INDICATION SI FRANC BEL
 AGGREGATE FCAG0003/DEFINITION UNIQUE D'UNE
 FIELD FCFL0028/NOMBRE DE COUPURES
 FIELD FCFL0029/FLAG BANCONTACT
 FIELD FCFL0133/FLAG ADAPT/FLAG ADAPT
 FIELD FCFL0018/DATE DE LA CREATION D'U

2.5 ENSEMBLE DES 'MODELS' DE DOCUMENTATION DISPONIBLES

a. Les modèles de types de membres de FUDOS

US : USER	FU : FUNCTION	ME : MESSAGE
ST : SECTOR	SH : SCHEMA	ER : ERROR
SY : SYSTEM	SS : SUBSCHEMA	AG : AGGREGATE
PR : PROCESS	DF : DATAFLOW	RU : RULE
AC : ACTIVITY	RF : REFERENCE	EL : ELEMENT
TK : TASK	OB : OBJECT	PJ : PROJECT
TR : TRANSACTION		

b. Les modèles de types de membres de TEDOS

AP : APPLICATION	SC : SCREEN	RC : RECORD
PG : PROGRAM	RP : REPORT	GR : GROUP
RO : ROUTINE	FI : FILE	FL : FIELD

c. Les modèles de types de membres de PRODOS

CH : CHAIN	JB : JOB
------------	----------

d. Les modèles de types de membres de IMS

GS : GSAM db	SHIS : S.HISAM db	IN : SEC.INDEX db
HD : HDAM db	HS : HSAM db	PCST : STRUCT. PCB
HID: HIDAM db	PCOU : OUTPUT PCB	SGIN : INDEX-PONTER sg
PCGS : GSAM PCB	SGLO : LOGICAL sg	MSDB : FP.-MSDB
SGPH : Physical sg	SHS : S.HSAM db	DEDB : FP.-DEDB
HIS : HISAM db	LO : LOGICAL db	

e. Les modèles de types de membres de DB2

DB2RD : DATABASE	DB2ST : STORAGE-GROUP	DB2VL : VOLUME
DB2IX : INDEX	DB2TB : TABLE	DB2VW : VIEW
DB2PS : PARTITION	DB2TS : TABLESPACE	

f. Les modèles de types de membres de PRODUCTION

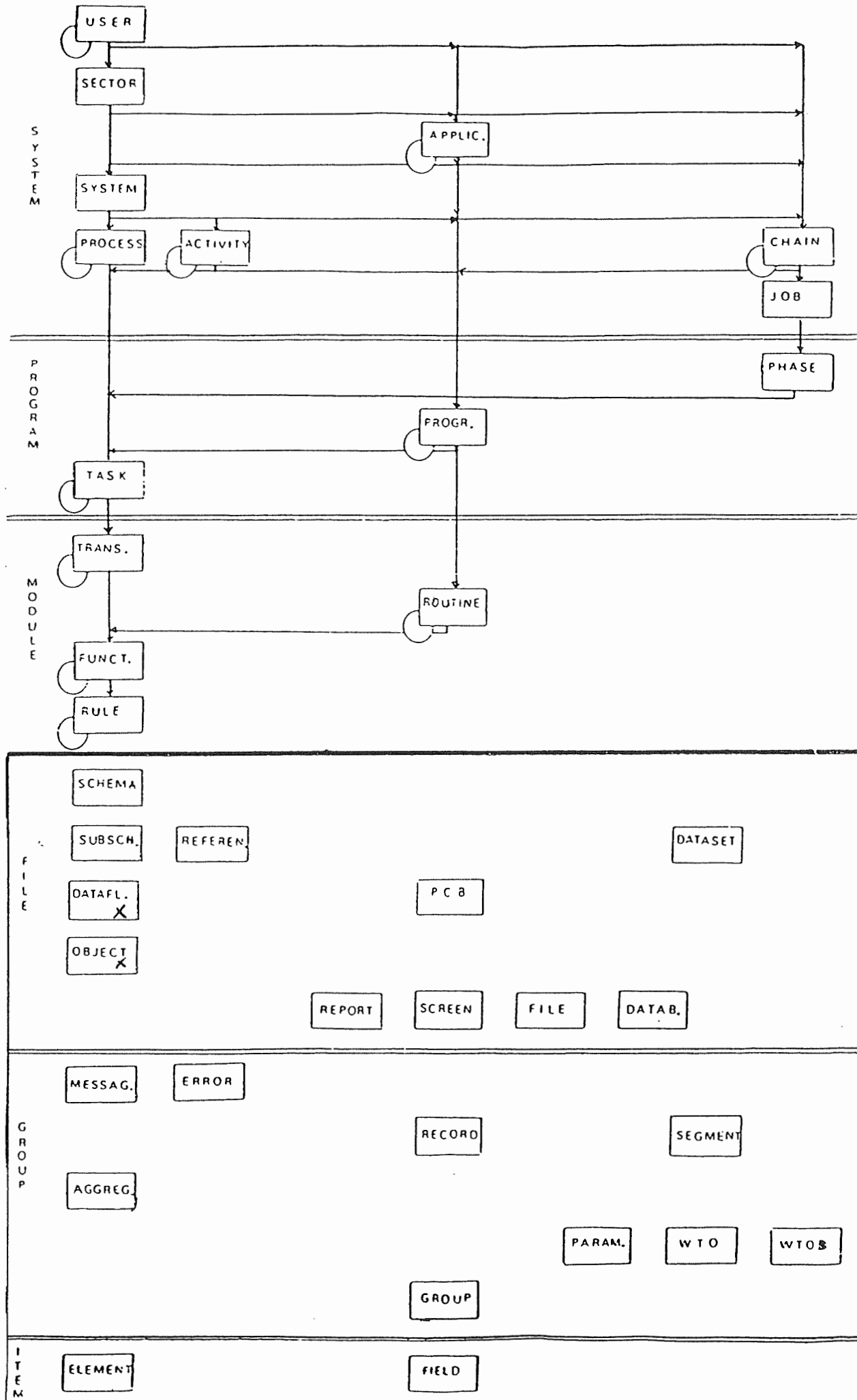
PDAP : APPLICATION	PDGR : GROUP	PDFL : FIELD
PDPG : PROGRAM	PDFI : FILE	PDCH : CHAIN
PDRO : ROUTINE	PDRP : REPORT	PDJB : JOB
Pddb : DATABASE		

2.6 SCHEMA HIERARCHIQUE DES TYPES DE MEMBRES

- via les relations CALLS
- via les relations CONTAINS

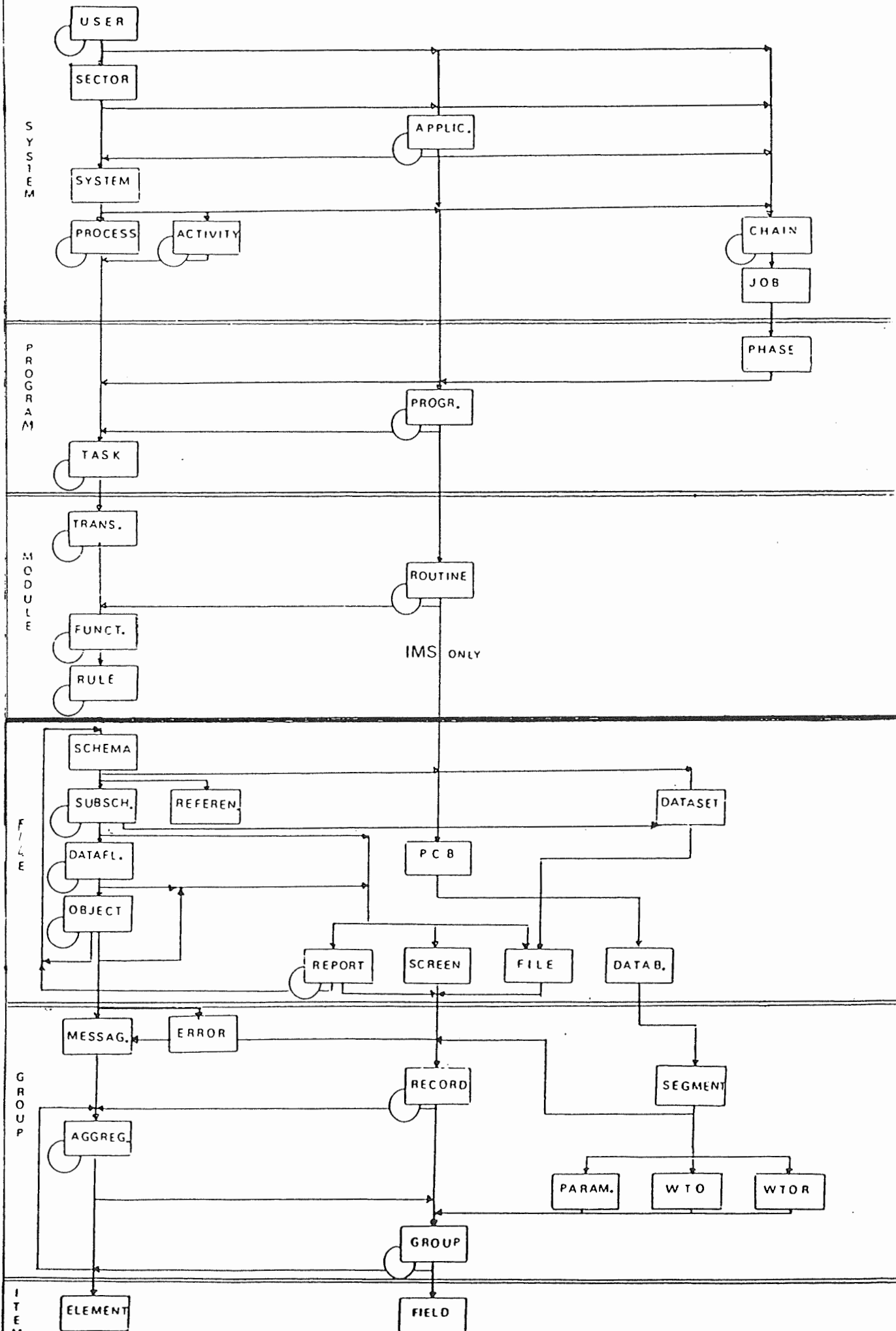
DATAMANAGER

RELATIONS CALLS



DATAMANAGER

RELATIONS CONTAINS



2.7 QUELQUES RUBRIQUES ESSENTIELLES

2.7.1 Le member-name

Chaque membre documenté dans le dictionnaire est une occurrence de type de membre.

Cette occurrence est identifiée par un 'member-name'.

Structure conventionnelle : xxttnnnn

xx : code application : AV assurances vie
UR universele rekening

tt : type de membre : OB objet
JB job

nnnn : identifiant unique : alphanumérique
AV PG T054
UR OB 5044

2.7.2 Alias

Il existe 11 types d'alias prédéfinis :
chacun permet de retrouver les membres décrits selon toute convention compréhensible des utilisateurs.

2.7.3 Catalog

permet un regroupement logique des membres qui sont 'catalogués' sous un même intitulé, que l'on peut considérer comme une clé de recherche.

2.7.4 Contains - calls

permet de créer les relations logiques entre les membres d'une façon descendante et / ou collatérale, en respectant l'organisation hiérarchique du dictionnaire :

- contains : aspect statique
- calls : aspect dynamique

exemple : une TASK 'contains' des TRANSACTIONS
une TRANSACTION 'calls' une FONCTION.

2.7.5 See

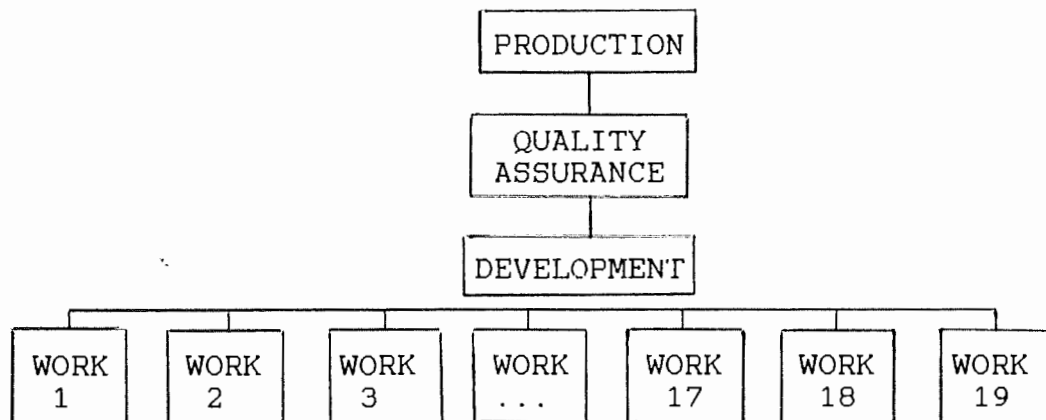
permet de faire référence à un autre membre d'un même type ou d'un type différent.

3. PARTITION DU DICTIONNAIRE - LES STATUS

3.1 GENERALITES

Le DMR est actuellement divisé en 23 'status' différents répartis en 4 niveaux.
Ces niveaux correspondent au cycle de vie de l'application documentée:

WORK	: 19x	read / update	phase de développement en cours
DEV	: 1x	read / update	phase de développement stabilisé
QUALITY ASSURANCE	: 1x	read only	phase des tests d'acceptation utilisateurs (operability requirements)
PRODUCTION	: 1x	read only	phase de mise en production



3.2 STATUS WORK

les 19 status 'Work' représentent différents domaines de développement qui, chacun, comprennent un ensemble de projets manipulant des données semblables.

Ils sont séparés les uns des autres de telle sorte que les commandes qui modifient le contenu des membres de DMR agissent seulement dans le status dans lequel on se trouve -le 'current' status -.

Certaines commandes d'interrogation agissent dans le status courant et dans les status qui lui sont supérieurs.

3.3 STATUS DEV

accessible aux 19 status inférieurs, il permet d'utiliser la documentation déjà définie dans d'autres applications et transférée en développement (il faut manipuler avec prudence la documentation appartenant à d'autres applications).

3.4 STATUS QA

prévu pour les tests utilisateurs après les phases de développement.

3.5 STATUS PRODUCTION

utilisation par les applications mises en production des membres qui y sont transférés.

4. LES COMMANDES D'INTERROGATION SIMPLE

4.1 GENERALITES

Ces commandes permettent d'afficher des listes de membres ou le contenu de ceux-ci.

4.2 LISTE ET BUT DES COMMANDES

<u>nom</u>	<u>but</u>
LIST ONLY	obtenir la 'liste' des seuls membres dont le nom commence par ...
LIST FROM a TO b	obtenir la 'liste' des membres dont la valeur du nom se situe 'depuis a jusqu' à b'.
LIST MEMBER	obtenir la 'liste des membres' spécifiés.
LIST ONLY XY WHEN ... TO ... EQ ... NE	obtenir la liste de tous les membres où les positions n à m du member-name sont égales / non égales à la valeur précisée.
PRINT	afficher l'enregistrement du membre choisi tel qu'il existe dans le fichier SOURCES.
REPORT	afficher la version ENCODEE du membre choisi.

GLOSSARY

obtenir, à partir de la liste des membres sélectionnés, une liste de ceux-ci contenant seulement les rubriques précisées dans la commande.

5. LES COMMANDES D'INTERROGATION COMPLEXE

5.1 GENERALITES

Ces commandes permettent d'obtenir la liste des membres répondant à des critères de sélection particuliers ou complexes

5.2 LISTE ET BUT DES COMMANDES

<u>nom</u>	<u>but</u>
WHAT IS	qu'est-ce que ce qui suit la commande?
<u>Remarque</u> : syntaxe : WHAT IS 'abcdefgh' où abcdefgh devrait être un alias, un catalog ou un nom de membre.	
WHAT USES	quels sont tous les membres qui utilisent le membre indiqué : - directement et indirectement - dans le sens ascendant - quel que soit le type de membre.
<u>Remarque</u> : cette commande utilise les références placées dans les rubriques CONTAINS, CALLS et SEE.	
WHAT USES ... DIRECTLY	quels sont tous les membres qui utilisent directement le membre indiqué : - dans le sens ascendant - quel que soit le type de membre. La recherche s'arrête au premier niveau de référence.
WHAT CONSTITUTES	quels sont tous les membres qui constituent des références pour le membre indiqué : - directement et indirectement, - dans le sens descendant, - quel que soit le type de membre.
<u>Remarque</u> : cette commande utilise les références placées dans les rubriques CONTAINS et CALLS	
WHICH ... USES ...	quels sont tous les membres du type choisi qui utilisent le membre indiqué dans le sens ascendant?

Remarque: cette commande utilise les références placées dans les rubriques CONTAINS, CALLS et SEE.

WHICH ...	quels sont tous les membres du type choisi
CONSTITUTES ...	si qui constituent une référence pour le membre indiqué dans le sens ascendant?

Remarque: cette commande utilise les références placées dans les rubriques CONTAINS et CALLS.

WHICH ...	quels sont tous les membres du type choisi
CONSTITUTES ...	si qui constituent une référence directe
DIRECTLY	pour le membre indiqué?

Remarque: cette commande utilise les références placées dans les rubriques CONTAINS et CALLS.

WHICH ... HAVE	quels sont tous les membres du type choisi
... SPECIFIED	si qui ont dans leur description la rubrique spécifiée?

WHAT FORMS	quels sont tous les membres qui ont comme mot catalogue celui qui est indiqué dans la commande?
------------	---

WHOSE ALIAS IS ...	quels sont les noms de membre dont l'alias est celui précisé dans la commande?
--------------------	--

WHOSE ALIAS ...	quel est ou quels sont les noms de membre
IS ...	dont l'alias du type indiqué est l' alias précisé dans la commande?

6. LES COMMANDES DE MANIPULATION

6.1 GENERALITES

Ces commandes permettent de modifier le contenu du dictionnaire au niveau de ses membres, et ce, uniquement dans le status dans lequel on se trouve.

6.2 LISTE ET BUT DES COMMANDES

<u>nom</u>	<u>but</u>
ADD	ajouter un nouveau membre dans le dictionnaire, dans le fichier SOURCES comme dans le fichier ENCODES.
MODIFY	modifier la documentation d 'un membre à partir du contenu du fichier des sources

et encoder la nouvelle source ainsi créée

Remarque: si la source à modifier est dans un status de niveau supérieur, la commande MODIFY génère automatiquement une copie de cette source dans le status WORK dans lequel on se trouve : la version modifiée de cette source est donc créée et encodée dans ce status. Dès ce moment, il y a deux versions de ce membre, dans des status hiérarchiquement différents.

REMOVE enlever le membre précisé du status dans lequel on se trouve.

Remarque: la commande est refusée si :

- le membre se trouve aussi dans un status de niveau supérieur.
- le membre est référencé dans un autre membre du même status

REVERT enlever un membre seulement du status dans lequel on se trouve (une autre version dans un autre statut n'est pas affectée).

Remarque: la commande est refusée si le membre est référencé dans un autre membre du même status.

REPLACE remplacer la source d'un membre par une nouvelle source de ce membre, et encoder cette nouvelle source.

REINSTATE rétablir la source qui correspond au membre encodé spécifié dans la commande.

Remarque: cette commande n'est utile que si la source actuellement présente diffère de celle qui a permis d'encoder le membre.

COPY copier un membre à partir de n'importe quel status en créant une source de ce membre pour le status dans lequel on se trouve.

Remarque: Il faut veiller à l'unicité du nom de membre dans l'ensemble des status WORK et la nouvelle source ainsi créée n'est pas encodée automatiquement.

ENCODE encoder un membre dont la description se trouve dans le fichier SOURCES.

RENAME ... AS ... changer le nom de membre d'un membre encodé.

Remarque: la commande est exécutée si le membre à renommer n'est pas référencé dans un autre membre de niveau supérieur.

7. LES COMMANDES DE CREATION DE LISTES 'KEPT DATA'

7.1 GENERALITES

La commande KEEP sert à mémoriser une liste des membres sélectionnés par la commande qui la suit.

La liste peut avoir un nom ou ne pas en avoir.

La liste ainsi créée peut ensuite être manipulée ou traitée par une autre commande : il suffit d'ajouter la clause KEPT à cette nouvelle commande pour que celle-ci traite les seuls membres de la liste ainsi créée.

La liste est gardée en mémoire temporairement pendant la durée de la session, sauf si une autre commande KEEP est introduite pour une liste de même nom.

7.2 LISTE ET BUT DES COMMANDES

nom

but

KEEP... créer une liste de membres sans nom.

Remarque : effet d'un nouvel usage de la commande :
destruction de la liste précédente.

KEEP IN ... créer une liste de membres dont le nom
est ...

Remarque : si une deuxième liste portant le même nom est créée, la précédente est détruite.

Il y aura autant de listes différentes gardées en mémoire qu'il y aura de noms différents.

ALSO KEEP ... ajouter les membres sélectionnés par la
commande qui suit, dans une liste déjà
existante avec ou sans nom.

DROP retirer d'une liste existante des membres
répondant à la commande qui suit.

* *

*

ANNEXE 6

**Description de l'outil de planning
Superproject**

CONTENU:

1. PRESENTATION GENERALE
2. QU'EST-CE-QUE LA GESTION DE PROJET?
 - 2.1 PLANIFICATION
 - 2.2 GESTION
 - 2.3 EVALUATION
3. LES TYPES DE DONNEES
 - 3.1 PROJETS
 - 3.2 TACHES
 - 3.3 RESSOURCES
 - 3.4 MISSION
4. FONCTIONNEMENT DE SUPERPROJECT
 - 4.1 CONSTRUIRE UN PROJET
 - 4.2 ECRANS DE CONSTRUCTION
 - 4.3 CHEMINS CRITIQUES ET NON CRITIQUES

Sources: " Superproject Expert version 1.0 ", (1988) manuel d'utilisation accompagnant le logiciel commercialisé par la firme Computer Associates.

Superproject Expert est un outil de planning commercialisé par la firme Computer Associates et qui fonctionne avec PC-DOS. Un manuel très précis mais très volumineux accompagne le programme. Cependant, ce manuel est très pratique et insiste énormément sur les détails de fonctionnement en omettant une présentation plus large et complète des avantages et des possibilités de l'outil. D'autre part, aucun manuel particulier n'a été réalisé à la CGER, ce a rendu la réalisation de cette annexe très délicate.

1. PRESENTATION GENERALE

Superproject Expert offre des techniques de gestion de projet sophistiquées - PERT, Gantt Charts, Work Breakdown Structure et la méthode du chemin critique - , afin de simplifier la planification, l'analyse des coûts et la mise à jour de projets, qu'ils soient simples ou complexes.

La figure suivante nous montre les principales vues du projet offertes par Superproject:

Nom	Contenu
PERT Chart	Réseau de séquençement des tâches.
Outline	Résumé des informations générales.
Task Gantt	Graphique des durées selon le temps.
Work Breakdown Chart	Structure des tâches et réseau hiérarchique; niveaux de tâches.
Details of Project	Informations générales sur la planification du projet et la définition des coûts.
Task Details	Plan, mission et information sur les coûts pour chaque tâche.
Resource Details	Plan, mission et information sur les coûts pour chaque ressource.
Calendars	Projet et planification personnelle.

2. QU'EST-CE-QUE LA GESTION DE PROJET?

La gestion de projet est un processus de

- ° Planification de projet
- ° Gestion de ses activités
- ° Evaluation des résultats

2.1 PLANIFICATION

Planifier un projet signifie définir les objectifs et développer une stratégie pour atteindre ces objectifs.

La phase de planification est constituée de deux parties: la planification stratégique et la planification tactique.

- **la planification stratégique** définit les objectifs, évalue les problèmes globaux tels les marges de profit, les priorités de l'entreprise, les produits compétitifs, l'impact du marché à long terme, les caractéristiques de l'entreprise, ... De cela, on développera un plan détaillé.
- **la planification tactique** est le plan détaillé. Ici, les détails se concentrent sur QUI fait QUOI pour COMBIEN DE TEMPS, DANS QUEL ORDRE et COMBIEN CA COUTERA. Ces détails servent de briques de construction du projet.

Superproject inclut différents écrans pour présenter le projet selon plusieurs perspectives différentes. Par exemple, on peut se concentrer sur les jalons de base du projet présentés sur un écran, et visionner les détails d'une activité spécifique sur un autre écran.

2.2 GESTION

Les activités de gestion de projet consistent à diriger les efforts du personnel responsable des activités prises individuellement. Cela revient à rouler au milieu de barrages routiers inévitables. La grande difficulté est d'évoluer autour de ces barrages le plus efficacement possible. Il peut y avoir diverses manières d'éviter un retard ou un conflit et c'est au gestionnaire de décider de la meilleure solution.

Superproject vous permet de choisir parmi une variété de solutions; il vous aide à choisir la meilleure.

En mettant en évidence les employés auxquels aucune tâche n'est assignée ainsi que les ressources disponibles, le programme vous aide à profiler le plan et à optimiser les charges de travail.

Durant la progression du travail, il est possible de modifier le plan et de signaler au programme qu'elles sont les tâches réalisées et celles qui sont en retard. Cette planification est dynamique - on introduit une date et le programme 'replanifie' immédiatement les activités concernées. Cela permet de toujours travailler avec un plan à jour.

2.3 EVALUATION

Evaluer un projet signifie la révision du chemin suivi par le projet terminé et l'évaluation de l'opportunité et de l'efficacité des décisions qui ont été prises.

Superproject possède des caractéristiques qui aident à revoir les estimations et plans qui ont été faits. Le programme permet de comparer les performances actuelles avec le plan de départ et permet de visualiser les retards de réalisation de certaines tâches, les journées surestimées,...

Ces données permettront de mieux estimer la réalisation de projets similaires dans le futur.

3. LES TYPES DE DONNEES

Les écrans de Superproject fournissent chacun une vue différente du projet.

Pris ensembles, ils offrent l'information suivante répartie en quatre sections principales: Projets, Tâches, Ressources et Missions.

3.1 PROJETS

Les données sur le projet apparaissent sur les écrans 'Details of project' et 'Project Calendar'.

- le nom du projet et sa description
- les congés pour le projet
- les préférences pour le projet. Ceci détermine le niveau de détail à afficher, l'étendue du projet,...
- les noms des projets liés (projets ayant des ressources communes). Ils permettent de réaliser le partage des ressources et d'éviter les conflits.

3.2 TACHES

Les données concernant les tâches apparaissent sur les écrans

'PERT Chart', 'Work Breakdown Chart', 'Outline', la partie supérieure de l'écran 'Task Details' et l'écran 'Task Gantt'.

- le nom de la tâche et sa description
- les estimations de dates
- les liens entre les tâches
- les délais
- les sous-projets
- les informations concernant les coûts
- dates maximales de début et de fin
- dates actuelles de début et de fin
- priorités des tâches

3.3 RESSOURCES

Les données concernant les ressources apparaissent sur la partie supérieure de l'écran 'Resource details', 'histogramme' et 'Resource Calendar'.

- le nom de la ressource et sa description
- la méthode d'allocation des ressources (24h/24h, nombre d'heures ou pourcentage de la journée de travail)
- journée de travail (allocation maximale avant conflit)
- weekends
- journées d'exception pour les ressources (jours fériés,...)
- coût horaire des ressources pour la facturation

3.4 MISSION

Il n'y a pas de nom de mission explicite. La mission est plutôt indiquée par le nom de la tâche ou de la ressource, si celui-ci est approprié.

- mission indiquée par la tâche ou la ressource
- numéro d'identification de la ressource, numéro WBS et noms
- priorité
- heures
- coûts
- type d'allocation (nombres d'heures fixé ou temps que ça prendra)

4. FONCTIONNEMENT DE SUPERPROJECT

Superproject peut fonctionner selon trois modes différents qui prennent compte de l'expérience de l'utilisateur: mode débutant, mode intermédiaire et mode expert, et il est aisé de passer d'un mode à l'autre.

4.1 CONSTRUIRE UN PROJET

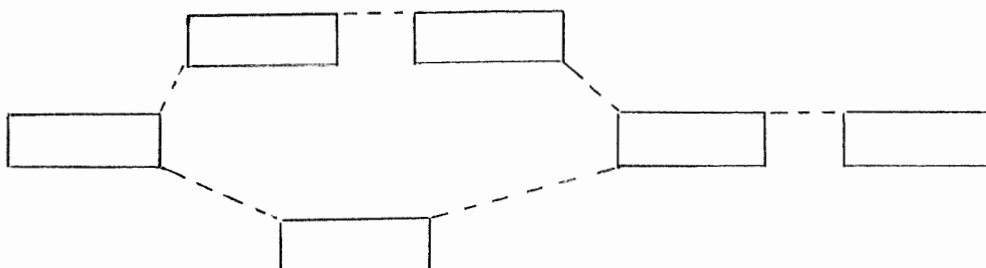
Superproject peut être utilisé de deux manières:

- de façon statique- comme outil de planning uniquement
- de façon dynamique - comme outil de planning, d'estimation et de 'replanification'

4.2 ECRANS DE CONSTRUCTION

Superproject offre quatre écrans principaux pour la construction de projet: le 'PERT Chart', le 'Outline', le 'Task Gantt' et le 'Work Breakdown Structure'.

- Le **PERT Chart** permet de construire un diagramme de flux des activités du projet. Ces activités, ou tâches, sont présentées comme des boîtes que l'on peut disposer dans un ordre séquentiel. Ceci permet de voir quelles tâches doivent être réalisées en séquence et quelles tâches occupent des places concurrentes.



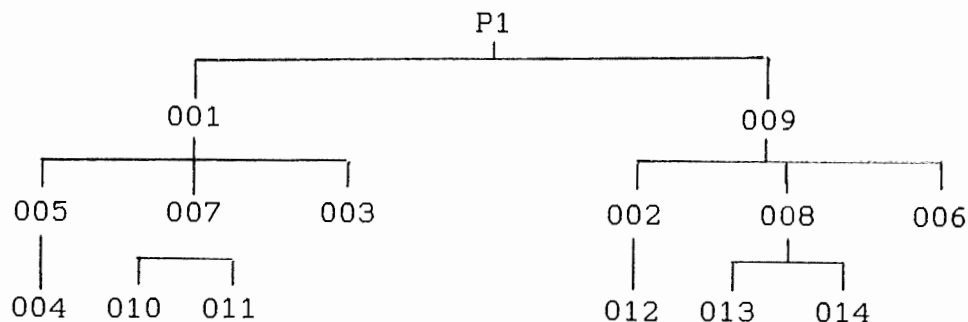
- Le **Outline** représente les tâches comme des entrées dans une table. Cet écran est utile pour introduire un grand volume de données de façon statique. C'est particulièrement utile lorsqu'on établit la structure du projet, le regroupement des tâches selon les phases du projet:

Heading/Task Resource ID	Task	Pr	Dur	Dev	Schd Start	Schd Finish
SALE.PJ	P1		19	0.00	05-27-87	06-22-87
Window Displays	008		5	0.00	05-27-87	06-02-87
Clothing	001		4	0.00	05-27-87	06-01-87
Lighting	002		3	0.00	06-02-87	06-04-87
Backdrop	003		2	0.00	06-12-87	06-15-87
Display Cases	007		5	0.00	05-27-87	06-02-87
Accessories	004		8	0.00	06-02-87	06-11-87

- Le **Task Gantt** fournit un histogramme montrant les durées des tâches sur le temps et les ressources qui leur sont allouées

1 Day Per Symbol			August 87				
ID	Heading/Task	Resource	27	03	10	17	24
P1	ACTPROJ.PJ						
001	Task-1						
001		Rsrc-1					
001		Rsrc-2					

- Le **Work Breakdown Chart** montre de façon graphique la structure du projet dans laquelle les tâches sont groupées sous la direction d'une autre tâche, sous la forme d'un graphe d'organisation. Ce graphe nous montre les différents niveaux du projet:



On préférera l'écran 'Outline' pour introduire les tâches et ajuster la structure du projet, on utilisera les autres pour revoir les dépendances entre les tâches (PERT Chart), l'estimation des tâches (Task Gantt) et la structure du projet (WBS Chart).

4.3 CHEMINS CRITIQUES ET NON CRITIQUES

Une fois les tâches créées, reliées selon l'ordre logique et leurs durées introduites, il est possible de visionner une estimation préliminaire présentée par le diagramme PERT. Sur cet écran, on distingue le chemin critique qui est le plus long chemin entre le début et la fin du projet. Ce chemin est important puisqu'il permet d'estimer la date de fin de projet.

ANNEXE 7

Description de l'Upper Case Tools
Excelerator

CONTENU:

1. PRESENTATION
2. TECHNIQUES DE MODELISATION
3. LES ETAPES DE DEVELOPPEMENT ET LE ROLE D'EXCELERATOR
 - 3.1 DEFINITION DES LIMITES DU SYSTEME
 - 3.2 IDENTIFICATION DES ENTITES EXTERNES QUI SATISFONT LES EXIGENCES
 - 3.3 DEFINITION DES FLUX D'INFORMATION AVEC LE SYSTEME
 - 3.4 DEFINITION DES PRINCIPAUX SOUS-SYSTEMES
 - 3.5 DEFINITION DES BESOINS EN DONNEES DU SYSTEME
 - 3.6 DEFINITION DES PROCESSUS DU SYSTEME
 - 3.7 VERIFICATION DE LA CONSISTANCE DU MODELE DES PROCESSUS
 - 3.8 INTEGRATION DES MODELES DES DONNEES ET DES PROCESSUS
 - 3.9 ASSURANCE QUE LE SYSTEME SATISFAIT AUX EXIGENCES
 - 3.10 A TRAVERS LA CONCEPTION PHYSIQUE
 - 3.10.1 Définitions des inputs et outputs du système
 - 3.10.2 Génération du code des designs d'écrans et de records
4. SYNTHESE DE L'UTILISATION DE EXCELERATOR DANS SDM
 - 4.1 AU NIVEAU DE L'ANALYSE CONCEPTUELLE
 - 4.2 AU NIVEAU DE L'ANALYSE FONCTIONNELLE

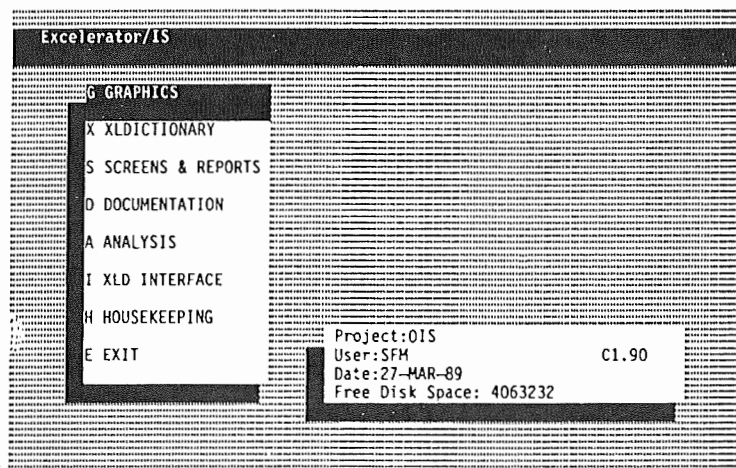
Source: "Excelerator/IS, application guide, release 1.9 ", (1990) guide d'application de Excelerator fournit avec le logiciel par la firme Index Technology.

1. PRESENTATION

EXCELERATOR/IS est un ensemble intégré d'outils d'analyse et de design, commercialisé par la firme Index Technology. Il permet aux analystes de définir, vérifier et documenter une conception de système d'information avant d'en entamer le codage.

EXCELERATOR est un CASE (Computed-Aided Software Engineering) tool qui permet d'automatiser des tâches dès les premières phases du cycle de vie du développement d'un système : les phases d'étude de définitions, d'analyse conceptuelle et d'analyse fonctionnelle.

Il permet de créer des **modèles graphiques**, de décrire leur contenu dans un dictionnaire de données centralisé et ensuite d'analyser et de documenter le modèle du système, comme le montre la figure suivante:



Les options du menu principal sont utilisées comme suit:

Graphics : elle permet de créer et de mettre à jour des représentations visuelles du système, ses composants et les relations existant entre eux. Vu que la modification des graphes peut se faire aisément, Excelerator supporte une approche itérative de l'analyse et du design d'un système.

XLdictionary : elle permet de définir le système ainsi que des rapports sur les données des spécifications. Etant donné que XLdictionary fournit un stockage central pour tout système d'information, il aide à assurer la consistance des spécifications.

Screens & Reports : elle permet de développer des prototypes des écrans d'input du système et des rapports d'output.

Documentation : elle permet de produire des copies sur papier de chaque aspect du système, organisés selon une ligne de conduite que l'on aura fixée.

Analysis : elle permet de s'assurer que les données sont consistantes et correctes en produisant des rapports pour examiner les données et vérifier leur adhérence aux techniques standards. On peut ainsi créer ses propres listes et rapports de données de spécification.

XLDinterface : elle permet de protéger et de partager les données créées de manière à ce que les analystes puissent travailler ensemble sur un large projet.

Housekeeping : elle fournit des fonctions pour établir et réaliser la maintenance des projets et le hardware.

2. TECHNIQUES DE MODELISATION

Les deux grandes techniques de modélisation utilisées par Excelsator sont: le diagramme de flux de données et le diagramme de modèle des données.

- le diagramme de flux de données : il sert à modéliser les tâches que le système doit accomplir.
- le diagramme de modèle de données : il illustre les associations entre les différentes catégories de données à l'intérieur du système. Il sert de librairie de toutes les définitions de données à utiliser avec le système à développer.

Ces deux types de graphe représentent les deux vues d'un système: la **vue orientée processus** et la **vue orientée données**. Lorsque les deux vues sont entièrement développées, définies et intégrées, elles fournissent une spécification logique complète du système. Ces spécifications deviennent la base pour l'implémentation du système.

3. LES ETAPES DE DEVELOPPEMENT ET LE ROLE D'EXCELEATOR

On commence l'analyse du système en modélisant la vue orientée données et la vue orientée processus du système.

3.1 DEFINITION DES LIMITES DU SYSTEME

L'équipe de développement commence par réaliser le premier niveau de la vue orientée processus en définissant l'environnement du système. C'est-à-dire que le système est placé dans un contexte plus large composé des autres systèmes et des personnes qui interagissent avec lui. Pour décrire le rôle du

système dans cet ensemble, on réalise un **diagramme de contexte**. Ce diagramme décrit les personnes et les systèmes qui interagissent avec le système à développer tandis que le diagramme du système décrit les principales activités que devra accomplir le futur système.

Le diagramme de contexte du niveau le plus élevé est à l'origine de tous les diagrammes de flux du modèle des processus. Le diagramme du système situé un niveau en-dessous du diagramme de contexte établit le cadre pour modéliser les tâches détaillées du système sur des graphes de niveau inférieur.

On débute un diagramme de contexte par un seul processus qui représente l'entière du système à développer.

Le diagramme de contexte représente une vue de haut niveau du système, les niveaux inférieurs de la suite des diagrammes de flux de données montrent le fonctionnement du processus de contexte selon des niveaux de détail de plus en plus précis. Pour établir le lien du niveau de processus de contexte au niveau suivant dans la suite des diagrammes de flux de données, on spécifie un pas d'explosion sur l'écran de description de processus. La figure suivante illustre le cas d'une explosion vers un diagramme de flux de données :

Process	0.0
Label	ORDER/INVENTORY SYSTEM
Explodes To:	DFD-SYC-STO-PPS-PRO
Type DFD Name	0.0 ORDER/INVENTORY SYSTEM
Location	
Process Category	
Duration Value	
Duration Type	
Manual or Computer C	
	Page

3.2 IDENTIFICATION DES ENTITES EXTERNES QUI SATISFONT LES EXIGENCES

Après avoir décrit le système comme un processus central, on identifie les objets du monde réel qui interagissent avec lui. Dans la plupart des cas, on identifie ces entités externes en examinant les exigences du système.

3.3 DEFINITION DES FLUX D'INFORMATION AVEC LE SYSTEME

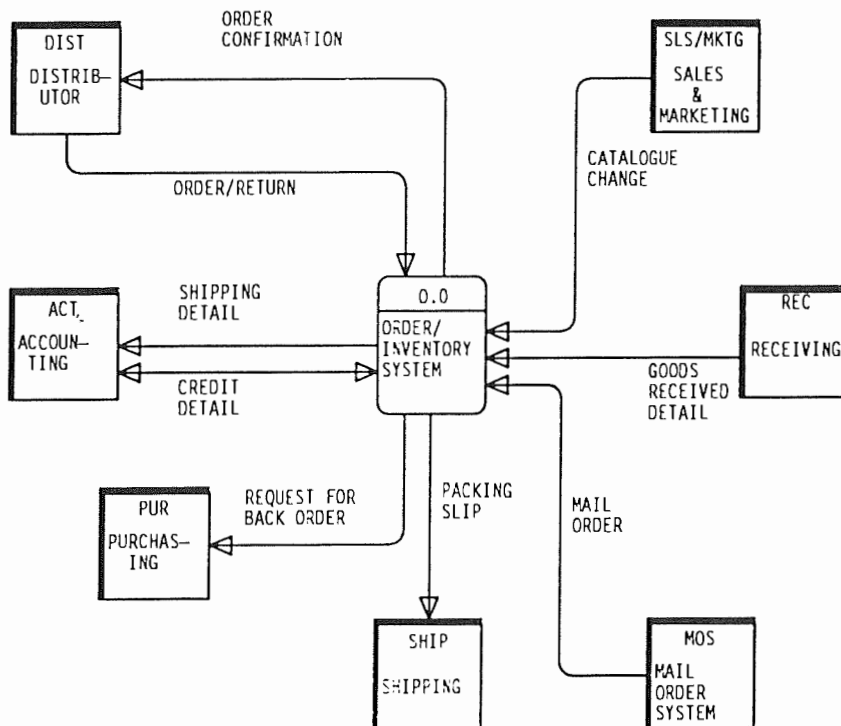
Les flux de données dans le diagramme de contexte montrent les mouvements des informations entre le monde extérieur et le système.

Excelsior permet de définir des pas d'explosion entre les composants des flux d'information et des records qui montrent le détail des flux.

Ces records peuvent contenir d'autres records qui, à leur tour, contiennent toujours d'autres records ou éléments.

De cette façon, on peut définir les contenus des flux de données selon des niveaux de détail de plus en plus grands.

Ceci termine la réalisation du diagramme de contexte dont un exemple est illustré par la figure suivante:

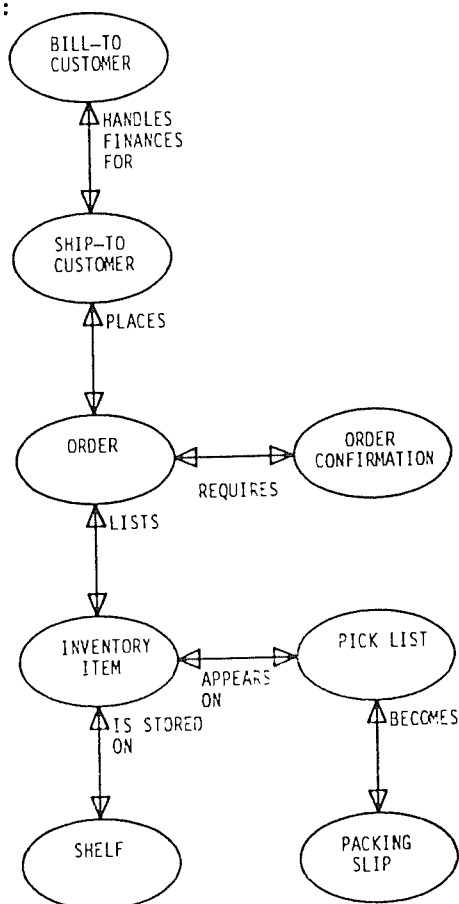


3.4 DEFINITION DES PRINCIPAUX SOUS-SYSTEMES

Une fois le rôle du système à développer défini dans une organisation plus large, on commence à diviser le système en sous-systèmes. On crée pour cela un diagramme de flux de données de haut niveau appelé le **diagramme du système**. Ce diagramme décrit chaque sous-système important comme un processus. Pour cela, on se base généralement sur les exigences du système.

3.5 DEFINITION DES BESOINS EN DONNEES DU SYSTEME

Une fois les processus de haut niveau définis, on développe le modèle des données. On identifie pour cela les données que le système utilise et produit et on trace ensuite un **diagramme de modèle de données** du système comme un ensemble d'entités de données reliées entre elles. La figure ci-dessous illustre un exemple:



Les entités définies servent de librairie des informations dont le système a besoin pour réaliser diverses opérations.

Chaque entité est décrite dans le XLdictionary et pour chaque entité, un pas d'explosion vers un record est spécifié. Le contenu de ce record représente les attributs du type d'entité.

Les figures ci-dessous présentent une description d'un type d'entité et d'un record dans le XLdictionary

[illegible]

Record

BILL-TO CUSTOMER

Alternate Name
Definition
Normalized

DISTRIBUTOR'S BILLABLE LOCATION

Name of Element or Record	Occ	Seq	Type	Sec-Keys
CUSTOMER NUMBER	1	0	C	
BILL-TO CUST NAME	1	0	F	
BILL-TO CUST STREET	1	0	F	
BILL-TO CUST CITY	1	0	F	
BILL-TO CUST STATE	1	0	F	
BILL-TO CUST ZIP CODE	1	0	F	
BILL-TO CUST TELEPHONE	1	0	F	
BILL-TO CUST CONTACT PERSON	1	0	F	
CUSTOMER CREDIT LIMIT	1	0	C	
CUSTOMER CURRENT ORDER BALANCE	1	0	F	
CUSTOMER ACCOUNT BALANCE	1	0	F	
CUSTOMER DISCOUNT RATE	1	0	C	

PgDn

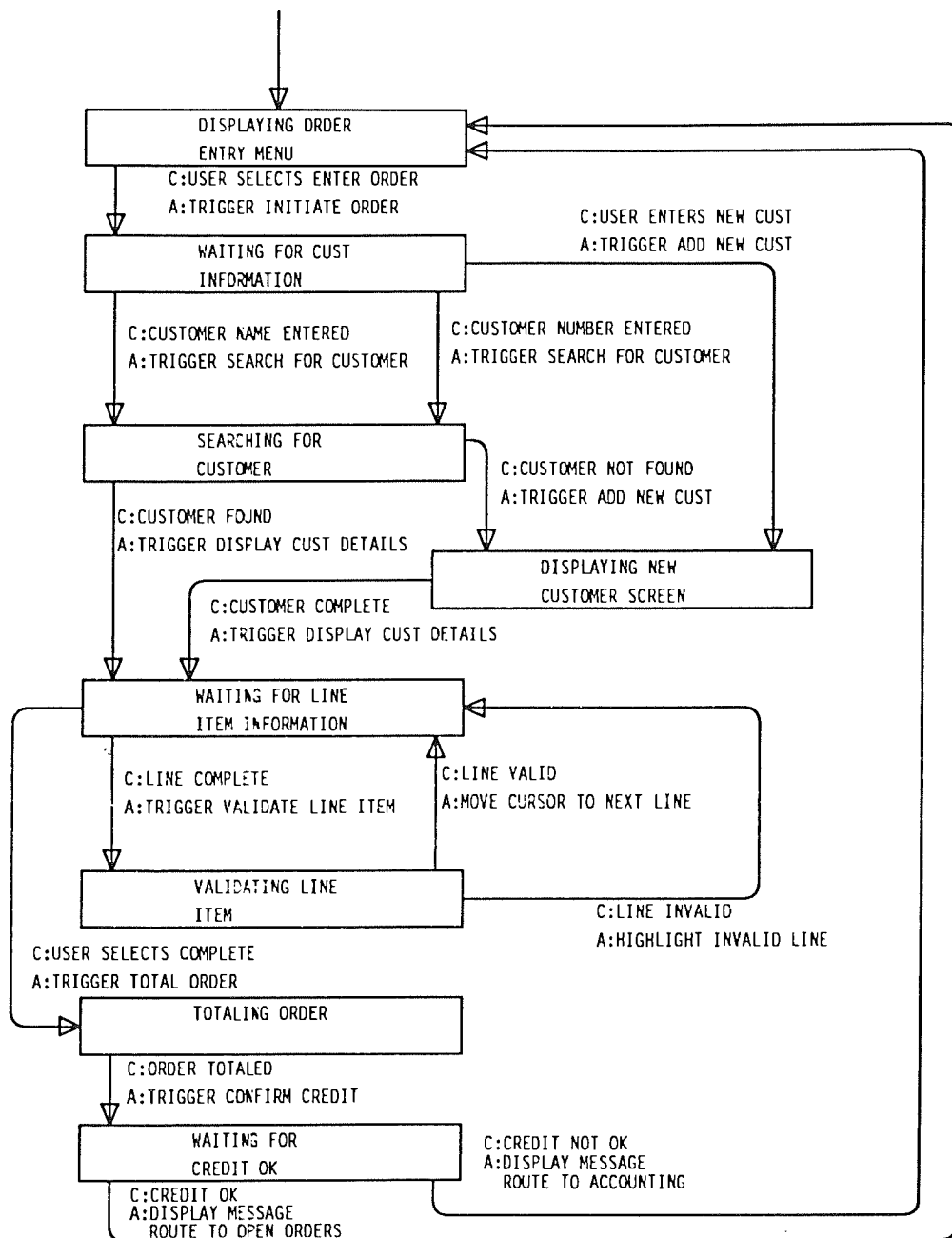
Primary key

3.6 DEFINITION DES PROCESSUS DU SYSTEME

Les principaux sous-systèmes et les données qu' ils utilisent étant définis, on va examiner les processus en détail. On explore chaque processus du diagramme du système vers un diagramme de flux de données d'un niveau plus bas et qui décrit le processus plus en détail. Ensuite, on explore chaque nouveau processus décrit dans ce diagramme vers un autre diagramme de flux de données de niveau inférieur et on répète cette procédure jusqu'à ce que les sous-processus atteignent un niveau de détail suffisant.

Si le système accomplit des tâches critiques, on modélise alors la séquence et le contrôle de fonctionnement du système

en créant un diagramme de transition d'état qu'on relie ensuite au diagramme de flux de données qui décrit la tâche critique. La figure ci-dessous illustre un exemple de diagramme de transition d'état:



3.7 VERIFICATION DE LA CONSISTANCE DU MODELE DES PROCESSUS

Une fois tous les diagrammes de flux de données tracés et complets, on vérifie que toutes les données représentées sur un graphe 'parent' sont reprises sur tous les graphes qui succèdent directement ce graphe. On réalise cela grâce au rapport 'Level balancing'. La figure ci-dessous illustre un exemple de rapport montrant qu'il existe deux éléments en output et deux éléments en input qui sont mentionnée dans le graphe parent mais qu'on ne retrouve pas dans les graphes suivants:

DATE: 31-MAR-89	LEVEL BALANCING	PAGE 1
TIME: 14:23		Excelerator/IS
PROJECT NAME: OIS		
LEVEL NUMBER: 1		
PARENT GRAPH NAME: 2.0 FULFILL ORDER		
Parent Process: 2.1		
Child Type: DFD Name: 2.1 GENERATE PICK LISTS		
Parent OUTPUTS not matched on child level		
TYPE	ID	CARRIED IN
ELEMENT	PICK LIST TOTAL LINES	DATA
ELEMENT	PICK LIST TYPE	DATA
		PICKER'S LIST
		PICKER'S LIST
Parent Process: 2.2		
Child Type: DFD Name: 2.2 PICK OUTGOING ITEMS		
Parent INPUTS not matched on child level		
TYPE	ID	CARRIED IN
ELEMENT	PICK LIST TOTAL LINES	DATA
ELEMENT	PICK LIST TYPE	DATA
		PICKER'S LIST
		PICKER'S LIST

3.8 INTEGRATION DES MODELES DES DONNEES ET DES PROCESSUS

Excelerator permet de déterminer les incohérences entre les modèles et aide à détecter les endroits où chaque modèle doit être modifié de manière à ce que les deux soient compatibles. Vu que le modèle des données sert de librairie de données pour le système que l'on définit, les données utilisées par les processus dans les diagrammes de flux de données du système doivent exister ou être dérivées des données du modèle des données.

Dans Excelerator, on peut utiliser des listes d'entités pour trouver les éléments qui sont utilisés dans les diagrammes de flux de données mais qui n'existent pas dans le modèle des

données. On crée une liste d'entités des éléments dans le modèle des données et une seconde liste des éléments dans le modèle des processus. Ensuite, on utilise le set des opérations agissant sur les listes d'entités pour comparer le con-

3.9 ASSURANCE QUE LE SYSTEME SATISFAIT AUX EXIGENCES

A l'aide des deux modèles intégrés, on vérifie que l'analyse satisfait toutes les exigences. On vérifie que les spécifications sont correctes et complètes.

Avant de commencer l'analyse, l'analyste a introduit un ensemble d'exigences dans le XLDictionary, en tant que 'User Requirements' et 'Engineer Requirements'. Ces exigences ont pu ensuite être reliées aux entités définies par la suite et susceptibles de répondre à ces exigences.

De la même manière on peut associer des 'Notes' et des 'Issues' aux entités de manière à identifier les problèmes ou clarifier ce que le système doit faire pour rencontrer les exigences.

On établit ce lien en complétant la région 'Associated entities' sur les écrans de description d'entités appropriées.

Une fois que l'on a stocké ce genre d'information, on peut utiliser Excelerator pour évaluer de quelle manière le système a rempli ses obligations. On peut vérifier que l'on a adressé tous les problèmes que l'on a rencontrés durant l'analyse ainsi que toutes les exigences définies.

On définit un problème lorsqu'on rencontre un problème durant l'analyse. Lorsqu'on a trouvé une solution, on entre "Resolved" dans le champ de statut du problème. Cette pratique permet de vérifier les problèmes non résolus.

On peut créer un rapport et imprimer les problèmes non résolus ainsi que les entités qui sont reliées à ce problème.

La figure suivante illustre un rapport concernant les problèmes non résolus d'un projet:

DATE: 4-APR-89	UNRESOLVED ISSUES	PAGE 1
TIME: 14:23		Excelerator/IS
TYPE ISS	NAME IS-DISTRIBUTOR ORDERS	
ISS IS-DISTRIBUTOR ORDERS	Has-Associated REF RS-5.1	
	Contained-In ELS UNRESOLVED ISSUES	
	Associated-With PRC 2.2	
	PRC 2.0	
	PRC 2.1	
	URQ ORDER SPEED REQUIREMENTS	
	URQ ORDER SIZE REQUIREMENTS	
	URQ ORDER LOAD REQUIREMENTS	

3.10 A TRAVERS LA CONCEPTION PHYSIQUE

Une fois le modèle logique du système terminé, il s'agit de créer le modèle physique.

Dans ce but, Excelerator offre deux fonctionnalités qui sont la définition des inputs et outputs du système ainsi que la génération de code pour les designs d'écrans et de records.

3.10.1 Définitions des inputs et outputs du système

Lorsqu' on a réalisé le modèle logique du système et qu'on a défini les données utilisées par le système ainsi que les fonctions qu'il réalise, on peut maintenant utiliser ces données pour créer des **prototypes** des écrans d'introduction de données.

Chaque élément compris dans les records devient un champ sur l'écran. Par la suite, il est possible de modifier les layouts des écrans en accord avec les utilisateurs.

Les figures suivantes montrent une description de record et l'écran généré par Excelerator à partir de cette définition:

The Records ACTUAL ITEMS PICKED and PICK LIST INFO

DATE: 4-APR-89	CONTENTS OF RECORDS: ACTUAL ITEMS PICKED	PAGE 1
TIME: 11:44	& PICK LIST INFO	Excelerator/IS
RECORD NAME	ELEMENT NAME	
ACTUAL ITEMS PICKED	PICK LIST NUMBER	
	ITEM NUMBER	
	SHELF NUMBER	
	QUANTITY PICKED	
PICK LIST INFO	PICK LIST NUMBER	
	PICK LIST DATE	
	PICKER ID	
	ITEM NUMBER	
	ORDER NUMBER	
	SHELF NUMBER	
	QUANTITY TO PICK	
	PICK LIST TOTAL LINES	
	PICK LIST TYPE	

```
POEM LIST RECONSTRUCTION
```

POEM LIST NUMBER [REDACTED] POEM LIST DATE [REDACTED]
[REDACTED] POEM LIST NAME
TOTAL LINES [REDACTED]

[REDACTED] POEM NO [REDACTED]

ITEM NUMBER	SHELF - CONNECTION	QUANTITY OF POEM	QUANTITY POINTS
[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]
[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]
[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]
[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]
[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]
[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]
[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]
[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]

HOURS MINUTES SECONDS HOURS MINUTES SECONDS HOURS MINUTES SECONDS

La conversion d'un record ou d'un design d'écran en une structure de données compilable que l'on peut incorporer ensuite et directement dans le programme du système, est réalisable en quatre langages: BASIC, COBOL, PL/1 et C.

[illegible]

```

*Record ITEMS-PICKED Generated: 4-APR-89
01 ITEMS-PICKED.
05 PL-PICK-LIST-NUMBER PIC 9(8).
05 PL-PICK-LIST-DATE PIC X(8)
VALUE '00/00/00'.
05 PL-PICK-LIST-TYPE PIC XX.
05 PL-TOTAL-LINES PIC XXX.
05 PL-PICKER-IO PIC X(4).
05 PL-PICK-LIST-LINES OCCURS 30 TIMES.
10 PL-ITEM-NUMBER PIC X(6).
10 PL-SHELF-NUMBER PIC XXXX
OCCURS 2 TIMES.
10 PL-QUANTITY-TO-PICK PIC S9(9)
COMP VALUE 000.
10 PL-QUANTITY-PICKED PIC S9(9)
COMP VALUE 000.

```

4. SYNTHESE DE L'UTILISATION DE EXCELERATOR DANS SDM

4.1 AU NIVEAU DE L'ANALYSE CONCEPTUELLE

Activité SDM		Diagrammes XL utilisés	Entités XL décrites	Rapports de documentation et d'analyse
n°	nom			
1.4	Définition des objectifs		User Requirements	Adéquation objectifs / entités
1.5	Points en suspens		Issues	Analyse de l'évolution des problèmes(1)
1.6.1	Modèle conceptuel des données	Diagrammes entité/ relation.	Data Entities Data Relationships Alias-Notes	Description des entités & associations Undescribed Graph Entities Entity-Relationship Validation Analysis Graph Explosion Report
1.6.2	Modèle conceptuel de gestion	Diagrammes de transition d'état	States Transition Vectors	Undescribed Graph Entities State Transition Diagram Verification
1.6.3	Modèle conceptuel des traitements	Diagrammes de flux de données	Process Data Stores External Entities Data Flows	Graph Explosion Report Data Flow Diagram Verification
1.9	Problèmes de conversion		Issues	Analyse de l'évolution des problèmes(1)

(1) L'analyse de l'évolution des problèmes peut se baser sur les dates charnières déterminées (ex: détection du problème, définition de la solution, implantation, mise en production).

4.2 AU NIVEAU DE L'ANALYSE FONCTIONNELLE

Activité SDM		Diagrammes XL utilisés	Entités XL décrites	Rapports de documentation et d'analyse
n°	nom			
2.4 à 2.6	Description des traitements	Data Flow Diagrams + explosion	Process Data Stores Data Flows	Graph Explosion Report Data Flow Diagram Verification
2.7 à 2.8	Problèmes de sécurité et d'organisation		Issues	Analyse de l'évolution des problèmes.
2.9	Structure logique des données	Record explosé du diagramme entité/relation	Records Elements Alias-notes	Record Content Analysys Key Validation Analysis Entity-Relationship Validation analysis Data Normalization Analysis

ANNEXE 8

**Présentation de la technologie de
l'AD/Cycle**

La philosophie d' AD/Cycle est présentée comme une volonté de transformer les outils indépendants supportant des technologies différentes en outils intégrés en vue de pouvoir réellement augmenter la productivité et la qualité du développement. Dès lors, il est intéressant de définir pour chaque étape du développement la direction à prendre pour réaliser le travail et d' opter pour certaines solutions proposées par la "platform". Ci-dessous se trouve un résumé(1) de ces différents points reliés à la méthodologie SDM applicable à la CGER.

1. PRINCIPES DE BASE DU FONCTIONNEMENT DES OUTILS AD/CYCLE (au 01/07/90)

Tableau Principes de base du fonctionnement des outils AD/Cycle (au 01/07/90).			
Phases AD/Cycle	Principes de base de la technologie des outils AD/Cycle	Solutions recommandées	Phases SDM concernées
1) Enterprise Modelling	- Création d'un modèle d'information - Définition des besoins généraux (Data & Process) - Intervention des gestionnaires non-informaticiens	- IBM DevelopMate (basé sur prototypage) - IEW Planning Workstation - PC Prism	0 et 1
2) Analysis & Design	- Utilisation des tools -> intégrant les principes de Software Engineering répandus et acceptés -> se basant sur des méthodes de design modernes. -> laissant la part large aux techniques graphiques.	- IEW Analysis Workstation - EXCELERATOR - Bachman Re-engineering	1 et 2
3) Generation	3 GL	- Protection des investissements - Intégration des outils classiques : éditeur, préprocesseurs, compilateurs, débogueurs statiques et dynamiques.	3 et 4
	4 GL	- Développement plus productif - La qualité du code produit doit égaler celle des 3 GL	
	5 GL (KBS)	- Nécessité d'intégrer les développements de type "Systèmes Experts" dans des applications classiques.	
4) Test & Maintenance	- Analyse de l'impact d'une modification sur le système.	- IBM Inspect for C/370 - IBM Inspect for PL/1 - Cobol Structuring Facility	5 et 7
Gestion de projet		ADPS	1 à 7

Signification des abréviations:

IEW : Information Engineering Workbench, produit de Knowledge Ware, Inc.
3GL : 3rd Generation Languages : langage de programmation de type symbolique et procédural (Ex.: COBOL, PL/1).
4GL : 4th Generation Languages : générateur de programmes de type procédural (Ex.: CSP, APS).
5GL : 5th Generation Language : générateur de programmes de type déclaratif (non-procédural) (Ex.: systèmes experts).

IRS : Integrated Reasoning Schell, logiciel KBS
KBS : Knowledge Based Systems : systèmes à base de
connaissance, "Systèmes experts"
ESE : Expert System Environment, logiciel KBS
ADPS: Application Development Project Support : logiciel de
gestion et de suivi de projet (IBM).

2. COMMENTAIRES

1. Un chevauchement apparaît au niveau des phases SDM entre les phases AD/Cycle 1 et 2. Il semble que la distinction entre la modélisation au niveau de l'entreprise et les outils d'analyse et de design se situe essentiellement au niveau de l'échelle de référence à laquelle on se situe pour réaliser ce travail. SDM ne fait pas de distinction entre la modélisation pour l'entreprise dans son ensemble et celle qui relève directement d'une application. Par contre, les activités de la phase d'analyse conceptuelle pourraient être parcourues pour modéliser l'ensemble des besoins en information de l'entreprise. Ensuite, ces besoins donneront naissance à différentes applications qui devront elles-mêmes être soumises à une analyse conceptuelle plus précise. Cette façon de procéder présente comme avantage principal d'assurer la cohérence entre les choix qui sont réalisés au niveau supérieur (Architecture des données de l'entreprise, AD/Cycle phase 1 et SDM phase 1) et ceux qui découlent de l'explosion du système d'information en ses différentes applications (Architecture des applications, AD/Cycle phase 2 et SDM phase 1). De plus, l'utilisation d'un seul outil de modélisation aux deux niveaux diminue le risque d'incohérence qui pourrait découler de ce travail en étapes.

Notons cependant que certaines fonctionnalités des outils Enterprise Modelling proposés ne figurent pas en Excelerator (ex : définition des systèmes informatiques dans une stratégie d'entreprise, étude des priorités, élaboration d'un planning global).

D'autres outils remplissent ces objectifs.

2. La phase d' Enterprise Modelling d'AD/Cycle peut être enrichie par un travail de type "Reverse Engineering" à partir de la situation existante en vue de confronter les deux résultats pour évoluer d'une manière itérative vers l'environnement intégré "idéal".
3. Remarquons sur le tableau ci-dessus la présence d' outils externes à IBM essentiellement dans les phases 1 et 2 tandis que les phases suivantes semblent surtout réservées aux logiciels IBM. La phase 3 (Générations) est plutôt décrite sous la forme de "promesses de développement futurs".