HAL
open science

Informatique pour les systèmes d'aide à la gestion du trafic : éléments pour un guide technique

Daniel Cholley, Patrick Gendre

To cite this version:

Daniel Cholley, Patrick Gendre. Informatique pour les systèmes d'aide à la gestion du trafic : éléments pour un guide technique. [Rapport de recherche] Centre d'études sur les réseaux, les transports, l'urbanisme et les constructions publiques (CERTU). 1999, 66 p., figures, tableaux, 31 références bibliographiques. hal-02162289

HAL Id: hal-02162289

https://hal-lara.archives-ouvertes.fr/hal-02162289v1

Submitted on 21 Jun 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

CERTU : Centre d'études sur les réseaux
les transports, l'urbanisme
et les constructions publiques

INFORMATIQUE POUR LES SYSTÈMES D'AIDE A LA GESTION DU TRAFIC

ELEMENTS POUR UN GUIDE TECHNIQUE

Rédacteurs ou coordonateurs

Daniel CHOLLEY (CETE de l'Est)

Patrick GENDRE (CERTU Département Systèmes)

Octobre 99

NOTICE ANALYTIQUE

Organisme commanditaire :

CERTU : Centre d'études sur les réseaux, les transports, l'urbanisme et les constructions publiques

9, rue Juliette Récamier 69006 Lyon Tel : 04 72 74 58 00 Fax : 04 72 74 59 00

Titre :

INFORMATIQUE POUR LES SYSTÈMES D'AIDE A LA GESTION DU TRAFIC

Sous-titre :	**Date d'achèvement :**	**Langue :**
Éléments pour un guide technique	**Octobre 99**	**Français**

Organismes auteurs	**Rédacteurs ou coordonateurs**	**Relecture assurance qualité faite par :**
CETE de l'Est : Centre d'études techniques de l'Équipement **CERTU :** Centre d'études sur les réseaux, les transports, l'urbanisme et les constructions publiques	Daniel CHOLLEY (CETE de l'Est) Patrick GENDRE (CERTU Département Systèmes)	NOUVIER Jacques CHARVIN Robert

Remarques préliminaires :

Résumé :

Ce rapport vient compléter la liste de documents constituant le corpus technique relatif à l'application du schéma directeur d'exploitation de la route pour les infrastructures classées en niveau 1A. Ce document est avant tout destiné aux DDE qui vont mettre en place un Système d'Aide à la Gestion du Trafic (SAGT) de niveau 1 (réseau de voies rapides urbaines), mais il devrait aussi intéresser les services qui ont déjà mis en place un SAGT, et plus largement d'autres exploitants qui ont mis ou mettent en place des SAGT pour exploiter le trafic sur d'autres types de réseaux, car une bonne partie des problèmes dépasse le seul besoin du SDER (Schéma Directeur de l'Exploitation Routière) de niveau 1, ainsi que les sociétés d'ingénierie et de service qui interviennent dans la définition, la conception et la fourniture de ces systèmes.

Ce document est une première version, encore incomplète, de ce qui pourrait constituer un guide technique pour l'informatique du SDER de niveau 1. Il a été constitué sur la base des comptes-rendus du groupe de travail « informatique SAGT » que le CERTU anime depuis juin 98, et de réflexions engagées par le CERTU parallèlement aux activités du groupe, en vue de capitaliser les connaissances accumulées et en améliorer la réutilisation pour les projets à venir. Pour améliorer la version actuelle et mieux répondre aux attentes du public auquel cette étude est destinée, nous remercions d'avance les lecteurs de nous faire part de toutes leurs remarques, critiques et suggestions.

Le document décrit d'abord les phases du cycle de vie d'un SAGT puis ses principales fonctions, et liste un certain nombre de recommandations ou remarques issues du groupe de travail « informatique SAGT » animé par le CERTU. Des documents complémentaires figurent en annexe.

Mots clés :	**Diffusion :**
Informatique, logiciel, guide technique, systèmes d'aide à la gestion du trafic, exploitation routière	Publique

Nombre de pages :	**Prix :**	**Confidentialité :**	**Bibliographie** :
62 pages	50 FF (7,62 Euros)	Non	Oui

Table des matières

1. INTRODUCTION 8

1.1. PRÉSENTATION DU DOCUMENT 8
1.1.1. OBJECTIFS 8
1.1.2. PUBLIC 8
1.2. CONTEXTE : CRÉATION D'UN GROUPE DE TRAVAIL « INFORMATIQUE POUR LES SAGT » (AVRIL 1998) **8**
1.2.1. MOTIVATIONS 8
1.2.2. OBJECTIFS DU GROUPE 9
1.2.3. OÙ EN EST-ON AUJOURD'HUI (ÉTÉ 99) ? 9

2. LES ÉTAPES D'UN PROJET INFORMATIQUE 9

2.1. CYCLE DE VIE 10
2.2. LIEN ENTRE PROJET INFORMATIQUE ET SDER 11
2.3. LES ÉTUDES PRÉALABLES 12
2.4. LE MONTAGE ET LE SUIVI DU PROJET 12
2.4.1. QUELQUES RECOMMANDATIONS CONCERNANT LA MAÎTRISE D'ŒUVRE 12
2.4.2. CONSULTATION - SIGNATURE DU MARCHÉ 14
2.5. LE DÉVELOPPEMENT 15
2.5.1. LA DÉFINITION DU SYSTÈME 15
2.5.2. LA CONCEPTION 23
2.5.3. LA RÉALISATION 23
2.5.4. LA MISE EN SERVICE 24
2.6. L'UTILISATION DU SYSTÈME 26
2.6.1. LE FONCTIONNEMENT 26
2.6.2. L'ÉVALUATION 27
2.6.3. LA MAINTENANCE 27
2.6.4. LE RETRAIT 30
2.7. SYNTHÈSE 31

3. FONCTIONS D'UN SYSTÈME INFORMATIQUE D'AIDE À LA GESTION DU TRAFIC 32

3.1. INTRODUCTION 33
3.2. BREF PANORAMA DES SAGT 33
3.2.1. AU SIER 33
3.2.2. À LA DDE DU RHÔNE 34

3.2.3. À LA DDE DES BOUCHES-DU-RHÔNE 34
3.2.4. À LA VILLE DE PARIS 35
3.2.5. « NOYAU GÉNÉRIQUE ». 35
3.3. EXEMPLES DE FONCTIONS 36
3.3.1. SUPERVISION TECHNIQUE DU SYSTÈME 36
3.3.2. RÉFÉRENTIEL DE CONFIGURATION DE L'APPLICATION 36
3.3.3. RÉFÉRENTIEL GÉOGRAPHIQUE DU RÉSEAU 36
3.3.4. ANALYSE DE DONNÉES DE TRAFIC (EN TEMPS DIFFÉRÉ) 37
3.3.5. MAIN-COURANTE 37
3.3.6. SYSTÈME DE BASE ET BASE DE DONNÉES 38
3.3.7. AIDE À LA DÉCISION 40

4. CAPITALISATION SUR L'INFORMATIQUE DES SAGT 45

4.1. UNE DÉMARCHE GLOBALE 45
4.2. RÉUTILISATION DE LOGICIELS 45
4.3. PROGICIELS 46
4.4. SERVEURS DE DOCUMENTS 46
4.5. SAGT GÉNÉRIQUE 47
4.5.1. CAHIER DES CHARGES " GÉNÉRIQUE " POUR UN FRONTAL, UNE MAO, UN RÉFÉRENTIEL TECHNIQUE DE SAGT/1 47
4.5.2. CAHIER DES CHARGES " IDÉAL " D'UN SAGT/1 SELON MARIUS 47
4.5.3. ÉTUDES À LANCER PROCHAINEMENT PAR LE CERTU 47
4.6. SITE INTERNET 48

5. CONCLUSIONS ET PERSPECTIVES 49

6. ANNEXE 1. LES ÉTAPES DANS CHEOPS. 50

7. ANNEXE 2. TRADUCTION D'EXTRAITS D'UN GUIDE AMÉRICAIN DE RECOMMANDATIONS CONCERNANT L'INFORMATIQUE POUR LES SYSTÈMES DE GESTION DE TRAFIC. [12] 51

7.1. LES BONNES PRATIQUES. 51
7.2. LES POINTS CLÉS (DU VOLUME 1). 51
7.2.1. DE LA NATURE DU LOGICIEL 51
7.2.2. CONTEXTE DE L'ACQUISITION DE LOGICIEL 52
7.3. LES POINTS-CLÉS (VOLUME 2) 52
7.3.1. SÉQUENCE D'ACTIVITÉS 52
7.3.2. CONSTRUIRE UNE ÉQUIPE 52

7.3.3. PLANIFIER LE PROJET 52
7.3.4. LES BESOINS 52
7.3.5. ACHETER OU FAIRE ? 53
7.3.6. LE CHOIX DU TYPE DE CONTRAT 53
7.3.7. L'ENVIRONNEMENT INFORMATIQUE 53
7.3.8. LA PROPRIÉTÉ INTELLECTUELLE 53
7.3.9. CALENDRIER DU PROJET 54
7.3.10. RECETTE 54
7.3.11. FORMATION, EXPLOITATION, MAINTENANCE 54
7.3.12. GESTION DE PROJET 54
7.3.13. GESTION DES RISQUES 55
7.4. QUE METTRE DANS LE PLAN DU PROJET 55
7.5. QUE METTRE DANS UN DOCUMENT DE DÉFINITION DES BESOINS 56
7.5.1. BESOINS FONCTIONNELS 56
7.5.2. PERFORMANCE 56
7.5.3. INTERFACES 57
7.5.4. ENTRÉES 57
7.5.5. SORTIES 57
7.6. SUGGESTIONS POUR UNE REVUE DES EXIGENCES 57
7.7. QUE PRENDRE EN COMPTE POUR L'ENVIRONNEMENT INFORMATIQUE. 57
7.8. DROITS DE PROPRIÉTÉ INTELLECTUELLE. 58
7.9. CHECKLIST : JALONS DU PROJET 59
7.9.1. NÉGOCIATION DU CONTRAT 59
7.9.2. BESOINS 59
7.9.3. DIMENSIONNEMENT 59
7.9.4. MANAGEMENT 59
7.9.5. RÉCEPTION 60
7.9.6. FORMATION 60
7.9.7. SUPPORT 60
7.10. RÉCEPTION 60
7.10.1. QUE METTRE DANS LE CAHIER DE RECETTE 60
7.10.2. QUE METTRE DANS LA PROCÉDURE DE RECETTE 61
7.10.3. QUE METTRE DANS LES CAS À TESTER 61
7.10.4. QUE METTRE DANS LE PROCÈS-VERBAL DU TEST DE RÉCEPTION 61
7.10.5. QUE METTRE DANS LE RAPPORT DE TEST 61
7.11. RÔLES DE SUPPORT SYSTÈME 62
7.12. GESTION DE CONFIGURATION 62

8. ANNEXE 3. PARTICIPANTS AU GROUPE DE TRAVAIL « INFORMATIQUE ET SAGT » (LISTE DE DIFFUSION AU 1/7/99) 63

9. GLOSSAIRE 64

10. RÉFÉRENCES 65

Le CERTU et les auteurs de ce document n'assument aucune responsabilité juridique ni ne s'engagent vis-à-vis de la complétude, de l'exactitude ou de l'utilité des informations présentées. La référence faite à un nom de marque, de produit, de procédé, de service, ou d'entreprise ne signifie pas qu'il soit soutenu ou recommandé par le CERTU ou les auteurs de ce document

1. Introduction

1.1. Présentation du document

1.1.1. Objectifs

Ce document est une première version, encore très incomplète, de ce qui pourrait constituer un guide technique pour l'informatique du Schéma Directeur de l'Exploitation Routière (SDER) de niveau 1. Il a été constitué sur la base des comptes-rendus du groupe de travail « informatique pour les Systèmes d'Aide à la Gestion du Trafic » (sur Voies Rapides Urbaines, ou SAGT/1) et des réflexions engagées par le CERTU depuis environ un an. Pour améliorer la version actuelle et mieux répondre aux attentes du public auquel cette étude est destinée, nous remercions d'avance les lecteurs de nous faire part de toutes leurs remarques, critiques et suggestions.

1.1.2. Public

Ce document est avant tout destiné aux DDE qui vont mettre en place un SAGT de niveau 1, mais il devrait aussi intéresser les services qui ont déjà mis en place un SAGT, et plus largement d'autres exploitants qui ont mis ou mettent en place des SAGT pour exploiter le trafic sur d'autres types de réseaux - car une bonne partie des problèmes dépasse le seul besoin du SDER de niveau 1 -, ainsi que les sociétés d'ingénierie et de service qui interviennent dans la définition, la conception et la fourniture de ces systèmes.

1.2. Contexte : création d'un groupe de travail « informatique pour les SAGT » (Avril 1998)

1.2.1. Motivations

Un Système d'Aide à la Gestion du Trafic comprend 4 volets techniques : exploitation, équipements, transmissions, informatique. Au ministère de l'Équipement, des groupes de travail nationaux fonctionnent sur les 3 premiers volets, l'informatique n'ayant pas été traitée (jusqu'en 97). L'informatique représente typiquement à peine 10% des crédits d'un projet du SDER de niveau 1, mais joue un rôle pivot entre les besoins fonctionnels de l'exploitation et le système technique.

En outre, il ressort par exemple de discussions avec des exploitants que les ressources à affecter à l'informatique sont très souvent sous-estimées (un applicatif " vit ", les besoins fonctionnels évoluent assez vite et le maintien en conditions opérationnelles ne se fait pas tout seul) ; plus largement, le problème de l'identification des tâches et de leur organisation, puis celui de la gestion et la valorisation des compétences sont encore mal appréhendés. Du point de vue technique, les choix de conception et d'organisation ne sont pas neutres vis-à-vis du bon fonctionnement du SAGT, et on n'a pas aujourd'hui de vision claire des solutions à partager ou des difficultés communes à résoudre.

La création d'un groupe de travail réunissant un petit nombre de personnes impliquées techniquement dans l'informatique des SAGT chez les exploitants routiers doit permettre d'échanger des informations entre techniciens et de faire remonter des recommandations aux exploitants, en abordant des problèmes techniques communs et en analysant les interactions entre le fonctionnel, les outils informatiques et l'organisation.

1.2.2. Objectifs du groupe

Comme défini lors de la création du groupe : « L'objectif est de constituer un petit groupe motivé se réunissant à échéances régulières (par exemple *tous les 2 mois*), fonctionnant de manière informelle, quitte à diffuser très largement les comptes-rendus et à tenir compte des retours. La production se limitera à la rédaction des comptes-rendus par le CERTU, plus si possible des études complémentaires commandées aux CETE, et l'organisation d'une journée technique.

Le thème est l'informatique (ni l'exploitation, ni les équipements et les communications avec le terrain) ; on s'intéressera en particulier à l'informatique en place dans les PC d'exploitation ; les projets informatiques (nouveaux développements) ne seront dans un premier temps traités que sous l'angle de leurs interactions avec les applications en place, de façon à ne pas trop se disperser.

Les sujets à explorer ne manquent pas, mais il vaut mieux être prudents sur les résultats que peut produire le groupe, et s'adapter en fonction de la qualité de la production après 6 mois.

Voici les thèmes que nous proposons d'aborder :

- description des métiers : compétences nécessaires pour l'informatique des SAGT, organisation des ressources.

- sous-traitance des tâches, relations avec les sociétés de service, interaction développement - opérationnel.

- prise en compte des exigences non-fonctionnelles (disponibilité, performances, testabilité, maintenabilité, etc.) : partir des solutions techniques et des problèmes rencontrés pour implémenter une fonction (par exemple, le recueil de données, ou la saisie du référentiel) pour mettre en évidence des leçons à tirer ; identifier les points-clés et comprendre les conséquences concrètes concernant les choix techniques des applications informatiques. »

1.2.3. Où en est-on aujourd'hui (été 99) ?

Le groupe de travail s'est réuni 6 fois depuis septembre 98 (la liste des participants est donnée en annexe). Un tour d'horizon du domaine a été fait, à partir duquel le plan de ce document a été établi.

La lecture des seuls comptes-rendus risquant d'être peu digeste, le groupe a décidé dès le départ de publier un document dont l'objectif serait de capitaliser toutes les remarques faites lors des réunions (sous forme de problématique des enjeux, recommandations, d'exemples « vécus », de pièges, de questions ouvertes...).

La structure générale proposée pour le document est la suivante :

1. une description du cycle de vie d'un SAGT avec quelques recommandations émanant du groupe de travail,
2. une description des principales fonctions, illustrée d'exemples,
3. une présentation des actions que le CERTU a engagées dans une démarche de capitalisation des connaissances relatives à l'informatique des SAGT/1,
4. en annexe, un glossaire, une bibliographie et quelques autres documents.

2. Les étapes d'un projet informatique

Cette partie présente les étapes typiques d'un projet et propose des recommandations issues du groupe de travail, qui n'ont pas l'ambition de faire le tour de la question, mais nous ont semblé pertinentes.

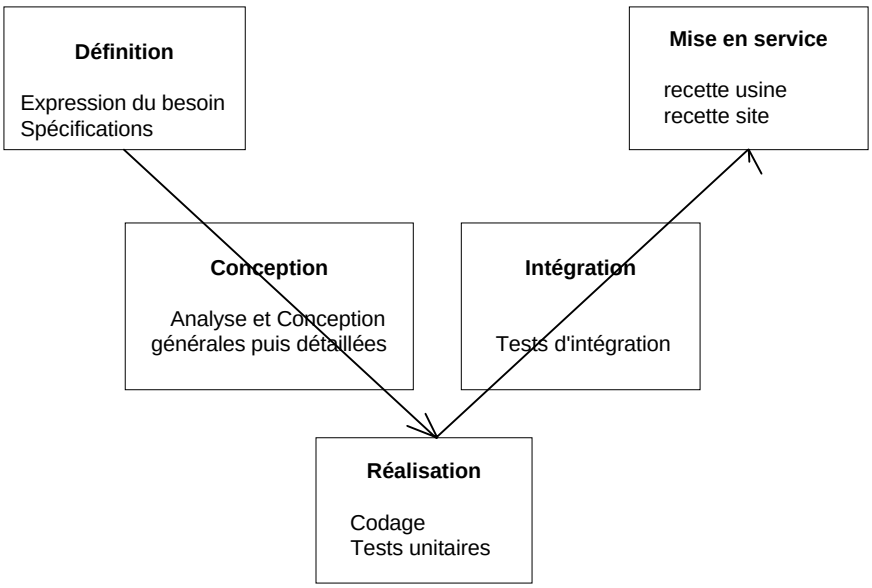
Pour d'autres recommandations tout aussi pertinentes, voir l'annexe (issues d'un document américain [12]).

2.1. Cycle de vie

La complexité fonctionnelle et technique du système et la diversité des personnes impliquées (informaticiens, utilisateurs) dans sa réalisation nécessitent l'emploi d'une démarche systématique et rigoureuse qui prenne en compte le cycle de vie. Selon les méthodes de conception et de développement employées (orienté objet, itératif) et le type de projet, les étapes du cycle de vie ont des appellations et des découpages différents, mais nous proposerons la liste d'étapes qui nous a semblé la plus proche des projets de SAGT, et qui est cohérente avec (bien que différente de) Cheops, la méthode de maîtrise d'ouvrage préconisée par le ministère de l'Équipement [1]. La description proposée s'appuie sur le document [8].

L'objectif de cette partie n'est pas de proposer une méthode de gestion de projets informatiques, il existe un grand nombre d'ouvrages sur le sujet[1] (cf. [9], [14], [16), mais d'utiliser le découpage en étapes pour structurer les contributions faites par notre groupe de travail.

Le cycle de vie d'un SAGT suit « classiquement » les phases d'étude, de réalisation, de validation et d'utilisation de tout logiciel. Ce découpage est également compatible avec le célèbre « cycle en V », qui met en évidence une phase descendante de définition qui comprend les spécifications et la conception, puis une phase ascendante de réalisation qui comprend l'intégration et la validation du système.

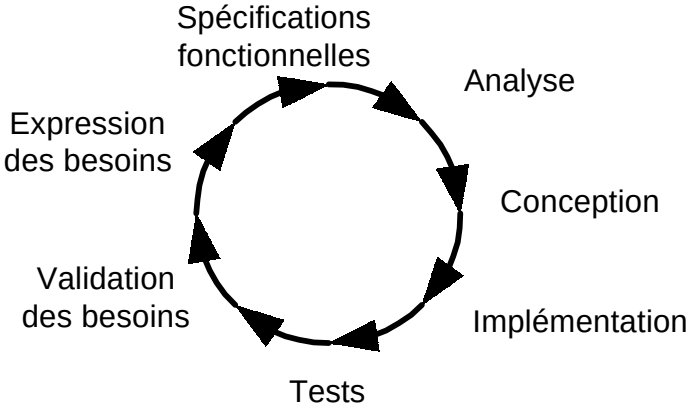


Dans ce cycle, chaque étape descendante est validée par l'étape ascendante se trouvant au même niveau.

Notez toutefois que ce modèle de développement crée un effet « tunnel » : le client du logiciel ne voit rien de son application jusqu'à la validation, ce qui pose problème notamment pour des systèmes dont les spécifications sont amenées à évoluer rapidement. On recommande donc souvent de découper le développement du système complet afin d'éviter un tunnel trop long.

[1] Par ailleurs, le gouvernement américain a financé et publié récemment une étude de recommandations pour les développements d'informatique pour les systèmes de transport intelligents et de gestion de trafic, dont un résumé en français figure en annexe.

Il est clair que d'une part les étapes ne s'enchaînent pas forcément linéairement (certains modes de développement favorisent même les allers-retours entre étapes successives), d'autre part que dans un service donné, plusieurs logiciels et applications peuvent coexister, à des étapes différentes du cycle de vie. La notion de cycle de vie est liée au caractère itératif et incrémental de tous les systèmes informatiques qui durent assez longtemps, comme le montre la figure ci-dessous. Enfin, le cycle de vie d'un projet de développement logiciel spécifique diffère d'un projet où le système final résulte de l'intégration de progiciels du marché (notamment au niveau de l'expression des besoins, où certains compromis devront être faits en fonction de l'offre de progiciels, voir [13]).



Des recommandations et remarques complémentaires peuvent être faites par le lecteur selon son expérience sur des projets.

2.2. Lien entre projet informatique et SDER

Pour ce qui concerne les SAGT au ministère de l'Équipement, ils sont développés dans le cadre du Schéma Directeur pour l'Exploitation Routière (SDER), de niveau 1 pour les réseaux de voies rapides urbaines. Les projets du SDER passent plusieurs d'étapes d'approbation formelle par la maîtrise d'ouvrage nationale DSCR : Dossier d'Études Préliminaires (DEP), Avant-Projet Sommaire (APS), Avant-Projet Détaillé (APD). Grosso modo, les études préalables correspondent au DEP (Dossier d'Études Préliminaires) et les études d'opportunité / faisabilité figurent typiquement au niveau de l'APS (Avant-Projet Sommaire), alors que les cahiers des charges pour la réalisation font partie de l'APD (Avant-Projet Détaillé) ; il est souvent fait appel à un consultant extérieur ou à un CETE pour aider à rédiger ces dossiers.

Le dossier d'APS comprend plusieurs sous-dossiers : contexte, stratégies d'exploitation, organisation, procédures, bâtiment CIGT, transmissions, équipements de terrain, programmation. Le volet informatique fait partie du sous-dossier bâtiment CIGT ; quoiqu'il en soit, il faut garder à l'esprit que l'informatique n'est pour le chef de projet qu'un des aspects à prendre en compte pour aboutir à un CIGT opérationnel. Au niveau de l'APS, le volet informatique doit partir de l'organisation et des stratégies d'exploitation, ainsi éventuellement que de l'existant (équipements, télécoms, applicatifs), pour identifier les acteurs, les fonctions et données susceptibles d'être informatisées, décrire les options possibles et justifier les choix d'informatisation envisagés, envisager les différentes possibilités en matière de montage du projet de réalisation, y compris le découpage en phases (marché clé en main, projet "classique" avec assistance à maîtrise d'œuvre pour rédaction du Dossier de Consultation des Entreprises (DCE) et suivi de la réalisation, rôle du réseau technique, logiciels existants, etc.).

L'APD informatique correspond ensuite à la rédaction du cahier des charges (CCTP), décrivant utilisateurs, fonctions et données, exigences et contraintes, et du Dossier de COnsultation des Entreprises (DCE).

2.3. Les études préalables

Elles ont pour but de valider la nécessité (opportunité) de développer un logiciel, puis d'estimer les ressources à mettre en œuvre (faisabilité), par exemple à 20% près. Les études préalables doivent prendre en compte un ensemble d'enjeux :

- L'enjeu économique : une étude financière doit déterminer la rentabilité économique de la mise en place d'un système face aux coûts d'étude et de développement.
- L'enjeu organisationnel : la mise en place d'un système bouleverse l'organisation des services. Une étude doit évaluer ces changements dans l'organisation des services et leur façon de procéder.
- L'enjeu technique : cet enjeu est déterminé d'une part par l'adaptabilité d'un système à une situation existante et d'autre part par le risque encouru vis-à-vis de la fiabilité et de la standardisation des matériels et progiciels utilisés.
  Par exemple, si le système doit utiliser les éléments d'un système existant, on devra évaluer les coûts d'intégration. Et si le système utilise un nouveau système d'exploitation, on devra quantifier le risque de ne pas le voir se pérenniser et d'être obligé d'évoluer vers un standard.
- L'enjeu humain : la mise en place d'un système engendre inéluctablement des modifications dans les façons de travailler. L'étude doit en particulier évaluer les besoins en qualification des personnels.

L'étude de faisabilité conclut en proposant une manière de monter le projet.

2.4. Le montage et le suivi du projet

Un projet informatique pour un SAGT (à créer ou à améliorer) oblige à définir un certain nombre de rôles, qui sont typiquement les suivants :

- Maître d'ouvrage (MOUV) : c'est le client qui paie le système (la maîtrise d'ouvrage peut être multiple et représentée dans un comité de pilotage dans le cas assez fréquent de projet en partenariat).
- Maître d'œuvre (fonctionnel) (MOEF) : c'est le chef de projet côté exploitant, typiquement un chargé de mission dans le cas d'un SAGT à créer dans une DDE.
- Maître d'œuvre (technique, ou assistant à maître d'œuvre) (MOET) : ce rôle est souvent (mais pas toujours) sous-traité à une SSII ou un Bureau d'Études qui va aider à formaliser le cahier des charges du système puis assurer la maîtrise d'œuvre des développements informatiques, typiquement dans le cadre d'un marché au forfait.
- Réalisateur (REA) : c'est typiquement une SSII qui intervient dans le cadre d'un marché informatique au forfait.
- Utilisateurs : ce sont les utilisateurs finaux de l'applications, qui ne se limitent pas aux seuls opérateurs.

Le montage de projet comprend :

- le choix du mode de consultation pour la maîtrise d'œuvre,
- le choix du mode de consultation pour la réalisation,
- la mise en place d'une structure de pilotage de projet,
- la formation des groupes d'utilisateurs (suivis ensuite jusqu'aux études de satisfaction),
- la mise en place d'un processus qualité.

2.4.1. Quelques recommandations concernant la maîtrise d'œuvre

2.4.1.1. Compétences de maîtrise d'œuvre

La maîtrise d'œuvre informatique est un métier et exige du professionnalisme ! Aujourd'hui il manque une " filière " de profils pour l'informatique d'exploitation du trafic, en tout cas au ministère de l'Équipement.

Traditionnellement, la culture des gestionnaires d'infrastructures est plus liée aux travaux de génie civil et de chaussée et ils ont du mal à appréhender le coût des développements et les relations avec les fournisseurs, à cause du caractère intangible du logiciel.

Comment d'autres gestionnaires de grands réseaux publics (EDF, SNCF, RATP ou autres sociétés de gestion des eaux) se sont-ils organisés ? C'est une question à creuser...

En règle générale, seules les grandes structures peuvent posséder en interne les compétences techniques nécessaires. Ainsi, SAPRR a choisi de (et a réussi à) assurer en interne la maîtrise technique de son informatique ; sa démarche qualité va désormais jusqu'à des revues de codes (commentaires des programmes sources). A contrario, ESCOTA a par exemple choisi de sous-traiter la maîtrise d'œuvre de son système Migrazur.

Le besoin prioritaire est de disposer de compétences internes en informatique " métier " (exploitation du trafic), car elles sont rares (et chères !) ; en revanche il est préférable de sous-traiter l'informatique " système ", compte tenu des évolutions rapides des produits.

Il est très utile de disposer en interne de capacités de développement afin de maîtriser les petites applications périphériques au système opérationnel et autres utilitaires. Il est indispensable d'avoir un administrateur informatique.

2.4.1.2. maîtrise d'œuvre technique

Le développement informatique est réalisé par des informaticiens dont on ne connaît pas en général la réelle capacité. Il est prudent que le MOET vérifie (ou fasse vérifier), dès les premières semaines de l'écriture du code, que celui-ci est conforme aux exigences qualimétriques. Il sera trop tard en fin de projet pour exiger une ré-écriture complète du code. Par ailleurs, il est important que le MOET dispose d'une compétence en ergonomie à tous les stades du projet. L'ergonome est le médiateur entre la vision du MOUV et les capacités des utilisateurs. Il faut avoir à l'esprit que l'informaticien a une structure de pensée fondamentalement différente de l'utilisateur d'un logiciel, et qu'une grand partie des logiciels actuellement développés est encore l'objet d'un rejet des utilisateurs.

2.4.1.3. Contrat d'assistance

Attention à la durée du contrat si vous prenez un contrat d'Assistance à MOUV (allant de la rédaction des spécifications à la recette) : si le projet prend du retard, le contrat d'assistance à MOUV risque de se terminer avant la livraison du système ! On commence à voir apparaître des clauses de marché qui pénalisent (fortement) le REA en cas de changement du chef de projet en cours de développement.

2.4.1.4. Relations avec les sous-traitants

Plus la MOE interne est forte, mieux elle définit ses spécifications, et donc plus le marché de ses fournisseurs potentiels peut être ouvert vers des généralistes, et ne plus se limiter à des constructeurs ou des SSII spécialistes ayant déjà livré des systèmes clés-en-main.

Il est sans doute plus difficile d'avoir une continuité dans les interlocuteurs responsables du projet si le contractant est une grande société (mais inversement celle-ci est plus pérenne). Il faut avoir à l'esprit la logique économique des SSII (notamment en matière de gestion des personnels).

En outre, le marché de la gestion de trafic n'est pas forcément le plus porteur pour un grand groupe industriel et peut faire les frais d'une réorientation stratégique, comme l'a constaté la Ville de Paris avec le retrait de ce marché de Dassault et Thomson dans les années 70.

On peut en tout cas prévoir dans le cahier des charges d'exiger des périodes de recouvrement en cas de changement de chef de projet.

2.4.2. Consultation - Signature du marché

2.4.2.1. Exemple de montage possible : appel d'offres sur performances

L'appel d'offres sur performances est une procédure possible de passation de marché, dont l'intérêt majeur est de permettre un dialogue direct avec les prestataires potentiels grâce à une ou plusieurs auditions, et d'offrir la possibilité aux entreprises de modifier leur offre initiale.

Cette procédure est guidée par la nécessité :

- de fixer des résultats (les performances) dans un programme fonctionnel détaillé,
- et surtout de prévoir leur mode d'évaluation.

Eu égard à sa longueur (6 mois au moins) et à sa lourdeur, l'appel d'offres sur performances est plus approprié pour des projets particulièrement complexes et coûteux. En fait il est souvent plus difficile de définir les bonnes métriques de performances que de spécifier le détail d'une application.

La DDE de la Drôme a employé la procédure d'appel d'offres sur performances pour développer le système de gestion de trafic de la LACRA de Valence, qui laisse une plus grande souplesse aux entreprises dans leurs propositions.

La procédure a également été utilisée pour la réalisation du PC SITER du Conseil Général des Hauts de Seine, et a permis d'aboutir à un système opérationnel avec un calendrier « tendu ».

Si parmi les lecteurs, certains connaissent d'autres exploitants qui ont aussi mis en œuvre cette procédure, nous les remercions de leur retour d'expérience. Selon vous, dans quelle conditions l'appel d'offres sur performances est-elle préférable à une consultation " classique " ?

Lorsqu'on lance un marché unique d'études et de réalisation, le cahier des charges reste très fonctionnel, or le coût de développement n'est estimé de façon réaliste qu'après les spécifications, ce qui pose problème dans le cas d'un marché au forfait !

Pourtant il existe bien d'autres solutions pour développer un SAGT, plus ou moins originales et adaptées au contexte. A la Réunion par exemple, le système MERLIN permettant l'exploitation en « 2+1 » voies de la route du littoral (RN1) a été confiée à un groupement (conduite par SPIE) qui a proposé une méthode de développement s'appuyant sur le maquettage / prototypage et la notion de contrôle intégré (ISIS), qui a amélioré la souplesse du projet et semble-t-il contribué à réaliser le système en un délai record en respectant les budgets [21].

Le débat porte en fait sur la frontière des responsabilités entre le MOUV et le REA. Dans le projet MARIUS, la doctrine est que « tout ce qui se voit est de la responsabilité du MOUV »: la conformité entre le terrain et sa traduction informatique, l'ergonomie générale et l'enchaînement des tâches opérateurs, la présentation des informations sur les écrans et sur les éditions. Le REA peut toujours proposer des amendements (améliorations possibles du fait des outils qu'il utilise, modifications nécessaires pour tenir les performances ou la cohérence des données si les spécifications ont des faiblesses, simplifications élégantes, etc.) mais ce n'est pas son rôle de définir ce que sera l'application. Inversement, c'est au REA de définir l'architecture logicielle, (pas au MOET, sinon le MOUV endossera la responsabilité des performances).

2.4.2.2. Documentation

On connaît l'importance d'une documentation à jour, en particulier d'avoir une bonne traçabilité entre les besoins, les spécifications et les évolutions successives du système réalisé. On connaît aussi le prix

de rédaction de la documentation... Il faut bien sûr trouver un compromis entre une bonne documentation et des coûts raisonnables.

2.4.2.3. Licences logicielles

Qu'ils aient ou non une signification juridique précise, des termes comme " accord de licence " ou " propriété " sont interprétés différemment d'une personne ou d'un service à l'autre.

Avant de signer le contrat :

- mettez-vous d'accord sur les droits de propriété intellectuelle liés au logiciel ;
- passez en revue la terminologie du contrat, assurez-vous que le contrat décrit clairement vos intentions ;
- explicitez la manière dont le code source, les exécutables, le support de formation et la documentation doivent être fournis et peuvent être utilisés, copiés, modifiés ;
- au besoin, ayez recours à un juriste spécialisé.

Par exemple, il peut être utile plus tard pour la maintenance de préciser dans le contrat que les utilitaires (simulation, test, debug) font partie de la fourniture. Dans le projet MARIUS par exemple, seuls les « run-times » sont propriété de la DDE. Les outils de développement sont à la charge du REA puis du mainteneur (qui assurent alors leurs mises à jour) : autant que faire se peut, il faut réfléchir à la cohérence entre le marché de développement et les marchés de maintenance ultérieurs...

Le CERTU recense actuellement les clauses de propriétés intellectuelles pratiquées dans les projets, afin d'essayer de proposer une clause-type.

2.5. Le développement

Le développement d'un système informatique s'appuie sur une suite d'étapes qui permettent à partir de l'expression des besoins d'arriver à la livraison d'un système opérationnel. L'utilisation d'un formalisme et de mécanismes fiables dans l'enchaînement de ces étapes permet d'obtenir un produit de qualité.

2.5.1. La définition du système

Nous avons volontairement inclus la définition du système dans la phase de développement, pour insister sur le fait que le développement ne se limite pas à la réalisation.

2.5.1.1. L'expression des besoins

Cette phase est primordiale. L'expression des besoins est le cahier des charges du système : elle doit traduire ce que la future application doit apporter à l'utilisateur. Elle est le fruit de discussions entre les utilisateurs, les concepteurs et les experts du domaine.

Durant cette étape, on analyse l'environnement et formalise les besoins des utilisateurs. La phase d'expression des besoins permet de s'accorder sur « ce que doit faire le système » avant de s'accorder sur la manière de le faire. L'objectif est de déterminer les éléments intervenant dans le système, leur structure et leur interaction. Il s'agit avant tout de comprendre le « métier » (l'exploitation routière) que l'on veut informatiser, et notamment :

- décrire les moyens et les méthodes de gestion existantes,
- décrire les différentes prérogatives des acteurs de la gestion du trafic,
- décrire le partenariat entre les acteurs,
- définir les méthodes de gestion coordonnées à mettre en place,
- définir un niveau de service du système.

La définition des besoins de l'exploitant va se traduire dans le cahier des charges du SAGT. Typiquement, le cahier des charges est utilisé pour rédiger le CCTP lors de la consultation pour la réalisation du système. En général, le document de définition des besoins n'est pas mis à jour au fur et à mesure de l'évolution du système (seules les spécifications sont mises à jour), de sorte qu'à un instant t, un travail de rétro-ingénierie est souvent nécessaire pour obtenir un cahier des charges représentatif a posteriori de l'état du système existant.

En règle générale, un cahier des charges comprend quatre grandes parties (sous diverses appellations) :

- description du contexte (essentiel car les candidats à une consultation ont besoin de se mettre « dans le bain » du projet ; de plus, une source importante de problèmes voir d'échecs dans les projets informatiques est qu'on a pas assez bien modélisé ou compris les processus avant de choisir lesquelles il fallait informatiser et comment) ;

- besoins fonctionnels (le service rendu aux utilisateurs : on en donnera des exemples dans la partie suivante) ;

- besoins non fonctionnels (Cf. ci-dessous : coût, disponibilité, maintenabilité, performance, facilité d'utilisation, sécurité, conformité à des standards, intégration avec l'existant, etc.) qui vont déterminer les choix de conception (et auront des impacts fonctionnels) ;

- exigences pour la réalisation (processus de développement, gestion de projet, assurance qualité, etc.).

2.5.1.1.1. Besoins non fonctionnels et contraintes techniques

2.5.1.1.1.1. Performances

Pour les temps de réponse, préciser par exemple les temps acceptables dans 95% des cas et les attentes maximales acceptables dans 5% des cas, car les temps de réponses ne sont pas uniformément distribués. Une fonction " chronomètre " dans l'application permettra de mesurer plus facilement les performances. Si le délai d'attente peut être long, prévoir un bouton « abandonner ».

Par ailleurs, des exigences de performances peuvent parfois coûter cher, surtout elles concernent des fonctions finalement assez peu utilisées. Dans le cas de grosses applications, des études spécifiques sur les performances s'avèrent parfois utiles. L'architecture (en particulier la gestion de la mémoire) a souvent un rôle important vis-à-vis des performances (et réciproquement, les performances dimensionnent l'architecture).

Notez aussi que les performances peuvent parfois s'effondrer lorsque la charge du système augmente, quelques mois après la réception, et qu'il faut alors faire évoluer le système (souvent le matériel et les configurations) pour maintenir les performances.

2.5.1.1.1.2. Facilité d'utilisation

L'importance de l'ergonomie du poste de travail des opérateurs n'est plus à prouver. Toutefois la manière de faire intervenir un ergonome n'est pas forcément évidente. Tout d'abord, l'ergonomie dépend beaucoup du " métier " ; un PC de gestion du trafic n'est pas un atelier industriel ou un standard de renseignements. D'autre part, l'ergonomie n'est pas synonyme de convivialité : dans certaines circonstances par exemple, le mode texte pourra être plus efficace que les fenêtres graphiques. Enfin, l'ergonome ne pourra guère améliorer un système dont les fonctions ont été mal conçues. Il est souvent difficile, en particulier si le système est déployé sur plusieurs sites, d'éviter les interfaces opérateur spécifiques... Attention toutefois à la multiplication des écrans dans les salles, même si il y a plusieurs applications : un poste de travail intégré est préférable. Faites intervenir un ergonome en amont de la réalisation de l'IHM !

La DDE13 et le CETE d'Aix ont été très satisfaits des conseils apportés par l'ergonome qu'ils avaient consulté pour les aider à rédiger le cahier des charges de l'informatique MARIUS. Ils recommandent notamment un poste de travail unique bi-écran, intégrant l'ensemble des fonctions informatisées.
A la Ville de Paris, on recommande l'utilisation d'un poste de travail banalisé et d'un profil utilisateur personnalisé, permettant à un opérateur de se " logger " n'importe où dans le PC en retrouvant son environnement de travail.

Dans tous les cas, la validation d'une maquette par les futurs utilisateurs semble pertinente lors de la phase de spécifications.

2.5.1.1.1.3. Disponibilité

Attention à ne pas formuler de trop grandes exigences en matière de disponibilité, sans avoir au préalable étudié la sûreté de fonctionnement et les modes dégradés de l'ensemble du SAGT. La disponibilité doit se concevoir au niveau du système complet, pas seulement de l'informatique [17].

La mise en place de solutions de redondance du système peut complexifier beaucoup le système et coûter cher alors que le maillon critique se trouve ailleurs…

2.5.1.1.1.4. Maintenabilité

La maintenabilité comprend plusieurs notions (portabilité, capacité d'extension, etc.) ; elle est très liée à la documentation ; un certain nombre de métriques permettre de l'évaluer [24].

Le plan de test doit faire partie des spécifications. La définition a priori des jeux et des outils de test (en-ligne et hors-ligne) afin de minimiser les risques de non-régression est un point essentiel (" built-in test ", module de supervision donnant une visibilité sur le fonctionnement interne du système et permettant d'automatiser des scripts de test de non-régression).

2.5.1.1.1.5. Évolutivité

Il faut éviter les vœux pieux du genre " le système doit pouvoir évoluer facilement " : tout besoin exprimé doit pouvoir être vérifié à la réception. Par exemple, on peut recenser un certain nombre d'évolutions probables ; demander au réalisateur d'indiquer comment le système pourra les absorber est un bon moyen d'évaluer a priori la robustesse d'une architecture. En outre, des spécifications d'interfaces publiques et bien documentées pour tous les modules permettront de faire évoluer plus facilement l'application.

2.5.1.1.1.6. Sécurité

Les exigences non-fonctionnelles jouent un rôle essentiel dans la conception technique des SAGT. Outre des critères comme la disponibilité, la modularité, l'intégrabilité, la réutilisabilité, la testabilité, la performance, la facilité d'utilisation, un autre critère important est la sécurité, d'autant plus que les réseaux informatiques sont désormais souvent ouverts vers d'autres partenaires voire vers l'Internet.

Pour l'instant, nous ne sommes pas en mesure de faire des recommandations sur ce thème, mais c'est certainement un point à creuser, en particulier pour les SAGT du ministère de l'Équipement, à l'heure où le réseau interne Icarré est déployé, et où de nombreux informaticiens du ministère y travaillent !

2.5.1.1.1.7. Architecture technique

Les seules exigences techniques qui devraient figurer dans un CCTP sont des contraintes de compatibilité avec l'existant (interfaçage, etc.) ou avec des directives issues de la direction informatique (par exemple, obligation que l'application fonctionne sous NT).

L'architecture technique doit répondre aux besoins : ce sont donc les besoins qu'il faut se donner les moyens de spécifier puis de valider. Les utilisateurs (les exploitants de CIGT en particulier) doivent

concentrer sur leurs efforts sur leur métier, et sur l'architecture fonctionnelle[2], et laisser les réalisateurs et les informaticiens prendre leurs responsabilités en matière de décisions techniques.

Le groupe de travail n'a donc pas de recommandations purement techniques à faire. En revanche, l'architecture technique doit être clairement documentée ; les choix techniques doivent être justifiés au " client ", en expliquant en quoi ils permettent de satisfaire ses exigences. D'où l'importance du document de conception générale.

2.5.1.1.1.8. Conformité à des normes et standards

Même si des standards sont loin d'être stabilisés (voire d'exister) dans tous domaines, il faut au moins connaître ceux qui existent, car on a souvent intérêt à les appliquer (si ce n'est pas une obligation !). Par exemple, pour faciliter les échanges de données entre exploitants, il faut intégrer autant que possible les mécanismes et dictionnaires d'échange standardisés (événements Datex, localisants Alert/C), tout en veillant à ce que les standards ne soient pas « en dur » dans l'application, tant ils sont sujets à évolutions. Par exemple, MARIUS a complètement externalisé le dictionnaire et les localisants pour Datex et AlertC, de sorte qu'uune modification dans le dictionnaire ou dans les points RDS ne nécessitent pas d'intervention dans MARIUS, mais uniquement dans la base de données de la DDE13 (GEOMARIUS).

2.5.1.1.1.9. Intégration à l'existant

C'est souvent un critère technique essentiel, qui est bien sûr à préciser au cas par cas.

2.5.1.1.1.10. Coûts

Au-delà des problèmes d'estimation financière (voir plus loin), il faut attirer l'attention du lecteur sur la diversité des solutions répondant à chaque fonction. Il faut savoir prendre du recul pour s'assurer du bien-fondé d'une fonction coûteuse : les techniques de l'analyse de la valeur sont intéressantes à cet égard. [27-29].

2.5.1.1.2. Exigences pour la réalisation (choix d'une méthode)

Nous préconisons de ne pas exiger l'emploi d'une méthode a priori : mieux vaut que l'entreprise travaille avec la méthode et les outils qu'elle maîtrise, plutôt qu'elle livre des documents purement de forme mais correspondant mal au travail effectif.

Il faut en revanche spécifier le plan-type et la taille de chaque document à produire, pour éviter des surprises et des « pavés » inexploitables !

Attention aussi à la complexité des méthodes Orientées Objets, surtout si elles sont mal maîtrisées par le réalisateur ou le MOE, et de plus en plus proposées par les entreprises (en partie parce qu'elles sont à la mode).

2.5.1.2. DCE

Après l'étape de définition des besoins, le cahier des charges va être complété (éventuellement reformulé avec l'aide d'un consultant) pour aboutir à un CCTP, lui-même intégré dans un dossier de consultation des entreprises (DCE) en vue d'une réalisation.

Le DCE apporte par rapport au cahier des charges une mise en forme des besoins et surtout des éléments concernant le volet contractuel et le suivi de projet (obligations du client et du fournisseur,

[2] il y a beaucoup de discussions sur les questions d'architecture, en particulier liées à une terminologie pas très stabilisée ; disons qu'il faut se limiter aux niveaux conceptuels et logiques, qui restent (autant que possible) indépendants des technologies.

phasage, procédures qualité suffisamment simples pour être respectées). Il serait intéressant de fournir des modèles de dossier « type », mais la construction de (parties de) CCTP-types ou de DCE-types ne peut se faire directement à partir d'une synthèse des CCTP existants, car les systèmes ont en général évolué et leurs CCTP ne correspondent plus à l'état actuel de l'application, et aussi car certaines erreurs ont été commises dans la rédaction initiale du CCTP, dont l'exploitant a su tirer la leçon. Pour obtenir ces informations, il faudrait un debriefing entre client et fournisseur à l'issue du projet. Dans le projet MARIUS, un tel debriefing était prévu initialement au marché, mais n'a jamais pu être conduit.

Il faudrait donc un travail de " reverse engineering " (ou rétro-ingénierie) pour aboutir à des CCTP utilisables par des tiers. Dans le cas particulier des SAGT exploités par les services de l'Équipement, ce travail peut être financé par le ministère ; le CETE d'Aix doit mener un tel travail pour l'informatique de MARIUS d'ici fin 99. Ce document sera mis à disposition des exploitants intéressés.

Pour éviter un travail de " rétro-ingénierie ", on pourrait a priori s'organiser pour que le projet soit conçu en vue de la réutilisation, en tout cas pour la rédaction des documents de définition des besoins, en séparant les besoins spécifiques et génériques, et en mettant à jour ces documents. Cette approche a été retenue à l'occasion de l'étude de définition d'une application comprenant les fonctions de frontal de communications, de maintenance du système, ainsi que le référentiel associé à la DDE du Rhône ; une version générique du cahier des charges sera disponible avant fin 99.

2.5.1.3. Estimation des coûts

En phase d'étude préalable, il faut avoir chiffré les enjeux (sur une durée de 10 ans) pour chaque fonction, chaque exigence non-fonctionnelle et avoir identifié les contraintes a priori dimensionnantes (la bien connue " analyse de la valeur ").

Pour pouvoir comparer clairement les prix de plusieurs réponses à un appel d'offre, il faut exiger une décomposition des prix qui identifie clairement chaque coût et les résultats attendus, pour la partie applicative comme pour le matériel et les logiciels de base. En particulier, il faut pouvoir connaître le coût prévu pour chaque fonction à implémenter (" principe de sécabilité ").

Un bordereau des prix obligeant les entreprises répondant à une consultation à chiffrer chaque poste selon une décomposition selon des fonctions bien identifiées dans le CdC facilite la comparaison des offres ?

Il semble quasi-impossible de comparer le coût des logiciels d'un exploitant à l'autre. En revanche, pour un projet donné, on devrait pouvoir proposer des ratios typiques pour un logiciel d'aide à la gestion du trafic, par exemple pour le prix et la durée relatifs par fonction, par phase. Mais il faudrait pour cela recenser et analyser les coûts constatés pour un grand nombre de projets...

De même, il n'existe pas à notre connaissance de ratios de coûts typiques (développement, conception, documentation, accompagnement, maintenance y compris non-régression). Il existe bien des méthodes d'estimation des coûts (basés sur les " points de fonctions "), mais elles sont assez lourdes et en pratique pas utilisées à notre connaissance…

Par ailleurs, les exploitants s'échangent sans doute assez peu d'information concernant les coûts des systèmes qu'ils ont fait développer ; une maîtrise d'œuvre expérimentée améliore beaucoup la connaissance des coûts " du marché ", a fortiori si elle est internalisée.

De même qu'il est leur difficile de " sentir " les prix du marché, les exploitants ont du mal à bien connaître l'offre des SSII. Toutefois les exploitants ont besoin de connaître l'offre technique, et cette connaissance de l'offre n'est pas (ou peu) capitalisée et échangée entre exploitants. On gagnerait également à inviter les consultants et SSII à contribuer à notre réflexion...

2.5.1.4. Les spécifications du système

La phase de spécifications est essentielle : elle doit permettre une reformulation complète des besoins exprimés dans le cahier des charges sous une forme utilisable pour la conception et le développement. Bien souvent cette phase est largement sous-estimée dans le planning.

Typiquement, cette phase se situe une fois choisie l'entreprise qui réalisera le système et sera la première tâche du marché de réalisation. Elle se traduira par la rédaction d'un document de spécifications, qui correspond à une définition du système commune aux utilisateurs et aux concepteurs. A partir de là, seules des modifications mineures peuvent être apportées dans l'expression des besoins (du moins en attendant un deuxième itération des développements). Le document de spécifications doit être compréhensible par l'exploitant, qui doit le valider avec le maître d'œuvre.

Attention, les dossiers de spécifications détaillées sont parfois illisibles ou dans un formalisme un peu barbare car générés plus ou moins automatiquement à partir d'outils informatiques. Il est recommandé d'avoir d'abord un document de spécifications générales (en français), puis un document de spécifications détaillées (qui décrive le modèle conceptuel, les algorithmes et comportements d'une manière plus formalisées). L'exploitant doit être capable de valider lui-même les spécifications générales, quitte à se faire assister dans la validation des spécifications détaillées.

Point important enfin, les spécifications doivent comprendre un volet test et validation, et un volet ergonomie (Cf. ci-dessous, maquettes et prototypes).

2.5.1.5. Modèles

Le terme 'modèle' est très général, il s'agit d'une description plus ou moins formalisée permettant de représenter le système selon différents points de vue. Sans rentrer dans les discussions méthodologiques, il faut en dire deux mots car des modèles sont mis au point et utilisés couramment dans les projets informatiques, d'une manière ou d'une autre.

C'est à partir de la phase de spécification que les modèles de l'application jouent un rôle-clé. Les modèles différent selon la méthode de développement utilisée ; certaines méthodes considèrent que chaque étape du cycle de vie consiste à compléter progressivement un modèle complet de l'application [19, 20]. En tout cas, la modélisation ne se limite pas forcément à la modélisation des données (dictionnaire, relations).

L'idée de proposer des modèles de référence pour la gestion du trafic n'est pas nouvelle, et a été en partie mise en œuvre sur certains domaines tels que :

- événements routiers (Datex en particulier),
- commandes des équipements de terrain (LCR, DIASER),
- géo-référencement (localisants Alert/C par exemple),
- description du réseau routier.

Toutefois dans chacun de ces domaines, il existe en fait plusieurs modèles, et/ou plusieurs variantes d'implémentation. Cela montre (s'il en était besoin) qu'il est (et sera) difficile de tout normaliser en matière de modèles, et aussi qu'il serait utile de pouvoir comparer les modèles entre eux.

Dans le domaine des transports collectifs, un modèle de référence a été développé au niveau européen (Transmodel). Il n'existe pas d'équivalent dans le domaine du trafic routier, mais toutes les applications existantes s'appuient sur des modèles plus ou moins bien documentés.

A quoi peut servir un modèle conceptuel de référence ? Son intérêt est de :

- gagner du temps en phase d'analyse pour les fournisseurs (SSII et industriels),
- fournir une base commune (termes, notions) de discussion,
- à terme, favoriser (en principe) une convergence des approches et l'ouverture du marché.

Les difficultés prévisibles sont :

- le passage du modèle au code (problèmes des outils de type Atelier de Génie Logiciel),
- l'effort nécessaire pour impliquer les exploitants dans la validation des modèles (animation),
- les évolutions du modèle (gestion des versions).

Du point de vue de l'exploitant, un modèle de référence pourrait être intégré en annexe à un CCTP, comme base de départ à utiliser dans l'analyse, et à adapter aux besoins spécifiques de l'application.

Comment aboutir à de tels modèles ?

- Tout d'abord, cela implique un travail préalable de terminologie : les notions et les termes présentent de nombreuses variantes d'un exploitant à l'autre (Cf. la norme pour la régulation de trafic urbain, de 91). Sans chercher à normaliser, on peut toutefois proposer une terminologie de base qui servirait dans le modèle de référence, et donner des exemples d'autres appellations possibles et de variantes. Le but d'un modèle de référence est bien justement que l'on s'y réfère et qu'on puisse expliciter les différences entre le cas d'un exploitant en particulier et cette référence.
- La définition de modèles de référence devrait s'appuyer sur le travail de capitalisation des cahiers des charges et des spécifications évoqué plus haut. Ce travail pourrait être mené par domaine fonctionnel. En effet, il semble que les applications d'aide à la gestion de trafic sont de plus en plus riches fonctionnellement, et la tendance est plutôt d'avoir un découpage en modules applicatifs communiquant entre eux qu'un seul logiciel monolithique ; on pourrait donc travailler à la

modélisation des domaines fonctionnels correspondant à ce découpage, si on spécifie des interfaces entre domaines.

- Dans un second temps, un ou plusieurs groupes de travail pourraient être lancés pour travailler aux divers domaines fonctionnels à modéliser (animation par des consultants, entretiens avec les exploitants, diffusion des résultats, mises à jour, etc.)

2.5.1.6. Évolution des spécifications[3]

Les spécifications sont une étape-clé, car elles traduisent un transfert des responsabilités entre client et fournisseur. Les spécifications sont signées par le client, et ont donc un caractère contractuel ; elles peuvent être opposables au CCTP initial en cas de contentieux. Une réutilisation de l'existant en matière de spécifications est donc potentiellement très utile...

De manière générale, on a vu que les documents de définition des besoins (cahiers des charges, CCTP) ne sont qu'exceptionnellement mis à jour. En revanche, les documents de spécifications et de conception, ainsi que les manuels utilisateurs doivent l'être si on veut pour voir maintenir le système de manière acceptable.

Toutefois, cette mise à jour n'est pas effectuée systématiquement, ou parfois, cette mise à jour est faite a posteriori, une fois que l'exploitant s'est rendu compte de son importance :

- à la ville de Paris, un effort assez récent a été fait pour mettre à jour l'ensemble des spécifications ;
- à la DDE69, la mise à jour doc-code a été intégrée au marché de maintenance ;
- à la DDE13, les dossiers de spécifications et de conception sont également mis à jour au fur et à mesure.

2.5.1.7. Maquettes et prototypes

Pour mettre au point les spécifications, il est souvent nécessaire de passer par une maquette, qui permet aux utilisateurs de préciser l'interface homme-machine (les écrans) qu'ils souhaitent.

Pour des projets particulièrement innovants, il est souvent nécessaire d'aller jusqu'à développer un prototype, qui implémente une partie des fonctions afin de valider une solution technique ; si bien que le prototypage peut aussi faire partie de l'étape de conception. L'innovation dans les logiciels des systèmes d'aide à la gestion du trafic passe nécessairement par du prototypage. Le processus d'innovation qui fait progresser les logiciels de SAGT doit faire intervenir ensemble chercheurs, exploitants, SSII et industriels.

Selon les méthodes de développement, le prototype peut être jetable ou réutilisé en partie pour le système opérationnel. Selon l'importance du prototype, son développement peut être un projet (de Recherche & Développement) à part entière ; réciproquement bien des projets passent par une première itération qui peut être considérée comme du prototypage.

Il faut aussi avoir à l'esprit un principe de base de gestion du risque technique : " en informatique, il est illusoire d'espérer faire bien du 1er coup ". D'où l'intérêt de prototyper avant de spécifier (Le prototypage favorise un découpage en modules, un développement itératif et la maîtrise de la complexité du système).

Cette approche a été utilisée à Marseille : tous les modules de commande d'équipements ont été prototypés avant d'écrire les spécifications du système " définitif ".

[3] Voir aussi plus loin le § sur la maintenance évolutive.

2.5.2. La conception

La phase de conception permet de définir la manière dont le système sera construit. Elle porte principalement sur les orientations techniques. Conformément au cahier des charges, un concepteur doit gérer les priorités en termes :

- de contraintes techniques,
- de besoins non fonctionnels (performances, disponibilité, sécurité, maintenabilité, coût, etc)
- de choix de méthodes d'outils de programmation.

Seuls des informaticiens sont consultés pour la définition de ces orientations, le MOET validant les dossiers détaillés, mais le client doit être capable de comprendre et donc valider les dossiers d'analyse et de conception générales. Le résultat de cette étape est constitué par des documents techniques justifiant les choix, mais peu accessibles à des utilisateurs non informaticiens. Cette étape ne sera réellement validée qu'a posteriori, lors des tests. La conception s'appuie en général largement sur des modèles[4].

La conception est classiquement séparée en 2 phases :

- la conception fonctionnelle (ou analyse), qui reste indépendante de l'implémentation,
- et la conception technique (ou conception tout court),

chacune de ces phases faisant l'objet d'une étude générale puis d'un dossier détaillé.

2.5.2.1. Conception fonctionnelle

La conception fonctionnelle générale donne les grandes lignes du modèle logique de données et des fonctions (ou des objets selon la méthode de développement) et précise les outils employés.

Un dossier détaillé est ensuite élaboré. Bien souvent, on y précise ou formalise certains points qui n'ont pas été approfondis lors des spécifications.

2.5.2.2. Conception technique

Lors de l'étude de conception technique générale, le concepteur définit les principales orientations techniques (outils, environnement informatique, choix des matériels, systèmes d'exploitation, réseaux, sécurité, middleware, etc.) et son architecture (découpage en modules et en processus), à partir des contraintes formulées dans les spécifications et d'éventuelles études techniques préalables. Il est vivement recommandé d'obtenir un dossier de conception générale qui présente de manière claire l'architecture et permette à l'exploitant de comprendre les justifications des choix techniques proposés dans la solution du fournisseur.

Les éléments d'architecture présents dans les dossiers de conception gagneraient à être mis en commun, mais il est clair qu'il sont difficilement disponibles, d'une part parce que parfois il n'en existe pas, d'autre part parce qu'ils sont souvent considérés comme confidentiels.

Le dossier de conception détaillée est ensuite essentiellement destiné aux développeurs ; il s'attache à définir la totalité des modules composant le système, et est difficilement compréhensible pour le client. Toutefois il peut être examiné (par échantillonnage) par le maître d'œuvre, lors d'un audit qualité ou d'une revue de projet.

2.5.3. La réalisation

[4] Cf. plus haut la discussion sur les modèles, dans le § sur les spécifications.

Elle est aujourd'hui encore trop souvent considérée comme l'étape principale dans la création d'un système informatique (« développer un logiciel, c'est écrire un programme »). Par manque de moyen et de temps, on privilégie cette étape par rapport aux précédentes.

En réalité la réalisation dépend très largement des étapes précédentes du cycle de vie. Une mauvaise étude préliminaire entraîne inéluctablement la réalisation d'un système ne répondant pas en totalité aux exigences. Il devient alors souvent coûteux d'effectuer des modifications.

La réalisation comporte plusieurs phases :

2.5.3.1. La programmation

Appelée aussi codage ou développement, la phase de programmation consiste à réaliser les différents modules du système dans un environnement de programmation.

2.5.3.2. Les tests unitaires et la mise au point

On teste les fonctionnalités de chaque module à partir de simulations et de jeux de tests, souvent de manière un peu « artisanale », c'est le développeur qui effectue lui-même ces tests.

2.5.3.3. L'intégration

Les différents modules sont assemblés pour former le système final.

Les tests d'intégration ont pour but de valider la complétude des différents modules, et l'adéquation entre le système et les exigences des utilisateurs définies dans les spécifications.

2.5.3.4. Réception « usine »

Le système est réceptionné sur le site du réalisateur, en présence du maître d'œuvre en général.

2.5.4. La mise en service

Cette étape comporte plusieurs phases :

2.5.4.1. Intégration sur site pilote

C'est l'installation du système. On intègre alors les éléments existants identifiés dans la phase d'analyse des besoins. Le système est raccordé aux systèmes de communication.

2.5.4.2. Vérification d'Aptitude au Bon Fonctionnement

La VABF a lieu sur le site de l'exploitation (ou sur un site pilote, si l'exploitant a plusieurs sites).

Des bancs de test ont été développés pour les équipements comprenant le Langage de Commande Routier, ainsi que pour les postes du Réseau d'Appel d'Urgence, à la DDE 13, à la DDE 69, et maintenant au SIER dans le cadre d'une collaboration avec le SETRA. Il est essentiel d'avoir des équipements de terrain dans le PC, dédiés au test.

2.5.4.3. Vérification de Service Régulier

La VSR dure plusieurs mois sur le site opérationnel et permet la validation définitive avant passage en garantie. La VSR est une phase essentielle, car vu la combinatoire, beaucoup de problèmes ne sont découverts qu'en conditions opérationnelles. Il faut passer rapidement de VABF en VSR, mais prévoir que la VSR ait une durée suffisante afin de constater un maximum de défauts. Autant que possible, les

conditions d'apparition d'un bug doivent être décrites en détail de façon à identifier plus facilement le problème à résoudre.

2.5.4.4. Déploiement

L'application est déployée :

- sur tous les sites prévus, sans oublier les mesures d'accompagnement (formation, etc.),
- et, en général, sur une plate-forme de tests chez l'exploitant permettant d'effectuer la maintenance sans perturber le fonctionnement opérationnel.

2.5.4.4.1. Migration

La migration vers une nouvelle version de l'applicatif est toujours difficile car il faut assurer la continuité de service, récupérer les données de référence et les historiques saisis sur l'ancien système (et toujours valides !).

Elle demande d'être validée avec soin. Il faut surtout bien planifier cette migration, afin qu'elle dure le moins possible. En effet, la phase de recouvrement (entre l'ancien et le nouveau système) oblige l'exploitant à utiliser et administrer deux systèmes. Dès que le nouveau système peut assurer le fonctionnement minimal, on a intérêt à basculer, quitte à enrichir progressivement le nouveau système.

La qualité des données alimentant le système est une responsabilité de la MOUV : l'informatique impose une rigueur, des procédures de qualité (notamment pour la définition du référentiel, la configuration du système, puis pour la qualité des mesures) que l'exploitant (pas le réalisateur ni le MOE) doit mettre en place. La mise en place de jalons de suivi du projet, et la rédaction des notices d'utilisation (consignes d'exploitation) sont à charge de la MOEF, sous le pilotage de la MOUV.

2.5.4.4.2. Outils de télédistribution

Les outils de télédistribution deviennent essentiels pour les mises à jour d'un outil déployé sur n sites. Par exemple, REGA est déployé sur 7 sites à la SAPRR (PC central + 6 régions), avec 100 postes clients.

2.6. L'utilisation du système

Elle comporte les phases de fonctionnement, de maintenance et d'évolution.

2.6.1. Le fonctionnement

La phase fonctionnement mérite d'être approfondie, car c'est bien sûr la raison d'être du système que de (bien) fonctionner. Le fonctionnement est rarement étudié dans le cycle de vie, car cette notion même de cycle de vie est une préoccupation des concepteurs et de la maîtrise d'œuvre, qui n'interviennent plus à cette étape.

Le fonctionnement démarre dès la VSR et continue jusqu'au retrait, les maintenances corrective et évolutive se déroulant en parallèle. Bien souvent, la conception et le développement sont dans une " logique projet ", et après la VSR, le système est supposer " tourner tout seul ". Pourtant, fonctionnement et maintenance sont liés...

2.6.1.1. Rôles des opérateurs

Les tâches d'exploitation et de maintenance sont en général séparées. Toutefois il est bon que les deux coopèrent.

- Par exemple, l'opérateur MARIUS a une vision fine du bon fonctionnement du système et alerte les mainteneurs ; c'est lui le " pilote " du système, un peu comme le conducteur d'un véhicule, qui fait appel au garagiste lorsque le tableau de bord ou le comportement du véhicule l'alerte.

- Au SIER, l'expérience a aussi appris que les tâches ne devaient pas être complètement séparées ; aujourd'hui ce sont les opérateurs trafic qui gèrent les priorités d'intervention des mainteneurs.
- A Marseille, l'opérateur a en outre un rôle d'information fort puisqu'il présente lui-même le bulletin trafic à Radio-France Provence, ce qui présente une charge supplémentaire difficile à assumer en cas de situation chargée, mais motivante et obligeant à valider la qualité des informations, à prendre du recul pour adopter le point de vue des usagers.
- A la Ville de Paris, l'opérateur a trois rôles : exploitation, information, et signalement des alarmes aux différents mainteneurs.

2.6.1.2. L'exploitation

Elle concerne l'utilisation du système. On s'intéresse alors à quantifier les améliorations apportées par la mise en place d'un nouveau système :

- Le système est-il en adéquation avec les attentes des utilisateurs ?
- Par exemple, la mise en place d'un SAGT améliore-t-elle les conditions de trafic ?

Dans cette phase on évalue également les nouvelles attentes des utilisateurs, les dysfonctionnements constatés. Ces problèmes seront ensuite traités dans la phase de maintenance.

2.6.1.3. L'administration

La « phase » d'administration concerne l'utilisation des fonctionnalités de maintien du système à un niveau de service correct. Elles sont effectuées par un technicien qui doit notamment configurer le système, gérer l'archivage des données, organiser la gestion des différentes versions, assurer l'administration des bases de données et du système d'exploitation.

Ces tâches, qui sont souvent conduites avec un outil d'administration et de configuration, sont à distinguer des tâches de « maintenance de l'administration » qui sont du ressort de la MOUV : contrôle du mainteneur informatique (vérification des mises à jour de la documentation, tests périodiques ciblés, suivi des tests de non-régression des nouvelles versions, planning des interventions...), gestion des archives de données, évolutions de configuration et de paramétrage.

2.6.2. L'évaluation

C'est un sujet important pour les SAGT qui doivent faire la preuve de leur rentabilité pour continuer à être financés. Il est important de mieux intégrer le travail de développement des outils SAGT avec les réflexions sur le « métier » d'exploitation routière (guide SDER / CIGT, etc.) : l'évaluation doit être pensée en amont (définition des besoins) pour être intégrée au SAGT et être ainsi rendue possible ou facilitée plus tard. En pratique, l'évaluation a lieu lors de l'exploitation et conditionne les actions de maintenance (Cf. ci-dessus).

Dans MARIUS, l'évaluation est intégrée au système à trois niveaux : fonctionnement du système, situation du trafic, algorithmes et actions.

2.6.3. La maintenance

Même si le retour d'expérience reste encore limité pour l'instant, la phase de maintenance devrait pouvoir faire l'objet de réutilisation d'un SAGT à l'autre.

2.6.3.1. La maintenance corrective

Elle consiste à réaliser des modifications mineures pour corriger des erreurs non détectées lors des tests.

2.6.3.1.1. An 2000

Ce paragraphe, bien qu'il sera rapidement périmé, est présenté à titre d'exemple de l'importance de l'organisation et de la maîtrise des projets informatiques.

Après l'Euro, le ministère de l'Équipement a bien sûr aussi défini une méthodologie pour le passage à l'an 2000 ; toutefois elle est conçue en priorité pour les applications de gestion. Les plans d'action sont en cours : recensement exhaustif des équipements et logiciels, analyse des risques, actions correctives, test (intégration de préférence été 99). Et il n'y a pas que l'informatique de gestion qui doive se préparer à l'an 2000 ! Une circulaire issue du Haut Fonctionnaire de Défense a été diffusée aux services déconcentrées fin 98.

- La DSCR a recensé les plans de passage à l'an 2000 pour les équipements et logiciels d'aide à l'exploitation.
- Le SETRA a été chargé de faire le point sur les PMV et le CERTU sur les équipements de régulation de trafic.
- Le CETE de Rouen est chargé de recenser le parc installé d'équipements dynamiques pour la DSCR.

Pour le niveau 1 (voies rapides urbaines), voici quelques informations :

- Au SIER, le responsable du passage à l'an 2000 est M. Ludovic GEORGES. Un plan d'action global a été défini, et lancé depuis mi-98. Au printemps 99, environ 50% des PMV étaient conformes ; un nouveau test des PMV aura lieu en février. Le test consistant à basculer tout le système au 1/1/2000 est prévu pour le mois d'août. Le sous-système le plus critique est le Réseau d'Appels d'Urgence, qui doit pouvoir fonctionner indépendamment du reste de SIRIUS : c'est vrai d'ailleurs pour tous les SAGT.
- A la DDE13, le mainteneur informatique a fait proposé une mise à niveau du système. Par exemple, les frontaux (sous Unix/SCO) doivent être changés. Par ailleurs, les équipements et logiciels ont été recensés, les principaux risques estimés, les courriers aux fournisseurs envoyés et la mise en conformité se fera progressivement.
- A la DDE69, un plan d'action a également été mis en place. Le système CORALY étant très récent une bonne part des équipements est déjà en conformité, mais la prudence reste de mise..., des équipements critiques pour le fonctionnement du système comme les routeurs par exemple, ne sont pas à l'abri. Les tests sont en cours...

Dans tous les cas, il est évident qu'il est préférable d'avoir visé de « boucler » le plan pour le 1er semestre, compte tenu du risque d'indisponibilité des fournisseurs en fin d'année et des délais administratifs pour les commandes.

La ville de Paris a demandé un budget de 4 MF pour faire migrer ses systèmes, notamment car le système d'exploitation actuel (HP-UX 9.0) ne " passe " pas l'an 2000, ce qui implique de nombreuses modifications (avec risque de régression)…

Pour information, l'administration américaine (FHWA : équivalent de la DSCR, DoT : ministère des transports) présente la façon dont le problème est abordé sur son site Web (www.fhwa.dot.gov/y2k, www.year2000.com, www.tasc.dot.gov/y2k).

2.6.3.2. La maintenance évolutive

Elle vise à des modifications plus importantes afin de mettre en place de nouvelles fonctionnalités ou s'adapter à un nouvel environnement.

2.6.3.2.1. Évolution mineure et évolution majeure

À partir de quel coût de maintenance évolutive faut-il lancer une évolution majeure (nouvelle version) ? Pour décider comment faire évoluer un système, il est souvent fait appel à un consultant pour réaliser un audit de l'existant : cela a été par exemple le cas en 99 pour la DDE69 (système CORALY et frontal de communication). On s'aperçoit qu'un simple 'audit' d'une application ne suffit pas. De plus en plus, une application fait partie d'un système et est interconnectée : il faut donc orienter les audits de manière à ce qu'ils puissent chiffrer et comparer divers scénarios envisagés par le maître d'ouvrage, en termes d'évolution fonctionnelle et d'intégration à l'existant ; en outre, un audit sur une application existante doit se traduire en termes d'exigences (méthodes, qualité) pour sa maintenance ou le développement de nouvelles fonctionnalités.

2.6.3.2.2. Durée de garantie d'un logiciel

Compte tenu de la rapidité d'évolution des environnements informatiques, les entreprises s'engagent rarement sur une garantie de plus d'un an (moins que pour un produit grand public !).

2.6.3.3. La maintenance préventive

La maintenance préventive est particulièrement efficace pour certains types d'équipements (sujets à usure), pour la gestion des stocks (approvisionnement en consommables), ou pour surveiller la mémoire disponible (disque notamment).

2.6.3.4. Prestataire de maintenance

2.6.3.4.1. Coût de la maintenance

La maintenance annuelle coûte au moins 5 à 10% du prix du logiciel. L'entreprise responsable de la maintenance corrective a typiquement un contrat de 3 ou 4 ans. Il est ensuite difficile d'en changer car elle peut " verrouiller " techniquement les évolutions et dicter ses prix. Il faut donc pour les évolutions majeures mettre le mainteneur en concurrence, le mainteneur jouant alors le rôle de maître d'œuvre et assurant la bonne intégration avec l'existant… Bien sûr, le mainteneur reste bien placé pour proposer des prix inférieurs.

Les budgets de maintenance sont difficiles à estimer a priori et souvent sous-évalués. Une difficulté supplémentaire vient du fait que la maintenance évolutive est souvent considérée comme faisant partie des crédits de fonctionnement (titre 35.43 en comptabilité publique). Mais en principe, la maintenance corrective doit relever du fonctionnement et la maintenance évolutive des crédits d'investissements ; pour un marché de maintenance globale, il faut donc un marché à double imputation (Ex. de CORALY - DDE69).

2.6.3.4.2. Tierce maintenance applicative

Peut-on sous-traiter la maintenance à un autre prestataire que le réalisateur ?

C'est ce qu'on appelle la tierce maintenance applicative (TMA) ; c'est souhaitable dans la mesure où cela oblige à avoir un logiciel bien documenté et rend l'exploitant moins dépendant de son prestataire de services informatiques. L'analyse des offres pour une TMA est grandement facilité si on a défini a priori les critères de qualité attendus du logiciel (qualimétrie du code et de la documentation).

L'intérêt pour l'utilisateur est de mieux maîtriser la prestation et notamment ses coûts.

La Commission Centrale des Marchés et la Direction de la Concurrence et des Prix sont attentives à ce que les principes de libre concurrence soient respectés, et à tout ce qui peut ressembler à du prêt illégal

de main-d'œuvre : on ne peut passer un marché avec un fournisseur de prestations intellectuelles sans préciser l'objet de ces prestations (en gros, le travail en régie est interdit). Les évolutions logicielles ne peuvent donc pas se faire au fur et à mesure de l'expression des besoins. Malheureusement, aucune solution miracle n'est proposée par les censeurs, à charge pour les MOUV de toujours savoir ce qu'ils veulent développer, avec parfois 3 ans d'avance...

Les marchés de réalisation devraient comprendre une phase de formation du prestataire TMA, et la fourniture d'une plate-forme de développement (c'est dorénavant le cas à la Ville de Paris).

Pour l'instant, le retour d'expérience reste limité en la matière, mais la DDE13, comme le SIER, ainsi que la DDE69 et la Ville de Paris, sont en train de travailler sur leurs contrats de maintenance informatique pour leur SAGT. Il serait intéressant d'interroger d'autres grands exploitants, comme le CIGREF[5], la RATP, la SNCF ou des concessionnaires d'autoroutes pour progresser dans ce domaine.

2.6.3.4.3. Exemples

- A la Ville de Paris, la maintenance du système (sauf des PMV et caméras) est prise en charge par le groupement GECIR comprenant EDF pour les équipements et la Générale des Eaux (Mercur) pour le logiciel.
- A Marseille, le système MARIUS a été audité en 96 ; le réalisateur du logiciel (Sagem) en est aussi le mainteneur
- A Lyon, le système CORALY n'est pas encore entré en phase de maintenance (99).

2.6.4. Le retrait

C'est l'ultime étape du cycle de vie. Elle consiste à mettre en place des procédures de changement de système tout en pérennisant les données et éventuellement les fonctionnalités du système. Cette étape est rarement étudiée en tant que telle et est plus souvent analysée dans la phase d'étude préalable d'un nouveau système devant assurer la succession de l'existant.

[5] Club Informatique des GRandes Entreprises Françaises.

2.7. Synthèse

Nous reprenons dans le tableau qui suit les phases et sous-phases du cycle de vie, en indiquant comment se partagent les responsabilités dans un projet « typique ». Ce découpage en phases va nous être utile pour le classement des contributions du groupe de travail, dans la partie suivante.

phase		**responsabilités**
Études préalables	Opportunité, faisabilité	validation MOUV
Montage du projet	Montage et suivi du projet	MOEF+T
Développement - Définition du système	Expression des besoins, cahier des charges et DCE	rédigée par MOUV, choix des MOET et MOEF
	Spécifications	REA, approuvée par MOEF et MOET
	Maquettage et prototypage	REA, approuvée par MOEF et MOET
Développement - Conception	Conception (générale puis détaillée)	REA validée par MOET[6]
Développement - Réalisation	Programmation	REA
	Tests unitaires, mise au point	REA, validés par MOET[7]
	Tests d'intégration	REA, validés par MOET[4]
	Réception[8] usine	REA, validés par MOEF et MOET[4]
Développement - Mise en service	Intégration sur site pilote, Vérification d'Aptitude au Bon Fonctionnement (sur site pilote)	REA, validés par MOEF et utilisateurs
	Vérification de Service Régulier (opérationnel) et réception finale	utilisateurs en exploitation, validés par MOEF
	Déploiement	en exploitation, (y compris accompagnement)
Utilisation du système - Fonctionnement	Exploitation et administration	utilisateurs
Utilisation du système - Maintenance	Maintenance corrective	REA (éventuellement une autre entreprise) + MOET
	Maintenance évolutive	REA (éventuellement une autre entreprise) + MOET et MOEF

[6] La MOET n'est pas engagée par une approbation – tout du moins en conception détaillée – mais les documents doivent lui être remis.

[7] MOET et MOEF peuvent demander à participer à ces phases, mais là non plus, ne sont pas engagées (seules la VABF et la VSR sont contractuelles).

[8] On distingue parfois le terme de 'recette' (où un test est « déroulé »), comme lors de la sortie d'usine, et le terme de 'réception', où le client accepte le produit, après une éventuelle période d'essai (cas de la VSR).

Utilisation du système - Retrait	Retrait (migration)	MOUV

REA : réalisateur ; MOE : maître d'œuvre (Fonctionnel ou Technique) ; MOUV : maître d'ouvrage.

3. Fonctions d'un système informatique d'aide à la gestion du trafic

Cette partie présente rapidement les fonctions présentes dans un SAGT (Cf. aussi [13]), puis propose quelques recommandations issues du groupe de travail concernant un éventail de fonctions fournies à titre d'exemples.

fonction	description
plateforme de développement	sous-système (comprenant son propre système de base et des applicatifs) permettant de développer, tester, intégrer des modules logiciels sans compromettre le fonctionnement opérationnel du PC, mais en utilisant éventuellement certains de ses composants en temps réel
système de base	matériel, système d'exploitation, réseau, et " middleware " : par opposition aux applications, sous-système qui permet au logiciel de s'exécuter
communications avec le terrain	dialogue avec les équipements et systèmes installés le long des infrastructures gérées par l'exploitant
coordination avec les autres exploitants	dialogue avec les systèmes qui gèrent des réseaux interagissant avec celui de l'exploitant
diffusion d'information	élaboration et transfert d'information vers un serveur destiné à informer les usagers
commande des équipements	actionne les équipements de terrain, indépendamment des couches de communication sous-jacentes
référentiel et base(s) de données	ensemble des données de référence, de la configuration du réseau, des équipements et des actions d'exploitation
main-courante	saisie et consultation des événements utiles pour les exploitants
interface opérateur	présente les informations utiles à l'opérateur et lui permet de piloter l'exploitation
interface configurateur	saisie, consultation et mise à jour du référentiel décrivant le réseau et des paramètres du système
interface administrateur	présente les informations utiles à l'administrateur du système et lui permet de piloter
surveillance du réseau (synoptique)	affiche en grand format des représentations graphiques de l'état du réseau ou toute information utile pour la salle d'exploitation
export et éditions	formate et transfère des données du système en vue d'une impression ou d'une utilisation par une autre application
analyses en temps différé	archivages, consultation, " rejeu " et analyse des données historiques en vue de planification, prévisions, et études statistiques
aide à la décision en temps réel	permet de décrire les stratégies d'exploitation, de proposer des actions en fonction de la situation de trafic en utilisant des algorithmes de régulation

traitement des données de trafic	pré-traitements des données et élaboration des mesures et d'informations agrégées
supervision et exploitation du système	donne une visibilité sur le fonctionnement des composants qui constituent le système

3.1. Introduction

Le Système informatique d'aide à l'exploitation du trafic routier comprend ici toute l'informatique (matériel, logiciels de base, logiciels spécifiques ou progiciels) utilisée par un gestionnaire de trafic ; par exemple, les logiciels des équipements de terrain en font en principe partie : on peut développer du logiciel spécifique pour une station de mesure météo, ou toute autre fonction pas proposée au catalogue du constructeur, et donc les équipements ne sont pas de purs " composants matériels " et font aussi partie de l'informatique... cela dit en pratique les logiciels et le sous-système associé aux équipements sont traités par ailleurs dans d'autres groupes de travail nationaux (et nous nous concentrerons sur le " cœur " des PC d'exploitation). La téléphonie et la vidéo ne font partie de l'informatique que dans la mesure où elle sont intégrées au logiciel d'aide à l'exploitation, ce qui est encore rarement le cas.

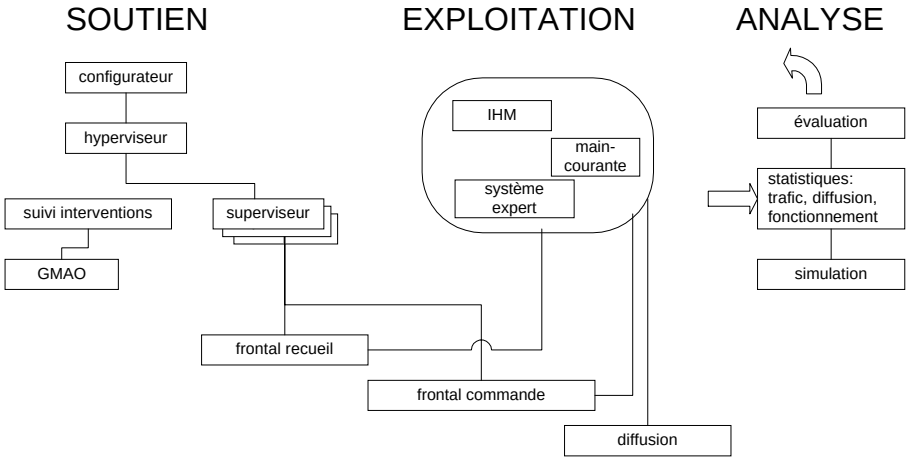
Le tableau précédent présente les principales fonctions informatisées dans un système d'aide à la gestion du trafic ; le découpage proposé correspond à celui que l'on rencontre souvent en pratique dans les systèmes, à de multiples variantes près, et avec des dénominations elles aussi variables.

3.2. Bref panorama des SAGT

A titre « pédagogique » (mais aussi parce que nous ne disposions pas des mêmes schémas pour chacun des systèmes !), nous présentons des vues de 4 SAGT/1 existants, qui permettent de constater qu'il y beaucoup de façons de représenter un système informatique : modules (applications), matériel (physique), processus (temps réel et communications), fonctions (logique).

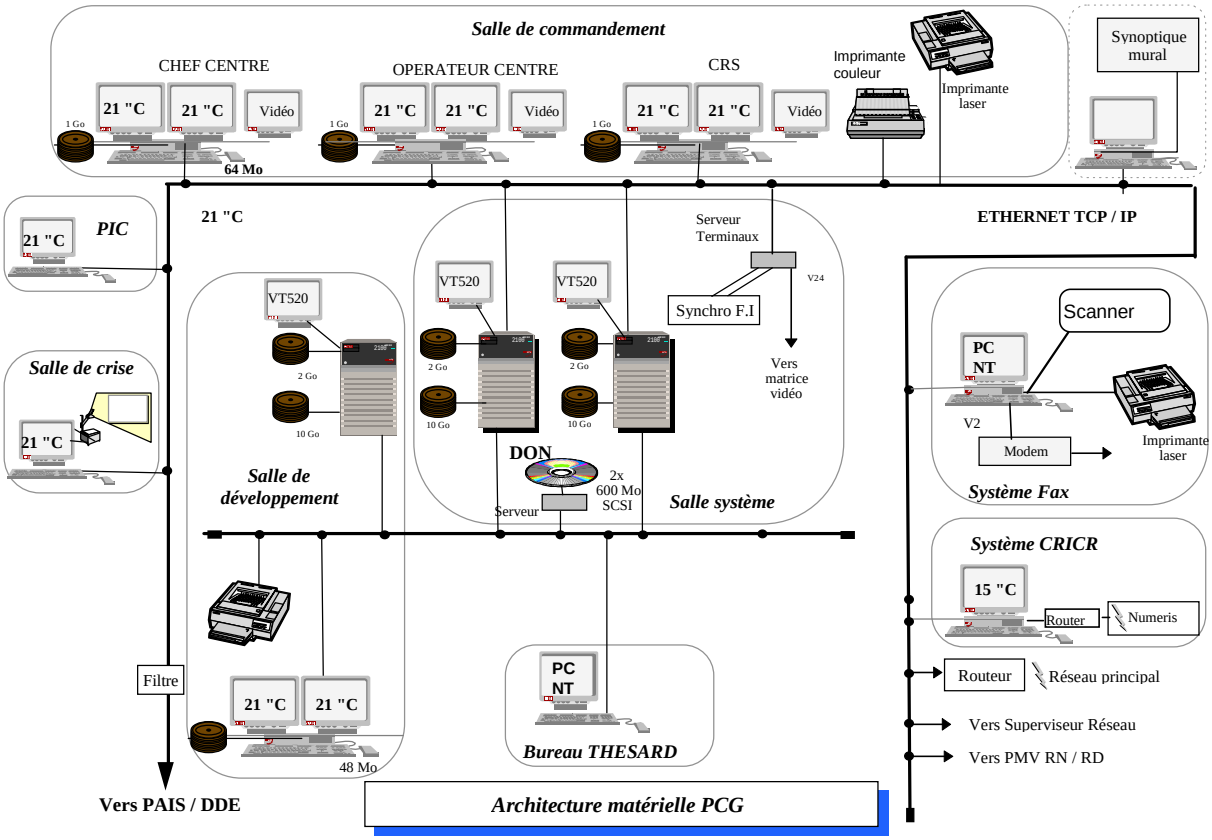
3.2.1. au SIER

Voici une " cartographie " simplifiée des applications au SIER dans la perspective de son schéma directeur informatique ; certaines fonctions ne sont pas encore informatisées ou doivent encore faire l'objet de développements majeurs. Dans une certaine mesure, ces fonctions sont aussi celles de tout SAGT de niveau 1.



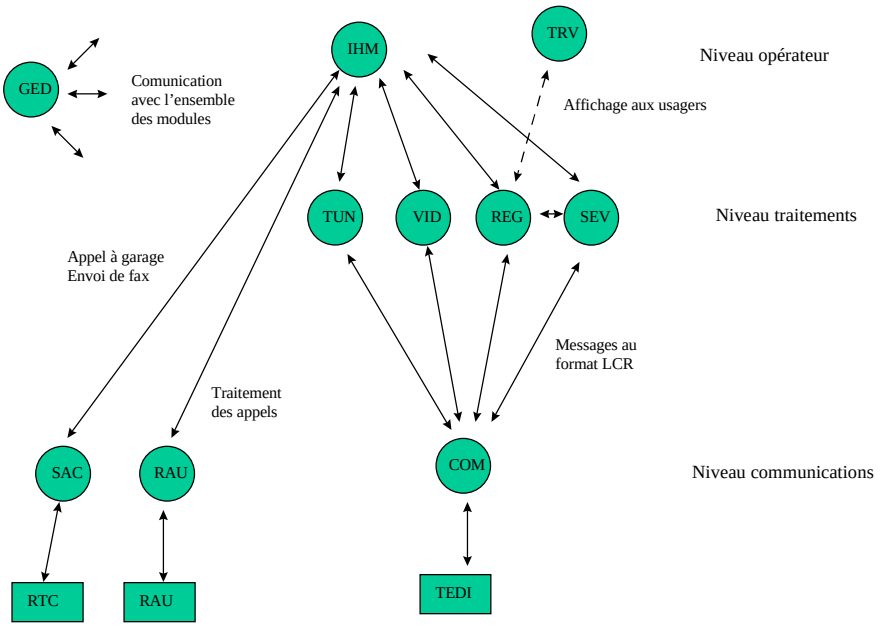
Modules applicatifs au SIER.

3.2.2. à la DDE du Rhône



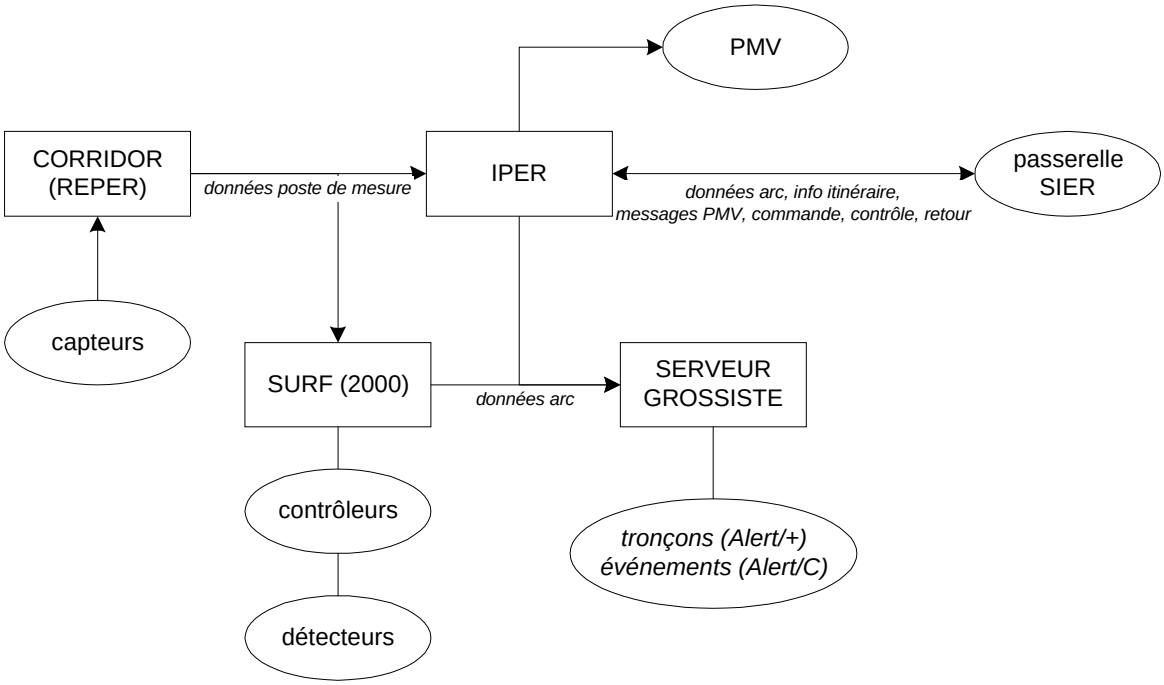
Vue schématique de l'architecture matérielle du PCG CORALY.

3.2.3. à la DDE des Bouches-du-Rhône



Vue schématique des processus du SAGT/1 MARIUS.

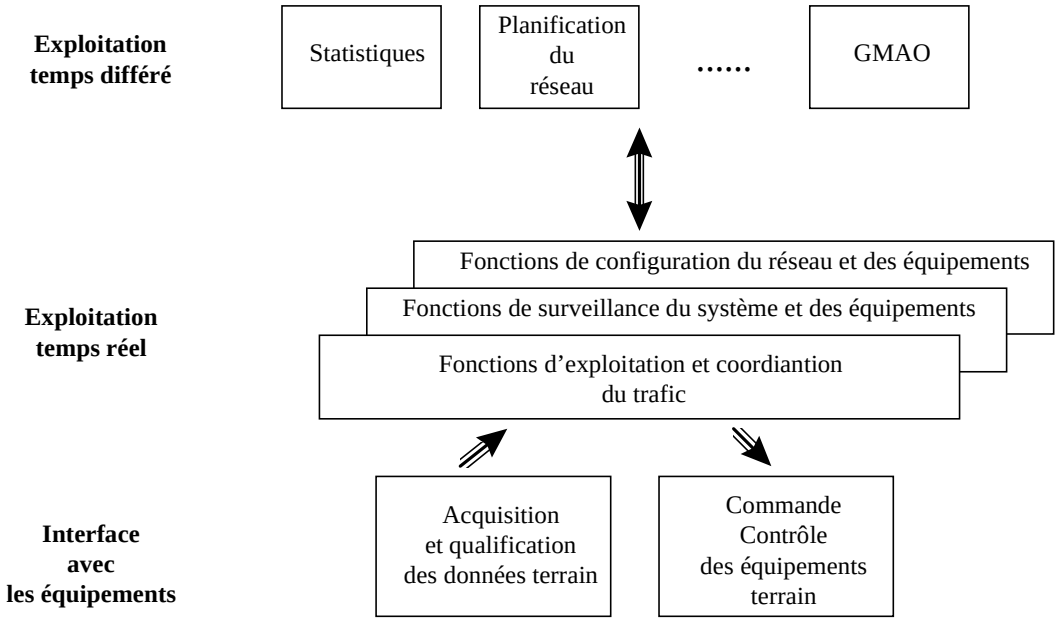
3.2.4. à la Ville de Paris



Vue schématique du contexte matériel des systèmes IPER et CORRIDOR (Berlier) et SURF (Lutèce).

3.2.5. « noyau générique ».

Les fonctions « de base » présentes dans tout SAGT identifiées dans [15] sont les suivantes :



Vue schématique des fonctions de base d'un SAGT/1.

3.3. Exemples de fonctions

3.3.1. Supervision technique du système

De l'avis général, les opérateurs doivent pouvoir superviser le bon fonctionnement du système, faire remonter les pannes aux mainteneurs, et aider à déterminer les priorités dans l'ordre des interventions.

- Le SIER, dont la taille justifie des moyens de maintenance lourds, introduit la notion d'hyperviseur, permettant les informations de supervision issues des différents sous-systèmes (équipements de terrain, courants forts, tunnels, réseau, etc.). La fonction d'hypervision sera d'abord implémentée pour SIRIUS Ouest. Sont distinguées les applications de supervision, de suivi d'interventions, et de GMAO (gestion de la maintenance assistée par ordinateur).
- Pour la DDE13, dont la taille du service d'exploitation routière est plus réduite, il n'a pas paru opportun d'informatiser la fonction de GMAO. De son côté, la DDE69 a lancé une étude de définition pour les fonctions de supervision et de maintenance, en même temps que pour le frontal de communication. Un document « générique » sera produit dans le cadre de cette étude et disponible avant fin 99.

3.3.2. Référentiel de Configuration de l'application

Les échanges entre systèmes de plusieurs exploitants se développant, la qualité des données devient un point de plus en plus critique. En particulier, il doit y avoir cohérence entre les référentiels de chacun.

De manière générale, la configuration du système est une tâche importante et lourde, et dans la mesure du possible, on évitera que la configuration des applications soit " en dur " (ce qui nécessite une intervention en cas de modification ultérieure). On cherchera aussi à ce que l'administrateur puisse définir une bonne partie des tâches de configuration dans des fichiers de " scripts " éditables.

Un problème " classique " en modélisation : si on veut pouvoir faciliter les évolutions de configuration, il faut penser à séparer les identifiants (physiques) et les adresses (logiques) des équipements. La phase de modélisation est aussi le moment de bien réfléchir au contrôle de cohérence des données.

Il est essentiel de " blinder " les spécifications en matière de contrôle de cohérence des saisies (paramétrage, configuration), et d'avoir des outils pour assurer cette cohérence. Les applications fonctionnent souvent en multipostes : attention, cela a des conséquences, notamment sur la configuration, le test, la maintenance et même sur le fonctionnel. Ainsi, la Ville de Paris a " blindé " les spécifications du système IPER afin d'éviter des modifications simultanées et intempestives de la configuration [31].

3.3.3. Référentiel Géographique du réseau

Il semble important pour les systèmes d'aide à l'exploitation du trafic de séparer la fonction " référentiel du réseau " de la fonction " visualisation de l'état du trafic " en temps réel, la première pouvant s'appuyer sur un SIG, la seconde gagnant à s'appuyer sur des vues pré-définies. L'analyse en temps différé peut à nouveau bénéficier de la richesses fonctionnelle d'un SIG. En revanche, on préférera en utilisation temps réel pour l'exploitation plutôt utiliser des vues synoptiques pré-définies qu'un SIG.

Cette fonction peut s'appuyer sur un SIG en amont du système opérationnel. La DDE13 a fait le choix de séparer son référentiel GéoMARIUS de l'application d'exploitation, de manière à réduire la complexité et d'améliorer les performances de MARIUS, et de mieux gérer les données de référence. Le coût est d'environ 50 kF pour la modélisation des données, 15 kF pour l'acquisition de

l'environnement de développement SIG GéoConcept, plus 4 mois de saisie en interne à la DDE13 ; il faut bien sûr ajouter à cela le développement d'une interface avec l'application MARIUS.

En plus du référentiel proprement dit, GéoMARIUS contient la définition des vues qui seront présentées en temps réel à l'opérateur sur le système MARIUS, ce qui simplifie la mise à jour de la configuration du système. GéoMARIUS permet à l'exploitant de s'approprier le système, en l'obligeant à tenir à jour toutes les informations de référence sur le réseau qu'il gère, et en dessinant lui-même ses vues synoptiques.

Contrairement à une cartographie classique, ce SIG contient finalement peu d'objets. GéoMARIUS est interfacé avec l'application d'aide à l'exploitation MARIUS, ce qui rend cette dernière en théorie réutilisable sur d'autres sites, puisqu'aucune vue du réseau marseillais n'y est définie " en dur ". Par ailleurs les vues synoptiques peuvent être animées, ou afficher des informations supplémentaires, par exemple des données de trafic en cas de clic souris sur une station de mesure (" menu contextuel ").

Une approche analogue a été entreprise par la DDE31 pour le projet ERATO à Toulouse ; le SIG s'appuie d'ailleurs sur le même produit GéoConcept de la société Alsoft, qui gère notamment les localisations en PR.

3.3.4. Analyse de données de trafic (en temps différé)

C'est un point important, très lié à l'évaluation des effets du SAGT sur le trafic, et à la définition des stratégies d'exploitation. Les données issues d'exploitation sont sous-utilisées, à la fois par manque d'outils, et surtout par manque de personnes compétentes pour les analyser. C'est typiquement un bon sujet pour des stagiaires ou des " thésards " en liaison avec des chercheurs. Un échange d'expériences en la matière permettrait peut-être de progresser. Á la DDE69 et à la ville de Paris, des petites applications ont été développées sous Excel ; à la DDE13, les fonctions d'analyse de données (édition de courbes, etc.) sont intégrées à l'applicatif MARIUS. Des progiciels tels que Mathlab ou Statistica sont également disponibles pour des analyses statistiques mais guère utilisés faute de compétences.

Le SIER dispose d'un outil " AES " d'analyse des données de trafic en temps différé, par ailleurs en cours de refonte ; un groupe de 7 personnes (animé par Gaëtan Juhel) est spécialisé dans les études d'ingénierie du trafic. On est loin d'avoir les mêmes effectifs chez les autres exploitants, en tout cas au ministère de l'Équipement.

Par ailleurs, signalons que le CERTU a organisé, le 23 septembre 99, une journée d'information technique consacrée aux outils de simulation de trafic urbain, destinée avant tout aux exploitants, mais pouvant aussi intéresser les architectes informatiques (qui sont d'ailleurs souvent les mêmes).

3.3.5. Main-courante

3.3.5.1. Saisie d'événements

Notion d'événement : Les événements jouent un rôle central dans les tâches d'exploitation ; un opérateur déclenchera telle ou telle action selon la nature des événements dont il a connaissance.

Cela dit, la manière dont il faut informatiser cette notion reste encore problématique, et dépend du type de PC. Contrairement peut-être aux données de trafic, dont l'informatisation est rendue plus " naturelle " par l'utilisation des équipements de recueil des données, les sources d'information sur les événements sont essentiellement humaines (patrouilles, appels d'urgence, analyse d'image vidéo, etc.).

- Au SIER, avec l'actuel système expert d'affichage PMV, l'opérateur détermine l'affichage sur un PMV en choisissant parmi les 2 ou 3 zones d'influence prédéfinies, qui peuvent influencer l'affichage pour un événement donné. Les événements saisis dans le système (bouchons, travaux, accident) sont déterminés par le besoin d'affichage sur les PMV. Le travail de l'opérateur est trop

orienté sur la commande des équipements, pas assez sur la vision globale de la situation de trafic. La future version du système (pour SIRIUS2) devra permettre à l'opérateur de définir les zones d'influence de manière beaucoup plus souple, selon le type d'événements en cours, et choisir automatiquement l'affichage en fonction du Plan de Gestion de Trafic.

- A MARIUS, une réflexion a été engagée pour reconcevoir la main-courante informatique d'exploitation : le principe central est de répondre à la question " qui a dit quoi à quel moment ? ". Le PC de la DDE13 doit mettre en avant son rôle de dispatching de l'information routière, en liaison avec les autres exploitation de l'agglomération marseillaise. En revanche, les actions d'exploitation (affichage PMV principalement) ont lieu de manière automatique, sauf dans des circonstances bien précises décrites dans un cahier de consignes et dans les PGT (en cours de définition et qui l'opérateur pourra consulter en ligne sous forme de pages Web).
- A Paris aussi, l'affichage PMV est automatique, pris en charge par le système d'exploitation du Boulevard Périphérique (IPER).

3.3.5.2. Mise en place d'une main-courante

La tenue d'une main-courante est de plus en plus importante pour les exploitants, notamment pour des raisons juridiques. Elle est parfois intégrée à l'application ; toutefois son utilisation dans le système est parfois peu pratique ou peu ergonomique. Il peut-être plus judicieux de tenir cette main-courante dans une application séparée, un peu pour les mêmes raisons que pour le référentiel SIG. Par exemple, la DDE69 a développé en interne (Access/VB) une main-courante informatique, ainsi qu'un planning des travaux prévus sur le réseau, en amont de l'application informatique. Ces deux logiciels seraient typiquement faciles à transférer à d'autres exploitants. Toutefois, la manière d'intégrer la main-courante au SAGT/1 doit être pensée a priori : voilà un bon exemple d'étude pour les années à venir (avis aux CETE intéressés !).

En outre, à défaut de normaliser les applications elles-mêmes, les interfaces peuvent être standardisées ; comme signalé plus haut un effort dans ce sens a été entrepris depuis quelques années (Datex, SEDT, LCR, DIASER, Alert/C...) a abouti à des résultats non négligeables, mais des efforts restent à faire, car les spécifications d'interface ne garantissent pas encore une parfaite interopérabilité, des variantes pas complètement compatibles pouvant en être implémentées ; en outre, des procédures et des outils de vérification de conformité à ces interfaces restent à développer. Par ailleurs, un effort de synthèse pour y voir plus clair sur l'ensemble des efforts de standardisation est encore nécessaire. Enfin, certaines interfaces ne sont pas encore du tout standardisées...

Il est utile de pouvoir proposer à l'opérateur une trace des dialogues entre le système et l'extérieur (communications fax, échanges de données, dialogue avec les équipements), sous forme de traduction " en français " de l'historique des actions. À la DDE13, un format particulier a été défini pour faire tenir cette trace sur une seule ligne.

3.3.6. Système de base et base de données

3.3.6.1. Évolution des logiciels de base

Ne pas confondre informatique et système (matériel + système d'exploitation et réseau) ; aujourd'hui l'essentiel du prix d'une informatique SAGT[9] se situe dans le logiciel applicatif (" métier ", spécifique ou pas). La durée de vie de l'applicatif est plus longue que celle du " système " (environnement informatique : matériel et logiciels de base).

[9] Qu'il ne faut pas confondre non plus avec l'ensemble du SAGT, qui comprend aussi le réseau et les équipements de terrain, plus coûteux que l'informatique d' un ordre de grandeur.

Attention aux évolutions incessantes des logiciels de base (système d'exploitation, réseau, mais aussi SGBD), qui imposent aussi d'avoir des administrateurs techniques compétents et « à jour ». Attention notamment aux " effets de bord " : le logiciel applicatif utilise des fonctions système ou base de données (ou de tout autre progiciel) qui ne sont plus disponibles dans la nouvelle version…

3.3.6.2. Temps réel

A cause de la disponibilité H24 et des exigences de temps réel, il y a beaucoup d'interactions entre l'applicatif et les logiciels de base ; le facteur temps (périodes et délais d'acquisition, puis d'élaboration, de mise à jour) pose souvent problème.

Par exemple, le système SURF de la Ville de Paris a un tel volume de données et des exigences de temps réel telles que les performances d'un SGBD " du marché " sont insuffisantes et qu'il utilise un système de fichiers spécifique. Les besoins de traitement en temps différé sont en revanche prévus d'être assurés par un SGBD classique.

De plus en plus, chaque module applicatif se doit d'avoir une fonction « exporter », notamment en raison de la prolifération des applications bureautiques ou utilitaires qui permettent d'exploiter des fichiers exportés.

Attention aux problèmes de synchronisation (alignement) des informations : deux postes opérateurs qui ont une copie locale de la base de données temps réel n'ont pas forcément la même vue ; de même, une base de données " miroir " n'est pas forcément à jour à tout instant. Étant donné le risque de désynchronisation dans une application répartie, il est préférable de modéliser les événements sous forme d'*états* que de *changements d'états*.

3.3.7. Aide à la décision

3.3.7.1. Problématique

L'aide à la décision (sous-entendu de l'opérateur, pas de l'usager) correspond en gros à l'informatisation des stratégies d'exploitation du trafic. Commençons par des questions que le groupe de travail s'est posé :

* Quelles sont les <u>stratégies d'exploitation possibles</u> dans un PC de gestion du trafic ?
- Quelles actions d'exploitation bénéficieraient en priorité d'une fonction d'aide à la décision ?
- Quels sont les points communs d'un exploitant à l'autre : quels sont les aspects génériques de l'aide à la décision ?
- Exemples d'action : diffusion de fax, déclenchement ou modification de plans de feux, affichages sur PMV, régulation des accès, régulation des vitesses, déclenchement d'une intervention sur incident, etc.
* L'aide à la décision cherche à traiter des activités de " haut niveau " de la gestion du trafic : <u>que doit-on informatiser ?</u>
- Jusqu'où doit aller l'aide à la décision : s'agit-il de bien présenter l'ensemble des informations utiles à l'exploitant ?
- S'agit-il de proposer automatiquement les décisions les plus pertinentes ?
- Comment présenter les conflits entre actions qui peuvent être déclenchées à un moment donné ?
- Quelles sont les types de décision d'exploitation qui peuvent être prises automatiquement, sans validation par un opérateur ?
* <u>Quand doit-on implémenter</u> un module d'aide à la décision : en même temps que l'on définit ses procédures de gestion de trafic, de manière à mieux intégrer l'outil et les pratiques, et à profiter de l'opération d'informatisation pour s'obliger à réfléchir de manière systématique ?
- Ou alors plus tard, lorsque les procédures sont déjà utilisées en exploitation depuis longtemps, de manière à en maîtriser la complexité ?
* <u>Faut-il utiliser un système expert ?</u>
- A la ville de Paris, le système SURF2000 s'appuie sur un système expert prototype développé avec l'INRETS (SAGE, qui est devenu CLAIRE après transfert industriel vers CGA, aujourd'hui opérationnel à Toulouse et Rennes), mais les règles ont été codées " en dur " dans l'application et il n'y a plus de système expert à proprement parler.

- A la DDE69, les Plans de Gestion de Trafic sont calculés de manière déterministe, et il n'y pas de module " expert ", bien que l'application fonctionne dans un environnement (G2) qui le permettrait.
- Au SIER, l'affichage des PMV est proposé à l'opérateur par un système expert.
- Notez que l'emploi de systèmes experts est un problème technique, pas fonctionnel.

* Quels sont les liens entre l'aide à la décision et les autres fonctions : communication avec le terrain, bases de données, analyse hors-ligne et évaluation ?

3.3.7.2. Contexte de l'aide à la décision

Les informations en entrée sont le trafic en cours (et passé récent), et le cas échéant une base de cas historiques, et des prévisions. L'aide à la décision fait un diagnostic de la situation en fonction d'une ou plusieurs stratégies d'exploitation, et de plans de gestion de trafic (PGT) prédéfinis, puis propose des actions, et éventuellement les met en œuvre. L'opérateur, selon le type de systèmes, et le type de stratégie, peut intervenir à différents niveaux :

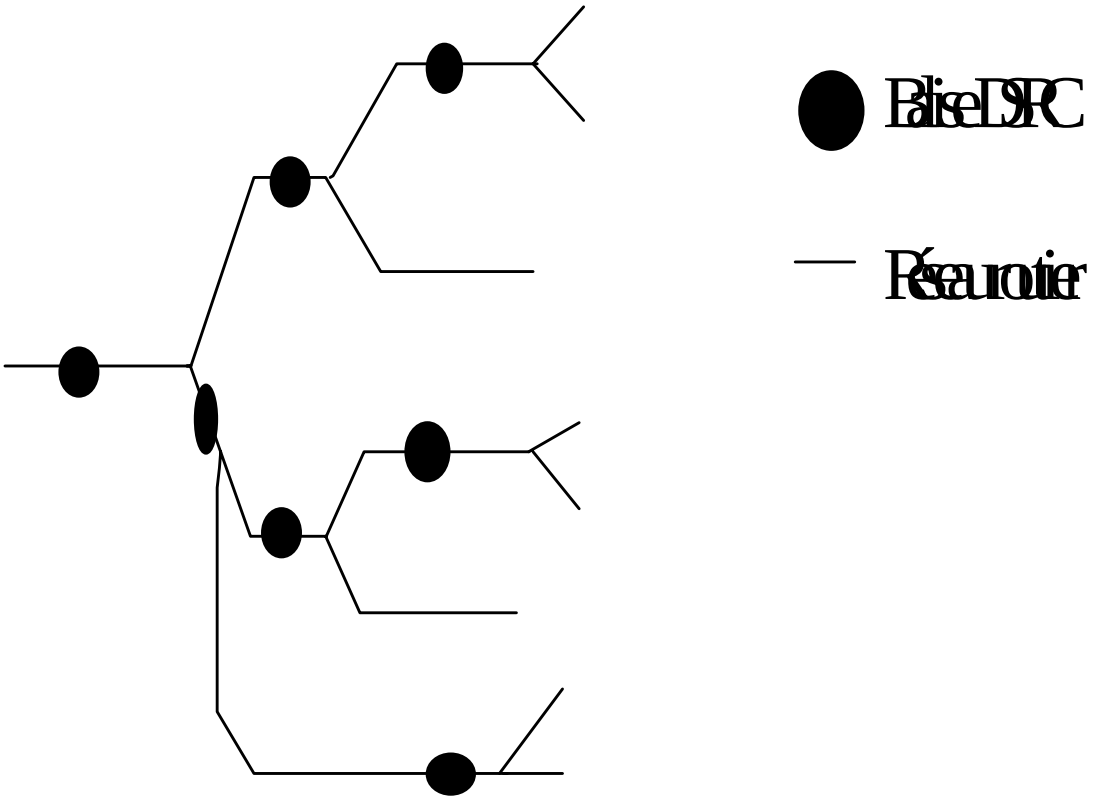
- action manuelle à partir du diagnostic fait par le système,
- validation d'une action proposée par le système,
- actions déclenchées automatiquement par le système.

Si plusieurs actions peuvent être déclenchées par le système, l'aide à la décision comprend aussi la gestion des conflits (priorités entre actions).

En outre, en temps différé, l'aide à la décision et les actions décidées par les opérateurs doivent pouvoir être évaluées, ne serait-ce qu'en fournissant une trace des actions proposées ou déclenchées automatiquement, voire en permettant de rejouer les actions (fonction " replay "), ou même de comparer les actions effectuées avec celles que l'automatisme aurait déclenchées " tout seul ". La trace doit pouvoir mettre en évidence les dysfonctionnements de l'automatisme (cf. importance croissante des aspects juridiques). Par exemple, la trace des actions dues à l'algorithme de régulation des vitesses à MARIUS est accessible facilement aux opérateurs.

Enfin, les stratégies de gestion de trafic évoluent (par retour d'expérience sur les actions d'exploitation), et les outils informatiques doivent permettre cette évolution.

Une représentation très schématique de la fonction " aide à la décision " est donc la suivante :



Quand une stratégie est-elle automatisée ?

- Cela dépend du nombre d'équipements à piloter, de la fréquence des actions, de la maturité de la stratégie d'exploitation (répétitivité, expérience acquise), de la faisabilité d'une commande manuelle, ainsi que de la formation et de la compétence des opérateurs.

Il faut estimer AVANT d'informatiser le volume de charge des opérateurs en actions, l'ergonomie, les performances souhaitables de manière réaliste : attention aux " usines à gaz " sans réflexion approfondie a priori sur l'utilisabilité de la fonction aide à la décision.

L'essentiel des actions des opérateurs consiste à diffuser l'information et à se coordonner avec les partenaires (fax, téléphone) ; à la limite, la commande des équipements devient secondaire (et gagne à être activée automatiquement ou de manière " presse-bouton ").

Voici une première ébauche de tableau essayant de répondre à la question :

stratégie	*automatisation*
régulation des vitesses	indispensable
temps de parcours	indispensable
régulation des carrefours à feux et des accès	indispensable
changement de plans de feux	validation manuelle possible
délestage	selon taille de réseau et complexité du PGT
diffusion d'informations (fax, messages)	validation manuelle
traitements des incidents	pas indispensable

3.3.7.3. Exemples

Ville de Paris :

Sur le Périphérique (IPER), l'affichage des temps de parcours est automatique, le reste est soit :

- « programmé » (manuel) :
  - travaux programmés (fermetures) : confirmation par l'opérateur
  - crues (fermeture des voies sur berge)
- sur événements : macro-commandes paramétrées par exemple en fonction du lieu d'un accident ou d'un match.
- " aléatoire " : purement manuel
- pour SURF : le traitement de la congestion est automatique (changement de plans de feux, favoriser / retenir sur un axe), mais manuel si capteurs en panne ou gros incident.

Des PGT (plans de gestion de trafic du SDER) ont été mis au point en commun entre la Ville et l'Équipement notamment, pour l'accès au " Grand Stade ".

Un cahier de consignes (auquel a participé aussi JM Chauvin du CETE de Rouen) existe pour préciser les actions en cas de coupures du réseau autoroutier d'Ile-de-France ou de fermetures (nocturnes) pour travaux.

DDE69 :

Dans CORALY, toutes les actions sont validées manuellement. Ce sont des actions de délestage : il n'y a pas de régulation comme dans SURF ou MARIUS. Le trafic (flux de transit) est très sensible (" obéissant ") aux actions de délestage, en particulier lors des vacances d'été ; l'opérateur doit en permanence ajuster le " robinet " commandé par les sites directionnels et les PMV. En outre, une application de calcul de temps de parcours vient d'être mise en place.

Dans CORALY, un événement est situé au niveau d'une branche, chaque événement est susceptible de déclencher un plan d'action qui est soumis à l'opérateur avant exécution. C'est l'opérateur qui gère les conflits entre les actions. Une des difficultés est " l'explosion combinatoire " des actions possibles pour le système. En effet, les actions sur les PMV ou SDV, les barrières, les messages fax, dépendent de la branche du réseau, de la configuration de trafic (prédéfinie selon le calendrier), de l'importance et de la nature de l'événement. Les plans d'actions ont été précalculés (il n'y a pas de système expert pour proposer en temps réel le " bon " plan d'action).

L'efficacité de l'aide à la décision dépendra du nombre de plans d'actions déclenchés et proposés pour validation à l'opérateur, ce que nous apprendra l'expérience opérationnelle.

DDE13

Toutes les actions sont automatisées dans MARIUS. Les stratégies d'exploitation sont d'une part la régulation des vitesses, d'autre part l'affichage sur PMV ('bouchon', ou 'accident' si saisie de l'opérateur de l'événement correspondant). Le bouchon est détecté en comparant les heures perdues actuelles et prévisionnelles (par différence entre vitesses constatées et vitesse libre, multipliées par le débit).

Des " fiches réflexe " (équivalent du cahier de consigne) ont été étudiées ; elles ne sont pas encore informatisées. Ces fiches seront sans mises sous forme de documents HTML accessibles sur le poste de l'opérateur, de type " aide en ligne " (pour les exploitants et les mainteneurs) : cet " intranet " de type " aide en ligne " évitera les problèmes liés au volume et à la mise à jour des documents papiers.

Les informations de consigne sont de deux types : permanentes et occasionnelles. Les utilisateurs sont de 3 sortes : opérateurs Police, DDE, et mainteneurs. Les alarmes contiennent les informations suivantes : nature de l'événement, conséquences possibles sur le système, action à conduire.

3.3.7.4. Suites à donner

Compte tenu de la complexité du domaine, et surtout de la diversité des stratégies possibles selon le type de réseau exploité, il semble impossible de travailler sur un modèle “ générique ” de l’aide à la décision, en tout cas dans ce groupe de travail. Pour aller plus loin, il faudra sans doute travailler à la consolidation des documents existants (besoins, CCTP, spécifications, modèles, implémentations), passer en revue les divers types de stratégie, avant de formuler quelques recommandations. Le premier travail concernerait sans doute la terminologie, qui permettrait de comparer les applications existantes, car elles utilisent toutes un “ jargon ” plus ou moins spécifique.

4. Capitalisation sur l'informatique des SAGT

4.1. Une démarche globale

Un grand nombre d'éléments issus d'un projet de SAGT sont potentiellement réutilisables pour d'autres SAGT, et en particulier les documents produits à chaque phase du cycle de vie (éventuellement après mise à jour). Pour faciliter cette mise à jour et la réutilisation, on recommande aux exploitants d'exiger la fourniture des documents sous-traités sous forme de fichier bureautique.

Il est intéressant tout d'abord de collecter et consolider des documents issus des projets, étape après étape, à partir des informations que chaque exploitant voudrait bien mettre en commun. Il sera évidemment plus facile de partager des documents " amont " (par exemple de définition des besoins) que des artefacts en aval du projet (en particulier du code source ou des exécutables), ne serait-ce que pour des problèmes de propriété intellectuelle - qui dépendent non seulement des exploitants, mais aussi bien sûr de leurs fournisseurs. D'autre part, cette démarche est plus facile à mettre en place entre exploitants relevant d'une même " famille ", par exemple entre DDE au sein du ministère de l'Équipement, ou entre exploitants appartenant à un même groupe de transports collectifs.

Une étude - confiée fin 98 à la société Euriware par le CERTU [14] - présente un état des lieux sur les systèmes informatiques de SAGT/1 dans les services du ministère de l'Équipement, dresse un premier diagnostic et propose des recommandations en vue de favoriser la réutilisation et la capitalisation concernant l'informatique des SAGT/1, qui ont été largement prises en compte par le CERTU et la DSCR et servent à définir notre programme de travail. Certaines de ces recommandations peuvent intéresser plus largement d'autres maîtres d'ouvrages de SAGT que la DSCR. Une véritable démarche systématique de réutilisation reste à mettre en place ; pour les SAGT/1, le CERTU a proposé plusieurs actions à la DSCR.

4.2. Réutilisation de logiciels

La réutilisation de composants peut-elle constituer un frein à l'innovation ? La situation est problématique quand c'est le réalisateur qui propose d'utiliser un progiciel ou des " composants " issus de projets qu'il a réalisés précédemment, en vue de réduire les coûts ; le client (l'exploitant) n'a pas toujours les moyens techniques et financiers de refuser. Ce n'est pas forcément le cas en revanche si c'est l'exploitant qui exige dans ses spécifications que le réalisateur utilise des composants du marché (en pratique essentiellement des progiciels), ce qui demande une bonne maîtrise technique de la part de l'exploitant. Dans tous les cas, le client doit exiger une transparence (spécifications claires) sur ces composants, qui doivent être réceptionnés dans les mêmes conditions que les autres composants développés spécifiquement pour l'application. En outre, il faut aussi faire attention à la propriété intellectuelle, et éviter de se retrouver piégé avec un réalisateur qui revend un produit dont il n'est pas propriétaire ! Enfin, ce choix peut poser des problèmes d'évolution. Dans le cas de progiciels, tout dépend si le progiciel est imposé en interne (car déjà utilisé dans l'organisation de l'exploitant, bureautique ou messagerie pare exemple), choisi par l'exploitant, ou proposé par l'entreprise dans sa réponse. [13]

Les systèmes informatisés d'aide à la gestion de trafic intègrent de plus en plus de fonctions, et de manière à limiter la complexité des logiciels, la tendance est plutôt à ce que le système soit composé de logiciels (modules) communiquant entre eux (interfacés).

Dans le contexte actuel, la réutilisation de logiciels d'aide à la gestion de trafic est difficile à envisager. Le développement des logiciels n'est pas a priori du ressort de l'exploitant, et si des progiciels " métier " répondant aux besoins des exploitants apparaissent, ce sera à l'initiative des industriels.

Toutefois, les grands maîtres d'ouvrages (DSCR pour les DDE, groupes de TC pour les réseaux urbains) peuvent envisager de déployer un logiciel unique de SAGT.

En revanche, les SAGT font appel à un certain nombre d'applications " périphériques " en amont ou aval du système " industriel ", souvent interfacées par échange de fichiers texte selon un format spécifique, et parfois développées en interne. Ce type d'applications semble plus apte à devenir un produit réutilisé par plusieurs exploitants (analyse de données, référentiel géographique, main-courante, banc de test).

L'article [10] présente la démarche engagée par l'état américain de Géorgie pour partager les logiciels développés pour son centre d'exploitation du trafic. En pratique, une démarche similaire ne semble pas facilement transposable en France ; elle a d'ailleurs posé des problèmes aux États-Unis. Il semble que la cession de code à d'autres exploitants, outre les problèmes de propriété intellectuelle qu'elle pose, ne puisse pas facilement être mise à profit ; une grande partie du code concerne l'IHM et les communications avec le terrain, et est pour l'essentiel spécifique à un site. En interne au ministère de l'Équipement, il est envisageable qu'un logiciel développé pour une DDE soir repris et adapté pour une autre, à condition toutefois de s'appuyer sur une clause de propriété intellectuelle convenablement rédigée (le CERTU étudie ce point).

L'échange de cahiers des charges ou de spécifications semble en tout cas plus facile à mettre en œuvre et pouvoir apporter des bénéfices plus immédiats.

4.3. Progiciels

Le CERTU a confié une étude concernant « l'intégration de progiciels dans un système d'aide à la gestion de trafic » à la société Sema Group. Cette étude est destinée aux exploitants de tous types de réseaux de transport (route, TC, etc.), et à leurs fournisseurs. Elle présente une démarche générale d'intégration de progiciels, liste les progiciels horizontaux (base de données, superviseurs, systèmes expert, etc.) utiles pour l'aide à la gestion de trafic, ainsi que quelques progiciels " métier ", puis détaille 4 types de progiciels sous forme de fiche. Il s'agit d'une étude " légère " qui demande à être complétée... en fonction des éventuelles réactions des lecteurs. Le document est disponible sous forme électronique sur le serveur Web du CERTU, ou en rapport d'étude papier [13].

Dans ce rapport d'étude, il est préconisé d'établir des priorités dans ses besoins lors de l'établissement d'un cahier des charges pour un projet d'application s'appuyant sur un ou plusieurs progiciels, car bien souvent il faudra faire quelques concessions pour s'adapter à un produit existant. Du point de vue de la consultation, cela oblige à procéder en 2 temps, soit en faisant une étude préalable afin de définir ses priorités en fonction des progiciels existants (et en autorisant éventuellement des options ou des variantes au CCTP), soit en lançant un appel d'offres sur performances qui permet de négocier les offres après une phase d'auditions.

4.4. Serveurs de documents

Une des priorités pour le CERTU en 99 est d'établir un serveur de documents sur l'informatique des SAGT de VRU qui facilite le retour d'expériences ; il conviendra bien sûr de valider le niveau de diffusion de ces documents en fonction des droits de chacun. Le cahier des charges du serveur a été rédigé par le CERTU et largement diffusé pour avis. Les premiers services concernés en tant que fournisseurs de documents sont le SIER, la DDE69 et la DDE13, qui participent à l'élaboration du cahier des charges et vont recenser les documents (électroniques) qu'ils pourront mettre à disposition après une éventuelle mise en forme. D'autres services sont sans doute aussi en mesure de fournir des documents intéressants (DDE31, DDE26, DDE38, DDE59, DDE974, DDE67, etc.), et pourquoi pas d'autres exploitants (villes, sociétés d'autoroute) pourraient à terme aussi fournir ou retirer des informations d'un tel serveur.

Le CERTU est en train de développer la version initiale de ce serveur de documents qui permettra d'accéder à l'ensemble des documents techniques que l'on aura pu mettre à disposition, en distinguant les documents publics accessibles à tous et des documents accessibles seulement à une liste fermée d'utilisateurs (typiquement, les actuels destinataires de ce compte-rendu) ; les résultats sont attendus pour l'automne 99. Le cahier des charges a fait l'objet d'une consultation ; la proposition de la société IFATEC a été retenue ; un prototype sera développé à partir de l'outil " cyberthèque " mis au point par cette société. Il s'agit d'une base de documents réalisée avec le progiciel Lotus Domino ; le serveur sera installé au CETE de Bordeaux.

4.5. SAGT générique

Au-delà de l'échange de documents entre exploitants, une étape supplémentaire vers une meilleure réutilisation du savoir-faire en informatique pour les SAGT serait de pouvoir spécifier les fonctions communes à plusieurs sites. Il est envisagé de mener un tel travail pour quelques fonctions " de base " des SAGT de VRU (ou SAGT/1), comme le recueil de données et la commande des équipements, ou le référentiel. Toutefois, d'une part ce travail nécessite de mieux cerner les besoins, d'autre part certaines fonctions de " haut niveau " (stratégies d'exploitation) ne font pas partie de ce noyau de base à analyser.

Du CdC pourra être extrait un CdC " générique ", plus directement réutilisable ; la mise en commun de plusieurs CdC " génériques " pourra alors permettre de proposer un " CdC-type ". Compte tenu de la diversité des besoins d'un exploitant à l'autre, il est plus réaliste de chercher à construire des " briques " de CdC par domaine fonctionnel, que de vouloir construire un seul Cahier des Charges.

4.5.1. Cahier des charges " générique " pour un frontal, une MAO, un référentiel technique de SAGT/1

Á l'occasion de l'étude de définition en cours à la DDE69 pour la refonte du frontal de communication avec le terrain et la mise en place d'une maintenance informatisée, un document " générique " sera établi par le titulaire du marché (ISIS/CIQS) à partir du cahier des charges de cette application. Il reprendra le document de définition des besoins élaboré pour la DDE du Rhône, en essayant autant que possible d'en " gommer " les aspects spécifiques. L'étude sera disponible avant fin 99, sans doute sous forme de rapport technique.

4.5.2. Cahier des charges " idéal " d'un SAGT/1 selon MARIUS

Dans un esprit un peu similaire, Gildas Lemaître rédige un cahier des charges " idéal " de SAGT/1, à partir de son expérience acquise depuis plusieurs années avec le système MARIUS de la DDE13. Le document sera disponible à la fin de l'année.

De manière générale, si vous êtes sur le point de lancer une étude de définition pour une application informatique d'un SAGT/1, il est possible d'adopter la même démarche, et d'établir pour un prix à peine plus élevé un document générique plus facilement réutilisable par d'autres services. Pensez-y...

4.5.3. Études à lancer prochainement par le CERTU

Trois sujets d'études devraient encore faire l'objet d'une consultation en 99 :

- référentiels de données pour les SAGT/1 : l'étude comprendrait un état des lieux établi à partir de l'expérience des systèmes existants, des recommandations plus générales en matière de référentiels et de configuration des applications, et enfin une ébauche de dictionnaire des principales notions (données ou fonctions) d'un SAGT/1.

- évaluation des logiciels pour les SAGT/1 : l'étude partirait d'un état des lieux concernant la satisfaction des utilisateurs des systèmes existants, présenterait des recommandations générales en

matière d'évaluation de logiciel applicatif, et formulerait enfin des propositions concrètes à court et moyen termes.

- propriété intellectuelle : il s'agit d'une part de valider les documents (clauses de CCAP/pi, contrat de cession de droits, charte d'utilisation) permettant le fonctionnement du serveur de documents SAGT/1 dans les meilleures conditions, d'autre part de proposer une clause-type pour la propriété intellectuelle des logiciels

4.6. Site Internet

Le CERTU dispose depuis quelques mois d'un site Web (www.certu.fr). Les comptes-rendus du groupe de travail sont disponibles sur ce site, ainsi qu'un certain nombre d'autres documents, à la page systèmes d'information de la rubrique " circulation - trafic "[10]. Si vous disposez d'un accès à l'Internet, et si vous vous êtes connecté(e) à ce site, nous vous remercions de nous indiquer comment améliorer le contenu et mieux répondre à vos éventuelles attentes, et ... d'être indulgents pour ce premier jet !

De nouvelles études et documents de travail devraient peu à peu être disponibles à partir de ce site.

[10] http://www.certu.fr/tribfra/travcour/cirtrav/CADRE.HTM

5. Conclusions et perspectives

L'informatique de systèmes d'aide à la gestion du trafic est un domaine encore peu documenté, c'est pourquoi nous avons jugé utile de publier ce premier document, malgré ses lacunes évidentes. Ce document a permis de donner un panorama des projets de SAGT/1 ainsi que des fonctions couvertes. Il complète ainsi le chapitre correspondant du guide SDER [5]. On y a largement souligné l'importance de l'informatique dans un projet d'exploitation routière, qui doit être placé au moins au même niveau par exemple que les équipements et les transmissions, et qui joue un rôle pivot entre les stratégies d'exploitation (fonctions) et le système technique (outil). En outre, insistons à nouveau sur le fait que le logiciel est une activité consommatrice en ressources humaines, et donc en compétences qui sont le plus souvent à créer ou développer.

Ce panorama n'a pas l'ambition d'être complet et même pour les points que nous avons essayé d'approfondir, nous sommes loin d'avoir fait le tour de la question. Il faudra notamment travailler sur des points souvent peu traités dans les projets (mais importants !) :

- gestion des risques,
- plate-forme de développement et de tests,
- documentation,
- intégration de l'informatique au reste du système d'aide à l'exploitation (équipements de terrain, transmissions).

Parmi les fonctions qu'il faudra étudier particulièrement, mentionnons les référentiels (base de données de configuration et SIG), la main-courante (intégration de la notion d'événement dans l'exploitation en milieu urbain), l'analyse en temps différé.

Nous avons enfin indiqué dans la partie précédente un certain nombre d'actions que le CERTU a proposé, en cohérence avec les autres volets techniques de l'exploitation routière dont la maîtrise d'ouvrage est assurée par la DSCR, et dont les résultats apporteront également une contribution à la maîtrise des systèmes et pourront compléter une deuxième version de ce document.

Pour conclure, la principale source d'amélioration de cette première version ne peut venir que des personnes impliquées dans les projets de SAGT/1 eux-mêmes, et nous espérons que la lecture aura un intérêt suffisant pour susciter des remarques, critiques et suggestions en vue de l'améliorer.

6. Annexe 1. Les étapes dans Cheops.

Cf. [1]

Le projet du maître d'ouvrage

- L'étude d'opportunité
- L'organisation de l'opération
- La communication
- Le management

Les projets des maîtres d'œuvre fonctionnel et technique

- La conception fonctionnelle détaillée
- La conception des jeux d'essai fonctionnels
- La conception, la réalisation et les tests techniques
- L'exécution de la recette fonctionnelle et technique
- (expérimentation en site pilote)
- La maintenance

Le projet du maître d'œuvre d'accompagnement

- L'analyse des attentes et l'étude d'impacts
- Le plan de conduite du changement
- La préparation de la diffusion
- La préparation de l'expérimentation en site pilote
- L'expérimentation en site pilote
- La diffusion

7. Annexe 2. Traduction d'extraits d'un guide américain de recommandations concernant l'informatique pour les systèmes de gestion de trafic. [12]

7.1. Les bonnes pratiques.

* autant que possible, utilisez des produits existants

* documentez le projet complètement, en allant jusqu'à la mise en exploitation et la maintenance

* construisez une équipe

* jouez la transparence : communiquez en permanence avec les prestataires et les autres membres de l'équipes

* faites un audit indépendant de vos coûts et délais

* documentez les besoins et utilisez les comme référence pour les autres activités (recette, budgets et délais, conception, etc.)

* trouvez un compromis entre besoins, coûts et délais

* utilisez une procédure formelle de sélection des offres, comprenant l'évaluation du processus d'ingénierie logicielle

* identifiez les problèmes, notez-les, suivez-les

* suivez les dépenses et l'avancement

* gérez les risques : détectez-les, résolvez-les. La gestion des risques fait partie intégrante de l'acquisition du logiciel, et se mène en coopération avec les prestataires, et les autres membres de l'équipe

* indiquez les critères d'acceptation du système dans le contrat

* définissez un plan de recette et conformez-vous y (ce plan peut être proposé par le prestataire et soumis à votre approbation)

* explicitez clairement les questions de licences et de propriété intellectuelle.

* mettez en place un programme de formation pour l'utilisation et l'exploitation du système

* définissez votre stratégie de support du système

* assurez-vous que le logiciel soit conforme aux standards nationaux ou européens en vigueur

7.2. Les points clés (du volume 1).

7.2.1. De la nature du logiciel

* l'acquisition d'un logiciel n'est pas un projet comme les autres

* les dérapages de calendrier, les dépassements de budget, le manque de visibilité sur le logiciel et son développement sont monnaie courante

* pour gérer un projet de développement de logiciel, il faut donc des approches différentes

* des techniques de management éprouvées existent

* les problèmes rencontrés lors du développement de logiciels de systèmes de transport intelligents sont semblables à ceux qui surviennent pour d'autres types d'applications

7.2.2. contexte de l'acquisition de logiciel

* le logiciel fait partie d'un système, et son acquisition se situe dans un processus global d'acquisition

des perceptions différentes

* les secteurs publics et privés ont une perception très différente du logiciel

* le client (public) et le fournisseur (privé) ont souvent à tort l'impression qu'ils sont dans une situation où si l'un y gagne l'autre y perd ; en fait ils sont sur le même bateau

Les thèmes à traiter pour réussir son projet

* construisez votre projet autour de quelques thèmes récurrents dans l'acquisition d'un logiciel

- humains (partenariat)

- managériaux (approche de l'acquisition)

- techniques (liés au système final)

* les thèmes permettent de répondre aux problèmes les plus fréquemment rencontrés, et de montrer en quoi le logiciel est différent

7.3. Les points-clés (volume 2)

7.3.1. séquence d'activités

- il n'existe pas « une » séquence d'activités qui « marche » pour tout type de projet

- la définition des besoins et le choix du type de contrat sont déterminants

- plusieurs activités se déroulent en parallèle et interagissent

7.3.2. construire une équipe

- construisez une équipe de professionnels aux compétences complémentaires

- si possible, essayez de récupérer ces compétences en interne, sinon passez des contrats

- intégrez l'entreprise prestataire dans l'équipe

7.3.3. planifier le projet

- écrivez un bref document qui présente les points-clés de votre démarche

- le plan du projet est un document qui vit ; on y ajoute des informations, on le modifie

7.3.4. les besoins

- la définition des besoins est essentielle

- tous les membres de l'équipe y participent

- documenter les besoins de manière formelles, en contrôlant les versions

- décrivez le quoi, pas le comment

- n'en demandez pas trop

- les facteurs de qualité et la capacité du système à pouvoir prendre en compte des changements font partie des exigences

- dès que possible une fois le contrat signé, faites une revue du document de définition des besoins avec votre équipe (et le prestataire). Signez alors ce document et faites-le évoluer en contrôlant les versions. Cette revue doit faire partie du contrat

- définissez un « noyau dur » stable dans vos besoins

- traitez les questions liées aux besoins au fur et à mesure, dans un processus de gestion des exigences

- utilisez le prototypage rapide pour définir l'IHM

- utilisez les besoins comme base de départ pour estimer les ordres de grandeurs, le calendrier et les coûts, ainsi que les activités de conception et de développement et le cahier de recette

7.3.5. acheter ou faire ?

* si possible, achetez le logiciel plutôt que de le développer

* sinon, essayez d'intégrer des logiciels existants dans le système que vous développez

* l'achat de logiciel n'est pas exempte de risques, mais ils sont maîtrisables

* avant d'acheter un progiciel, comprenez bien le produit et les conséquences qu'entraîne son utilisation

7.3.6. le choix du type de contrat

* l'approche utilisée en génie civil (conception / appel d'offres / réalisation) n'est pas appropriée

* travaillez en amont avec le responsable des marchés ou le juriste pour explorer toutes les options

* un contrat au forfait peut ne pas vous apporter toute la flexibilité dont vous aurez besoin. Envisagez d'autres alternatives

* ne pas travailler en régie (« cost-reimbursement ») si en fait vous avez un budget fixé à l'avance

* ne pas mixer forfait pour le matériel et régie pour le logiciel

* le forfait est plutôt adapté à l'acquisition d'un progiciel

* aucun type de contrat ne vous dispense d'un suivi actif

7.3.7. l'environnement informatique

* identifiez l'environnement dans lequel le système fonctionnera - y compris les interfaces avec l'existant -

* ne contraignez pas la conception en spécifiant prématurément le matériel ou l'OS

7.3.8. la propriété intellectuelle

* qu'ils aient ou non une signification juridique précise, des termes comme « accord de licence » ou « propriété » sont interprétés différemment d'un personne à l'autre

* avant de signer le contrat, mettez-vous d'accord sur les droits de propriété intellectuelle liés au logiciel

* passez en revue la terminologie du contrat, assurez-vous que vous le contrat décrit clairement vos intentions

* explicitez la manière dont le code source et objet doivent être fournis

* le cas échéant, ayez recours à un juriste spécialisé

* réfléchissez à vos besoins réels avant d'exiger certains droits

7.3.9. calendrier du projet

* deux pratiques courantes mais à proscrire :

- le calendrier est établi indépendamment des besoins

- le calendrier est irréaliste

* définissez un calendrier réaliste qui corresponde à la définition des besoins et que vous êtes prêts à tenir. Ne prenez pas des estimations optimistes, vous ne les tiendrez pas. Des estimations pessimistes sont souvent plus réalistes

* au cours du projet, ajustez conjointement le calendrier, les exigences et le budget

* partir sur un calendrier réaliste est un des meilleurs moyens de réduire le coût complet de développement d'un projet

* définissez des jalons clairement (fait / pas fait ; on continue : oui / non)

* obtenez des estimations indépendantes, autant que possible, de la part du prestataire ou d'experts informatiques de votre équipe. Examinez les différences entre ces estimations

* ré-estimez le calendrier au cours de l'avancement

7.3.10. recette

* planifiez de manière formelle la procédure de réception, y compris la documentation, avant le lancement de la consultation. Faites apparaître cette procédure dans le contrat

* prévoyez dans le calendrier que les activités liées aux tests et à la recette commencent dès que possible après signature du contrat et progressent pendant toute la durée du développement

* basez votre recette sur la définition des besoins

* faites toutes sortes de tests rigoureux ; des tests élémentaires de bon fonctionnement ne suffisent pas

* menez le test en équipe

7.3.11. formation, exploitation, maintenance

* planifiez le support suffisamment tôt, et faites le apparaître au contrat. Les activités de support doivent être préparées en parallèle des activités de développement

* prévoyez assez de budget pour le support. Sur la vie du système, le support consomme souvent plus de ressources que le développement

7.3.12. gestion de projet

* le client a un rôle actif à jouer même après la signature du contrat

* les revues de projet et de documentation, ainsi que les mesures quantitatives permettent d'améliorer la visibilité dans le projet et le processus de développement

* faîtes apparaître vos procédures de gestion de projet dans le contrat

* assurez-vous que le contrat vous permette un contact direct et des communications avec chacun des prestataires du logiciel

* exigez des entreprises répondant à la consultation qu'ils décrivent leur processus de développement, en particulier le plan d'assurance qualité logicielle. Suivez ce plan lors des revues de projet et par des vérifications ponctuelles

* gérez les attentes des acteurs qui ne travaillent pas directement dans le projet

gestion de configuration du logiciel

* une procédure (formelle) de gestion de configuration est essentielle

* définissez des situations de référence (photographie de tout ce qui concerne le logiciel) et utilisez une procédure lorsque des changements doivent y être apportés

* assurez-vous que le prestataire pour la réalisation du logiciel a des procédures claires de gestion de configuration et s'y conforme

7.3.13. gestion des risques

* durant tout le projet, ayez un œil sur la gestion de risques

* les étapes d'une gestion des risques sont l'identification des risques, l'analyse, la planification, la résolution et le suivi

* la gestion des risques est plus efficace si menée en équipe, avec le client et le prestataire

* la gestion des risques ne marche que l'ambiance de travail permet aux gens peuvent pointer sur des risques sans arrière-pensée

7.4. Que mettre dans le plan du projet

* description du projet : court résumé (mission, objectifs, étendue de l'application)

* justification : pourquoi acquérir le logiciel ? (économies, réduction de la congestion, améliorer le service, etc.)

* calendrier : étapes majeures

* rôles : qui gère le projet ? taille et composition de l'équipe projet ? Quelles organisations sont concernées ? Quels sont les personnes à contacter dans chaque organisation ? Quel sont leurs rôles respectifs ? Qui sera responsable de la formation et de l'accompagnement ? De la maintenance ? Au besoin, mettre un organigramme

* estimations des sources de financement

* environnement : Où le développement sera-t-il effectué ? Où sera installé l'applicatif ? De quels outils ou environnement a-t-on besoin ?

* stratégie d'acquisition : Le système sera-t-il développé à partir de rien ? Dans quelle mesure utilisera-t-on des progiciels du commerce ? Peut-on réutiliser des parties d'autres projets ? Le système sera-t-il construit incrémentalement, en plusieurs phases ? Un prototype sera-t-il développé ? Comment les progiciels seront-ils intégrés ?

* environnement : y a-t-il des systèmes existants auxquels s'interfacer ? Des capteurs, des équipements de terrain ou embarqués ? Quels sont les liens avec d'autres organisations ou collectivités ?

* standards : quels sont les standards auxquels il faut se conformer ?

* risques : comment les risques seront-ils gérés ? Les risques-clés ont-ils été identifiés ?

* contrat : quel travail sera réalisé en interne ? Qu'est-ce qui sera sous-traité ? Des consultants seront-ils embauchés ?

* type de contrat : quelles options ont été étudiées ? Forfait , régie ? spécification / réalisation ou clé en mains ? Combien de contrats sont-ils nécessaires ?

* gestion des contrats : comment seront-ils suivis ?

* les utilisateurs : Qui utilisera le système ? Qui l'administrera ? Qui le maintiendra ? Quelles seront les ressources en personnels et financières correspondantes ?

* recette : quelles seront les conditions pour accepter le système livré ?

* formation : comment les utilisateurs seront-ils formés ?

* maintenance : comment le système sera-t-il maintenu une fois réceptionné ?

* contraintes : quelles sont les « réalités » à prendre en compte ?

7.5. Que mettre dans un document de définition des besoins

7.5.1. besoins fonctionnels

* de quoi le système doit-il être capable

* chaque fonction est écrite sous la forme d'une phrase comprenant le mot 'doit' et être testable. Exemple : 'le système doit afficher une information de bouchon sur les PMV'

* précisez si la fonction est manuelle, automatique ou semi-automatique (ex. le système doit choisir une message et l'afficher ; l'opérateur doit taper le message à afficher ; l'opérateur choisit parmi plusieurs messages proposés par le système lequel afficher)

* description des exigences ergonomiques pour l'IHM. Pour préciser l'IHM, il est recommander de prototyper

* des algorithmes ou des équations

* compatibilité an 2000

* conformité à des standards ou à une architecture

7.5.2. performance

* temps de réponse, sous forme de moyenne / écart-type ou de quantiles (ex. 'le système doit avoir une temps de réponse moyen de 3 secondes, avec 90% des temps de réponses inférieurs à 5 secondes)

* charge (ex. capacité à traiter simultanément un certain nombre de capteurs), y compris mode dégradé en cas de charge trop importante

* débit (ex. nombre de transactions par minute)

* capacité ('le système doit stocker 30 jours de bulletins d'information')

* taux de fausses alertes, y compris les algorithmes pour déterminer ces taux

* précision (y compris algorithmes)

* disponibilité et maintenabilité (MTBF, MTTR)

* sécurité

* innocuité

7.5.3. interfaces

* depuis / vers les équipements de terrain

* depuis / vers les écrans

* depuis / vers les utilisateurs

* depuis / vers d'autres systèmes (y compris existants)

* depuis / vers d'autres organisations ou collectivités

* entre sous-systèmes (par exemple entre bus et SAE)

* entre composants (modules) du logiciel (par exemple entre l'algorithme de détection d'incident et le recueil de données)

7.5.4. entrées

* sources (humaines ou machines)

* fréquence (débit)

* domaine de validité, unité de mesure

* leur donner un nom (identifiant)

7.5.5. sorties

* en temps réels (ex. alarme) ou non (bulletin à imprimer)

* destination (équipements ou utilisateurs)

* fréquence (débit)

* domaine de validité, unité de mesure

* leur donner un nom (identifiant)

7.6. Suggestions pour une revue des exigences

* clarifier les exigences ambiguës ou floues

* supprimer les exigences contradictoires ou incohérentes entre elles

* compléter les besoins manquants

* remplacer les besoins existants par de meilleures alternatives identifiées par la suite

* supprimer les exigences inutiles ou impossibles à tenir, ou réduire leur priorité

* donner des niveaux de priorités aux exigences

* éliminer les besoins peu prioritaires ou estimés chers

* simplifier les exigences

* prendre des options moins chères envisageables

* reporter des exigences moins prioritaires à des versions ultérieures

7.7. Que prendre en compte pour l'environnement informatique.

* interfaces à d'autres applications ou systèmes existants (avec quelles applications le système à développer doit-il interopérer, y compris supervision ?)

* réseaux de communication (protocoles, bande passante, liaisons louées ou commutées, type de réseau)

* interfaces avec des applications futures déjà prévues

* utilisateurs et IHM

* locaux (bureaux ou extérieur, éclairage, espace disponible, alimentation)

* sécurité (locaux fermés, protections logicielles, passerelles pare-feux, etc.)

* performance

* standards

si nécessaire seulement (au moins du point de vue des standards) :

* matériel

* système d'exploitation

* SGBD

* langage de programmation, méthodes et outils de développement

7.8. droits de propriété intellectuelle.

Généralités

* qui possède le logiciel ? A quels droits cela correspond-il ?

* qui possède le copyright ? A quels droits cela correspond-il ?

* doit-on rajouter en commentaire dans le code une note concernant le copyright ? Qui doit détenir le copyright ?

droits du client

* le client peut-il faire d'autres copies du logiciel pour ses besoins dans ce projet ? Dans d'autres projets ?

* Le client peut-il distribuer des copies du logiciels à d'autres services ?

* Le client peut-il distribuer des copies du logiciel à d'autres organisations, ministères ou collectivités ? Le bénéficiaire de la copie est-il autorisé à modifier ou améliorer le logiciel, ou seulement à l'utiliser ?

* Le client peut-il distribuer des copies gratuites ? Peut-il les faire payer ?

* Le client peut-il modifier le logiciel ou développer des extensions ?

* Le client peut-il fournir le source à d'autres entreprises ou les autorise à le modifier ?

* Pour les 6 points ci-dessus : quelles parties du code peuvent être copiées : le source ? le code objet ? la documentation ? combien de copies peuvent être faites ?

* Le client a-t-il des droits sur des versions ultérieures développées par le prestataire ?

* Y a-t-il des parties du logiciel nécessaire au bon fonctionnement du système qui ne sont pas couvertes par les accords de licence ? Cela peut comprendre l'OS, un SGBD, un SIG, une carte numérique, pour lesquels un nombre de copies autorisées peut être différent que pour le reste du logiciel.

* Le client aura-t-il accès au code source ? Sous forme de listings ou de fichiers compilables ?

* Le client aura-t-il accès aux outils et environnement de développement utilisés pour compiler, configurer, tester le logiciel, etc. ?

* Le client aura-t-il des droits sur tout le support de formation ?

* Le client aura-t-il des droits sur l'environnement d'exécution ou ces droits doivent-ils être acquis séparément auprès d'autres fournisseurs ?

* Le client aura-t-il accès à la documentation décrivant les formats de BD et les protocoles d'interfaces ?

* Combien de machines peuvent héberger une copie du logiciel ? Combien peuvent exécuter le logiciel ?

* Combien de machines peuvent exécuter ou accéder au logiciel simultanément ?

* Combien d'utilisateurs ont-ils le droit d'exécuter logiciel ? Simultanément ?

* Le logiciel peut-il être exécuté à travers un réseau ?

droits du fournisseur

* Le réalisateur peut-il fournir le logiciel à d'autres ? Peut-il faire payer ?

* Le fournisseur peut-il réutiliser des parties du logiciel pour d'autres contrats ?

* Le fournisseur peut-il déposer des droits de licence ou de brevet sur tout ou partie du système ? Si oui, le client devra-t-il s'en acquitter ?

* Le fournisseur a-t-il des droits sur des versions ultérieures développées par le client ?

7.9. checklist : jalons du projet

7.9.1. négociation du contrat

* revue des questions de propriété intellectuelle

* signature du contrat (jalon)

* dates auxquelles le client doit contractuellement fournir des pièces à l'entreprise (jalons)

7.9.2. besoins

* revue de la définition des besoins

* signature du document de définition des besoins (jalon)

* prototypage rapide

7.9.3. dimensionnement

* estimation indépendante de la taille et du calendrier du projet par le prestataire

* accord entre estimations contradictoires

7.9.4. management

* revues des risques

* revues de projet

* inspections

* revues de document

* approbation de documents (jalons)

7.9.5. réception

* plan détaillé de la recette

* conduite des tests de bon fonctionnement

* analyse des test

* acceptation du système (jalon)

7.9.6. formation

* préparation et planification de la formation

* cours

7.9.7. support

* mise en place

* transition du développement à la mise en exploitation et maintenance

7.10. Réception

7.10.1. que mettre dans le cahier de recette

* rôles. Qui sera responsable de

- faire tester ?

- enregistrer les données ?

- analyser les résultats et faire le rapport ?

* où la recette aura-t-elle lieu :

- chez l'entreprise ?

- chez le client ?

- sur le terrain ?

* calendrier :

- machines de test

- équipement de terrain

- autres systèmes

* logiciel nécessaire pour le test

* critères d'acceptation

- taux d'échec acceptable (ex. réussir tous les points critiques et 90% des autres)

* que faire lorsqu'un test échoue ou ne se déroule comme on l'attend

- test de non-régression

* liste des tests à exécuter

- identifiant du test

- objectif du test

- données à enregistrer

- critère de succès

* traçabilité

- pour chaque test, indiquer quelles exigences sont testées

- pour chaque exigence, indiquer les tests associés

7.10.2. que mettre dans la procédure de recette

pour chaque test

- activité de préparation ou de configuration (mettre en route tel ou tel équipement, charger tel ou tel logiciel)

- description pas à pas de la procédure pour le test

- procédure de traitement des données (calculs)

- machines nécessaires (pour exécuter le test et analyser les résultats)

- équipements de terrain nécessaires

- autres systèmes nécessaires

7.10.3. que mettre dans les cas à tester

* données d'entrée

* source d'entrée (manuelle, équipement, simulation, etc.)

* durée du test (ex. recueil de données des boucles pendant une heure)

* valeur(s) attendue(s) en sortie

* critère de succès

* traçabilité entre les cas à tester et les tests

7.10.4. que mettre dans le procès-verbal du test de réception

* nom du test

* date et heure de début

* date et heure de fin le cas échéant

* nom du testeur

* remarques (écarts par rapport à la procédure prévue)

* sorties constatées

7.10.5. que mettre dans le rapport de test

* informations généralités (ex. où le test a-t-il eu lieu)

* résultat général : succès du test, suites à donner

résultats, test par test :

* identifiant du test, procédure de test utilisée, cas de test effectivement testés

* écarts par rapport aux procédures

* données enregistrées dans le log

* données calculées

* succès / échec (selon le critère d'acceptation)

7.11. rôles de support système

* rotations pour la supervision du système

* administration

* rapports de gestion du système

* revue des données produites

* installation de nouveaux matériels

* ajout d'équipements de terrain supplémentaires

* corrections de bugs

* mises à jour :

    - mises à jour de progiciels du commerce

    - nouvelles fonctions

    - modifications (par exemple d'algorithmes)

* diagnostic de problèmes et réparation de matériel

* formation pour toutes les activités ci-dessus (à la mise en service puis lors de mouvements de personnel)

7.12. gestion de configuration

* existe-t-il un plan de gestion de configuration pour votre projet ?

* les produits dont on veut gérer la configuration sont-ils définis ?

* le processus de gestion de configuration fait-il partie intégrante du projet ?

* toutes les versions sont-elles contrôlées ?

* Y a-t-il un référentiel archivant plusieurs versions de base ?

* Des outils de contrôle de configuration sont-ils mis en œuvre ?

* Les demandes de modification et les problèmes remontés sont-ils enregistrés, approuvés, et suivis selon une procédure documentée ?

* Toutes les modifications aux versions de base du référentiel ont-elles lieu selon cette procédure ?

* Les versions de base sont-elles passées en revue périodiquement ?

* Toutes les informations partagées par deux organisations ou plus sont-elles sous gestion de configuration ?

8. Annexe 3. Participants au groupe de travail « Informatique et SAGT » (liste de diffusion au 1/7/99)

Prénom	Nom	Organisation
Joël	BOMIER	ASF
Jean-Luc	BOSSEBOEUF	ASF
Pierre	Boulanger	ASF
Paul	MAAREK	ASF
Michel	BURDEAU	ASFA
Jean-François	BEDEAUX	CERTU
Sylvie	CHAMBON	CERTU
Robert	CHARVIN	CERTU
Jacques	NOUVIER	CERTU
Roger	BOUR	CETE de l'Est
Serge	CARECOLIN	CETE de l'Est
Daniel	CHOLLEY	CETE de l'Est
Eric	MOULINE	CETE de l'Est
Jean-Yves	GIRARD	CETE de l'Ouest
Christophe	Damas	CETE de Lyon
Christophe	Dubois	CETE de Lyon
Ludovic	ALIBERT	CETE Sud-Ouest
Jean-Christophe	Dargenton	CETE Sud-Ouest
Patrick	Olivero	CETE Sud-Ouest
Pierre-Yves	APPERT	CETE Méditerranée
Gildas	Lemaître	CETE Méditerranée
Jean-Marc	CHAUVIN	CETE Normandie-Centre
Jean-Marc	HOTTEAU	CETE Nord-Picardie
Christian	MACHU	CNIR
Daniel	KIMMEL	Conseil Général Hauts-de-Seine
François	HEITZ	C.U. Strasbourg
Frédéric	RICARD	DAFAG
André	BOYER	DDE 06
Serge	FORESTI	DDE 67
Claudia	LEROUX	DDE 94
Michel	FRANCES	DDE 13
Stéphane	NICOUD	DDE 26
Bernard	DURAND	DDE 31
Christian	Françoise	DDE 31
Pierre	TARQUIS	DDE 31
Gilles	DUCHAMP	DDE 33
Pierre	SAMSONOFF	DDE 35
Hervé	LENOIRE	DDE 38
José	CAIRE	DDE 44
P.	Fromentin	DDE 44
M.	Navez	DDE 44
Pascal	PLATTNER	DDE 44
André	MAGNIER	DDE 54
Paul	QUILLOU	DDE 57
Jérôme	WEYD	DDE 59
Cédric	ABERT	DDE 67
Victor	DA FONSECA	DDE 69
Jérôme	MAYET	DDE 69
Philippe	GOUVARY	DDE 974
Denis	Cayeux	DPS/IS3
Michel	MUFFAT	DRAST
Etienne	DE SUSANNE	DREIF / SIER
Jacques	MALBOSC	DREIF / SIER
Jean-Charles	SANTUCCI	DREIF / SIER
Brigitte	THORIN	DREIF/SIER
Marie-Christine	Prémartin	DSCR
André	Maisonneuve	DSCR
Marie-Claire	FABARDINE	Le Grand Lyon
Michel	RIZZI	RATP
Joël	PETRUCCI	SAPRR
Frédéric	BARDOU	SETRA
Christian	PEYRONNE	SETRA
Denis	Servières	SETRA
Jean-Noël	THEILLOUT	SETRA
M.	Sztanke	UTP
Christophe	DARIES	Ville de Marseille
Gérard	DELTHIL	Ville de Paris, Section Exploitation et Circulation
Serge	Mathieu	Ville de Toulouse

9. Glossaire

AGL	Atelier de Génie Logiciel
APS	Avant-Projet Sommaire
APD	Avant-Projet Détaillé
BD	Base de Données
CCTP	Cahier des Clauses Techniques Particulières
CdC	Cahier des Charges
CDES	Cellule Départementale d'Exploitation et de Sécurité
CIGT	Centre d'Ingénierie et de Gestion du Trafic
CNIR	Centre National d'Information Routière
CRICR	Centres Régionaux d'Information et de Coordination Routières
DCE	Dossier de Consultation des Entreprises
DPS	Direction du Personnel et des Services
DOE	Dossier des Ouvrages Exécutés
DR	Direction des Routes
DSE	Dossier de Spécifications Externes
DSCR	Direction de la Sécurité et de la Circulation Routières
GMAO	Gestion de la Maintenance Assistée par Ordinateur
HTML	HyperText Markup Language (syntaxe des pages Web)
IHM	Interface Homme-Machine
LCR	Langage de Commande Routier
MOE	Maîtrise d'OEuvre
MOEF	Maîtrise d'OEuvre Fonctionnelle
MOET	Maîtrise d'OEuvre Technique
MOUV	Maîtrise d'Ouvrage
PGT	Plan de Gestion du Trafic
PMV	Panneau à Message Variable
REA	Réalisateur
SAGT	Système d'Aide à la Gestion du Trafic
SDER	Schéma Directeur d'Exploitation de la Route
SDV	Site Directionnel Variable
SIER	Service Interdépartemental d'Exploitation de la Route
SGBD	Système de Gestion de Base de Données
SIREDO	Système Informatisé de REcueil de DOnnées
SSII	Société de Service en Ingénierie Informatique
TMA	Tierce Maintenance Applicative
VABF	Vérification d'Aptitude au Bon Fonctionnement
VRU	Voie Rapide Urbaine
VSR	Vérification de Service Régulier

10. Références

[1] DPS/IS, Ministère de l'Équipement, méthode d'assistance à la maîtrise d'ouvrage informatique CHEOPS.

[2] SAPRR Manuel d'Exigences Qualité Informatique, référence SAPRR/Syn/MExiQ, version 3.00 du 04/06/1998, non diffusable.

[3] Les marchés publics d'informatique, Richard GANEM, Pascal BOURET, Que sais-je?, PUF, Novembre 97.

[4] La qualité des données à l'âge de l'information, Thomas REDMAN, InterEditions, Masson, 1998.

[5] Guide méthodologique, Exploitation des réseaux principaux des voiries d'agglomération, SDER niveau 1, rapport d'étude CERTU, décembre 1996.

[6] Questionnaire sur l'exploitation du réseau informatique d'un Centre de Gestion du Trafic, envoyé lors du stage de M. Christophe HACHEMI (étudiant à l'ENIC) au CETE de Lille en décembre 1996.

[7] Groupe de travail IVF " régulation du trafic ", enquête sur les PC de régulation du trafic urbain, rédigé par JF Bedeaux, CERTU, mai 98.

[8] Modélisation d'un Système d'Aide à la Gestion du Trafic - Cas pratique : ALIENOR (Agglomération Bordelaise), 50 p. et 2 annexes, Rapport de stage ENTPE pour le CERTU et la DDE33, Y Sauvestre, G Duchamp, J.P Mizzi, Juin 98.

[9] Comment établir un appel d'offres en informatique, François Petit, InterEditions, Masson, 1997.

[10] Newsletter of the ITS Cooperative Deployment Network, Accelerating the ITS Learning Curve: State DOTs Increasingly Willing to Share Software and Experiences, disponible sur http ://www.nawgits.com/code_share.html, Juin 98 (en anglais).

[11] Rôle d'un système de gestion de trafic, Gérard Delthil, Gérard Briet, Ville de Paris, Mars 98.

[12] Traduction en français du résumé d'un rapport d'étude américain, " the road to successful ITS software acquisition ", USDoT, FHWA-JPO-98-035, Mitretek, July 1998.

[13] Progiciels pour les systèmes d'aide à la gestion du trafic : démarche d'intégration de progiciels, Rapport d'étude CERTU, SEMA GROUP, Janvier 1999.

[14] Stratégies pour développer juste, Steve Mc Connell, Microsoft Press, 1997.

[15] Démarche de réutilisation pour les systèmes d'aide à la gestion du trafic sur voies rapides urbaines, Rapport d'étude CERTU, EURIWARE, Septembre 1999.

[16] Recommended approach to software development (revision 3), NASA, Software Engineering Laboraties, SEL-81-305, 1992, disponible sur le web.

[17] Sûreté de fonctionnement pour les systèmes d'aide à la gestion du trafic sur voies rapides urbaines, Rapport d'étude CERTU, IXI, 1999.

[18] Informatisation de la décision dans l'exploitation du transport, Gérard Scémama, Étienne Gaudin, Revue RTS n°61, Octobre-Décembre 1998, pp. 53-67.

[19] Modélisation par objets; Philippe DESFRAY; 1997; InterEditions.

[20] Modélisation objet avec UML ; Pierre Alain MULLER ; 1998 ; Eyrolles.

[21] « 2+1 : un premier bilan », journées d'évaluation, 29 et 30 avril 1999, St Denis de la Réunion, DDE974.

[22] Processus du cycle de vie du logiciel, ISO/CEI 12207.

[24] Évaluation des produits logiciels, ISO/CEI 9126.

[25] Software Process Improvement and Capability determination, SPICE/ISO 15504.

[26] Système d'information documentaire en ingénierie informatique, Z67-100-3.

[27] Analyse de la valeur (recommandations pour la mise en œuvre), X 50-153.

[28] Analyse de la valeur, Analyse fonctionnelle, NF X 50-151.

[29] Analyse de la valeur, Caractéristiques fondamentales, NF X 50-152.

[30] Recommandation de plan qualité logiciel, Z67-130.

[31] Vérification de cohérence du paramétrage d'un système de gestion du trafic, Gérard Delthil, Ville de Paris, mars 1999.