

Méthodes de développement

Guide de rédaction d'un document de définition de produit

1 OBJET DU GUIDE.....	2
2 OBJECTIF DU DOCUMENT DE DEFINITION DE PRODUIT.....	2
3 PLAN DE DDP.....	2
4. TRAVAUX DE PREPARATION DU DDP.....	3
4.1 Projet de solution technique.....	3
4.2 Choix de méthode de développement.....	3
4.3 Identification des tâches.....	3
4.3.1 Les tâches et leur utilisation.....	3
4.3.2 Hiérarchisation des tâches.....	4
4.3.3 Décomposition progressive en tâches.....	4
4.3.4 Découpage d'un projet logiciel en tâches.....	4
4.3.5 Découpage dans les méthodes agiles.....	5
4.3.6 L'organigramme des tâches et l'index WBS.....	5
4.4 Planification - calendrier.....	6
4.5 Estimation des coûts.....	6
5 REDACTION DU DDP.....	7
5.1 Introduction.....	7
5.1.1Objet.....	7
5.1.2 Terminologie et sigles utilisés.....	7
5.1.3 Documents de référence.....	7
5.2 Objectif du projet.....	7
5.3 Présentation de la solution.....	7
5.4 Méthode de développement.....	7
5.5 Organisation proposée.....	7
5.6 Moyens nécessaires.....	8
5.7 Identification des tâches.....	8
5.8 Calendrier proposé.....	8
5.9 Estimation des coûts.....	8
5.10 Annexes architecture (diagrammes).....	8
6 EXEMPLE DE DDP.....	8

1 OBJET DU GUIDE

Ce guide est destiné à aider dans la préparation d'un document de définition de Produit (DDP) rédigé lors de la préparation d'un projet qui sera réalisé avec une méthode agile. Il est illustré par un DDP rédigé en partie.

2 OBJECTIF DU DOCUMENT DE DEFINITION DE PRODUIT

L'objectif du document de définition de produit est d'archiver les résultats des travaux de définition de produit faits durant la préparation de projet.

Il peut être utile :

- à l'équipe qui réalisera le projet lorsque la décision de réalisation sera prise (il peut se passer un long moment entre la préparation d'un projet et le début de réalisation),
- pour les décideurs qui auront à accorder les financements nécessaires et approuver le démarrage du projet,
- pour préparer d'autres projets.

Le document de définition de produit doit donc :

- présenter le projet de solution technique,
- présenter les calendriers prévisionnels et prix de revient estimés.

3 PLAN DE DDP

Le plan suivant est proposé, il pourra être adapté aux caractéristiques du système :

1	Introduction 1.1 Objet 1.2 Terminologie et sigles utilisés 1.3 Documents de référence
2	Objectifs du projet
3	Projet de solution technique et d'architecture 3.1 Caractéristiques principales 3.2 Composition du système 3.3 Principes de fonctionnement 3.4 Services offerts 3.5 Présentation générale de l'architecture 3.6 Base de données 3.7 Interface opérateur 3.8 Techniques de réalisation
4	Méthode de développement
5	Organisation proposée
6	Moyens nécessaires
7	Identification des tâches
8	Calendrier proposé
9	Estimation des prix de revient
Annexes	Architecture du produit -> diagrammes souhaités

4. TRAVAUX DE PREPARATION DU DDP

4.1 Projet de solution technique

Pour pouvoir évaluer les travaux à réaliser, les planifier et évaluer leur coût, il faut commencer par faire un projet de solution technique.

On cherchera d'abord à identifier les fonctions à réaliser. Si le cahier des charges est précis et assez détaillé, il suffira de lister les fonctions demandées par le cahier des charges, sinon on établira une proposition de liste de fonctions à partir du cahier des charges et autres informations recueillies.

On fera un projet d'architecture matériel, et à partir des fonctions à assurer et des interfaces, un projet de découpage du logiciel et de répartition sur les matériels. On précisera les fonctions et caractéristiques principales de chacun des morceaux de logiciel identifié.

On pourra par exemple aborder les aspects suivants pour présenter la solution :

- caractéristiques principales,
- composition du système,
- principes de fonctionnement,
- services offerts,
- présentation générale de l'architecture,
- base de données
- interface opérateur
- techniques de réalisation.

Il ne s'agit pas de faire tout le travail de définition de la solution qui sera fait pendant la réalisation du projet, mais de montrer un principe de solution possible pour réaliser le projet, en vue de pouvoir faire des évaluations.

4.2 Choix de méthode de développement

La méthode de développement a des incidences sur les délais, les coûts, les fournitures, le découpage en tâches... Elle doit donc être choisie dans son principe.

En fonction des contraintes du client, en particulier au niveau des délais, et de l'expérience de l'équipe, on choisira un cycle en V simple, une méthode avec prototype, une méthode itérative par prototypes successifs, une méthode incrémentale, une méthode agile...

Dans le cadre d'une méthode agile, on pourra prévoir plusieurs releases (ou versions) successives du produit. Il faudra préciser les objectifs de chaque release en terme de contenu et de délai, et les fournitures pour chaque release.

4.3 Identification des tâches

4.3.1 Les tâches et leur utilisation

Les tâches sont la base de la gestion de projet. Elles sont utilisées pour la planification et le suivi des délais, et pour l'estimation et le suivi des coûts. Tous les travaux prévus et réalisés dans le projet doivent faire partie d'une tâche du projet. Toutes les fournitures prévues au projet doivent être des produits en sortie d'une tâche du projet.

Une tâche est un ensemble de travaux auxquels on peut attribuer :

- un responsable unique,

- un objet clair, un début et une fin,
- des produits en entrée et des produits en sortie.

4.3.2 Hiérarchisation des tâches

On peut dans un projet vouloir avoir un découpage plus ou moins fin en tâches afin d'assurer un suivi plus fin de certaines parties du projet. Pour cela on peut décomposer certaines tâches en tâches plus petites, et redécomposer certaines tâches...

On obtient ainsi une structure arborescente de tâches. Un certain nombre de règles doivent alors être respectées :

- le niveau de chaque tâche doit être clairement indiqué (niveau 1 pour le premier niveau de décomposition, puis niveau 2, niveau 3...,
- lorsqu'une tâche est décomposée, l'ensemble des activités composant cette tâche doit se retrouver dans une des tâches de décomposition, et la somme des activités des tâches de décomposition doit être identique aux activités de la tâche décomposée,
- on peut attribuer à chaque tâche résultant de la décomposition un responsable unique, un objet clair, un début et une fin, des produits en entrée et des produits en sortie.

4.3.3 Décomposition progressive en tâches

Il n'est souvent pas simple au début d'un projet d'identifier toutes les tâches de bas niveau du projet.

On pourra alors au début du projet se limiter à identifier les tâches des niveaux supérieurs de la décomposition, et ne faire un redécoupage que des premières tâches à réaliser.

On pourra utiliser ces tâches de niveau supérieur pour répartir les responsabilités, faire des plannings généraux, évaluer les coûts et commencer le suivi des coûts.

Au fur et à mesure que le projet avancera, on pourra préciser le contenu des tâches, les décomposer et faire des plannings détaillés.

4.3.4 Découpage d'un projet logiciel en tâches

Le découpage en tâches d'un projet logiciel est fait à partir des activités de développement du logiciel ou du système, des activités projet, de l'organisation de l'équipe et ses sous-groupes, et des itérations ou étapes de la méthode.

Le nombre de tâches à définir varie avec la taille du projet. On considère que pour un petit projet, en dehors des méthodes de type incrémental, le découpage en une dizaine de tâches de premier niveau permet un bon suivi du projet.

Dans une méthode de développement séquentielle, on peut définir une tâche de premier niveau par activité du cycle en V (analyse des exigences, conception de l'architecture, conception détaillée, codage et essais, intégration, validation). On complétera cette première liste de tâches par les activités complémentaires au niveau du système ou du projet tels que la mise en place des plates-formes, l'évaluation des outils, la gestion de projet...

On pourra ensuite décomposer certaines tâches en tâches de niveau 2, par exemple décomposer les conceptions détaillées, codage et essais par sous-groupe de réalisation.

Dans une méthode incrémentale ou itérative (agile ou non agile), le premier niveau de tâches correspondra aux étapes de la méthode, et on pourra découper chaque étape en quelques tâches de niveau 2 en examinant l'objectif et les travaux à réaliser (par exemple sur la base des activités du cycle en V et activités complémentaires, ou selon les produits à réaliser).

4.3.5 Découpage dans les méthodes agiles

Dans une méthode agile, on réalise en général des releases successives, dans ce cas le premier niveau de tâches correspondra aux releases, et le deuxième niveau aux étapes ou itérations des releases. Cependant si l'on n'a prévu qu'une seule release dans le projet, on prendra comme premier niveau les étapes ou itérations de cette release.

Au niveau de chaque étape ou itération, on pourra décomposer la tâche en tâches de niveau inférieur correspondants aux principales activités : gestion du projet, réalisation de logiciel, rédaction de documents...

Cependant pour garder toute l'agilité, le contenu des tâches de réalisation de logiciel ne sera défini qu'au début de chaque itération, et la décomposition de ces tâches de réalisation en tâches élémentaires ne sera aussi faite qu'au début de l'itération.

4.3.6 L'organigramme des tâches et l'index WBS

Dans un grand programme, il peut y avoir beaucoup de produits et des centaines de tâches à gérer. Il faut alors structurer les produits et les tâches. On dispose pour cela d'un outil, l'organigramme des tâches (ou WBS Work Breakdown Structure).

L'organigramme des tâches s'appuie sur une décomposition arborescente du système (sous-systèmes, matériels, logiciels, modules de logiciels...) dite arborescence technique.

Pour chaque niveau de l'arborescence, on fait l'inventaire :

- des produits
- des tâches
- et des moyens

nécessaires pour obtenir le produit de niveau supérieur.

Pour identifier chaque produit, tâche ou moyen, on associe à chacun un index. Cet index est un chiffre ou une lettre au premier niveau, et à chaque niveau on ajoute un chiffre ou une lettre à l'index de l'élément du niveau supérieur de rattachement.

Il n'est pas utile de réaliser un organigramme des tâches dans un petit projet, mais il est recommandé d'utiliser les principes de l'index WBS pour identifier les tâches (voir exemple en annexe 1).

Nota

Attention si vous faites un planning avec un outil de type MS Project. En effet dans un planning (voir l'exemple), on représente en général les tâches du projet et des "jalons" (événements significatifs du projet), et MS Project représente les jalons comme des tâches de durée nulle faisant partie du projet. N'utilisez pas les codes WBS générés automatiquement qui prennent en compte les jalons, et faites votre codage WBS en ne codant que les tâches.

4.4 Planification - calendrier

Pour chaque tâche identifiée, il faut estimer le délai de réalisation et la charge de travail et définir les conditions pour pouvoir commencer la tâche (fin d'autres tâches, arrivée de matériels...).

A partir de ces estimations et des contraintes imposées par le client (délais imposés pour certaines fournitures), on pourra construire un premier calendrier prévisionnel du projet en s'aidant d'un outil de planification. Pour les petits projets, on pourra utiliser MS PROJECT ou MAC PROJECT ; pour les gros projets, on trouve sur le marché des outils plus élaborés.

L'outil de planification peut sortir le calendrier sous forme de GANTT (voir exemple en annexe) qui montre comment les tâches se placent en fonction du temps, ou sous forme de PERT (voir exemple en annexe) qui montre l'enchaînement des tâches et les chemins critiques. L'outil peut aussi indiquer l'utilisation des ressources (personnel, plates-formes...).

A partir de ce premier calendrier, on pourra mettre des contraintes sur les ressources, ajuster les contraintes entre tâches, éventuellement modifier le découpage en tâches, et itérer avec l'outil jusqu'à obtenir un calendrier prévisionnel satisfaisant.

Pour les méthodes incrémentales et itératives, il est recommandé de décomposer le calendrier en un calendrier général montrant l'enchaînement des étapes et itérations, et un calendrier par étape ou itération.

4.5 Estimation des coûts

A partir des tâches identifiées, il faut estimer le prix de revient du projet. Pour cela, on estime la charge de travail par catégorie de personnel et les achats pour chaque tâche, et on en calcule le prix de revient en appliquant les taux horaires de la société.

Les taux horaires sont généralement fixés par catégorie de personnel. Pour la réalisation de logiciel, qui fait appel à des ingénieurs, on distingue en général 3 catégories : ingénieur débutant, ingénieur confirmé, chef de projet.

Si on ne dispose pas de taux horaires société, on pourra estimer des taux horaires. On part du salaire annuel brut moyen de la catégorie et on le multiplie par le taux de charges sociales patronales (environ 1,6) et le taux de frais hors production (direction, administration, locaux, téléphone... 1,2 à 1,6 selon l'organisation et la taille de la société), on obtient ainsi le prix de revient annuel. On passe aux taux horaire en divisant par le nombre d'heures facturables (35H x 52 semaines moins jours fériés, congés, formation, autres activités non facturables, trous de planning... soit 1200 à 1600 heures).

(voir l'exemple)

Pour l'estimation des charges de travail, il n'y a pas de recette miracle, il faut se baser sur l'expérience de la société. Si l'on peut trouver un projet antérieur de même nature pour lequel les données ont été archivées, on comparera les tailles estimées, les complexités... pour extrapoler les chiffres du projet antérieur, et on corrigera de l'augmentation estimée de productivité.

A défaut, on essaiera d'estimer la charge totale de travail (nombre de personnes de l'équipe prévue x durée du projet) et on appliquera des ratios pour les diverses activités : 20 à 30 %

pour l'analyse des exigences, 20 à 30 % pour la conception, 15 à 20 % pour codage et essais, 20 à 30 % pour l'intégration, 10 à 20 % pour la qualification (si cette activité est retenue).

On peut alors dresser un tableau des prix de revient des tâches (si on a décomposé des tâches, attention à ne pas les compter deux fois).

(voir l'exemple)

5 REDACTION DU DDP

5.1 Introduction

5.1.1 Objet

Préciser le système objet du DDP et brièvement sa fonction.

5.1.2 Terminologie et sigles utilisés

Définir les abréviations employées et les termes qui ont un sens particulier.

5.1.3 Documents de référence

Indiquer les documents auxquels le DDP se réfère.

5.2 Objectif du projet

Ce paragraphe présente en une dizaine de lignes les objectifs du projet :

- produits à réaliser, leurs missions, caractéristiques principales et points durs,
- délais prévus et contraintes.

5.3 Présentation de la solution

Ce paragraphe présente les résultats des réflexions sur la solution technique.

On pourra décrire par exemple :

- les caractéristiques principales du système vis à vis des utilisateurs,
- composition du système en matériels, postes utilisateur, réseaux...,
- principes de fonctionnement des différentes fonctionnalités du système,
- services offerts par le système,
- présentation générale de l'architecture,
- base de données : principales tables, SGBD...
- interface opérateur : quelques maquettes d'écrans,
- techniques de réalisation : langage, méthode de conception, outils...

Il ne s'agit pas de recopier le cahier des charges, mais de montrer que vous avez réfléchi au problème, identifié les points durs, et que vous pouvez maîtriser la réalisation.

5.4 Méthode de développement

Indiquer les grandes étapes de la méthode de développement (itérations dans un processus itératif, releases dans un processus agile), et le contenu et les fournitures prévus pour chaque grande étape.

Si un plan de développement est rédigé, on ne recopiera pas le plan de développement, mais on se limitera à attirer l'attention sur les points les plus importants.

5.5 Organisation proposée

La présentation de l'organisation indiquera :

- la place de l'équipe projet dans la société,
- la composition de l'équipe projet,
- les principales responsabilités déléguées.

L'organisation sera illustrée par un organigramme.

(voir exemple)

Si un plan de développement est rédigé, on ne recopiera pas le plan de développement, mais on se limitera à attirer l'attention sur les points les plus importants.

5.6 Moyens nécessaires

Faire une synthèse des infrastructures et moyens nécessaires pour réaliser le projet : matériels, outils logiciels, réseaux, bâtiments...

Préciser la disponibilité de ces infrastructures et moyens, et, pour les infrastructures ou autres moyens à acheter ou développer, les travaux à réaliser et leur calendrier.

Si un plan de développement est rédigé, on ne recopiera pas le plan de développement, mais on se limitera à attirer l'attention sur les points les plus importants.

5.7 Identification des tâches

Faire un tableau des tâches qui ont été identifiées.

5.8 Calendrier proposé

Faire un calendrier général sous forme d'un diagramme de GANTT montrant l'enchaînement des étapes, les objectifs calendaires du projet et de la réalisation des tâches de niveau supérieur.

5.9 Estimation des coûts

On présentera ici :

- un tableau indiquant le coût horaire par catégorie d'intervenant (chef de projet, ingénieur confirmé, analyste-programmeur),
- un tableau indiquant pour chaque tâche identifiée :
 - le nombre d'heures prévues par catégorie,
 - le nombre total d'heures,
 - le prix de revient.

5.10 Annexes architecture (diagrammes)

Tous les termes utilisés doivent être définis dans le paragraphe "Présentation de la solution".

6 EXEMPLE DE DDP

La suite de ce guide est un exemple de DDP. Il n'est pas fourni un DDP complet mais seulement certains chapitres ou parties de chapitres.

Cet exemple répond au cahier des charges pour des navettes d'aéroport que vous trouverez dans l'introduction aux guides de rédaction. Le système à réaliser est un démonstrateur

simulant des bornes interactives et le centre de contrôle. Un objectif de fournir un produit au bout d'un an a été fixé.

Avertissement

Pour vous éviter l'envie de recopier trop bêtement cet exemple, les étapes de la méthode sont différentes de celles de votre projet. Vous aurez donc à faire les adaptations nécessaires.

1 - Introduction.....	11
1.1 Objet.....	11
1.2 Terminologie et sigles utilisés.....	11
1.3 Documents de référence.....	11
2 - Objectifs du projet.....	11
3 - Projet de solution technique et d'architecture	12
3.1 Caractéristiques principales.....	12
3.2 Composition du système	12
3.3 Principes de fonctionnement	12
3.4 Services offerts	14
3.5 - Présentation générale de l'architecture	15
3.6 La base de données.....	16
3.7 Interface opérateur	17
3.8 Techniques de réalisation	17
4 - Méthode de développement.....	18
4.1 Méthode	18
4.2 Release 1.....	18
4.3 Release 2.....	18
4.4 Fournitures	19
5 - Organisation proposée	19
6 - Moyens nécessaires	20
7 - Identification des taches	20
8 - Calendrier proposé.....	20
9 - Estimation des prix de revient	21
9.1 Taux horaires	21
9.2 Prix de revient (euros)	21
Annexe 1 - Architecture du produit.....	22
Annexe 2 - Planning général	24

1 - Introduction

1.1 Objet

Ce document présente la solution proposée pour le démonstrateur de navettes d'aéroport

Il répond au cahier des charges cité en document de référence.

1.2 Terminologie et sigles utilisés

Navette d'aéroport Autobus faisant la navette entre un aéroport et les villes voisines

Usager Client de l'aéroport qui demande des services au système

CP Chef de projet

IC Ingénieur confirmé

ID ingénieur débutant

ODBC Open DataBase Connectivity

QT Bibliothèque de classes C++ fournie par Nokia

SGBD Système de gestion de base de données

SGB Spécification générale de besoin

SVN Sub Version Number

UML Uniform Method Language

WBS Work Breakdown Structure (organigramme des tâches)

1.3 Documents de référence

Cahier des charges pour des navettes d'aéroport N°.... édition .. du

2 - Objectifs du projet

L'objectif du projet est de fournir un logiciel démonstrateur en vue d'organiser des démonstrations du système de navettes d'aéroport à des clients potentiels.

Il devra donc prendre en compte les principales fonctionnalités du système : simulation de borne interactive, centre de contrôle. Il doit pouvoir s'exécuter avec simplement deux ou trois PC en réseau.

L'objectif de délai est d'être prêt pour des démonstrations au bout d'un an.

3 - Projet de solution technique et d'architecture

3.1 Caractéristiques principales

Le système Navettes d'aéroport est destiné aux usagers de l'aéroport qui désirent utiliser les autobus navettes de l'aéroport pour se rendre à l'aéroport ou quitter l'aéroport en vue :

- de les informer sur les lignes existantes, les arrêts, les horaires et les tarifs,
- de les informer sur les heures de départ ou passage prévues des prochaines navettes,
- de leur permettre d'acheter les billets.

Le système comprend un centre de contrôle et des bornes interactives à la disposition des voyageurs aux points d'arrêt des navettes. Le démonstrateur comprendra le calculateur du centre de contrôle et la simulation d'une ou plusieurs bornes sur des PC.

3.2 Composition du système

Le démonstrateur est composé d'un calculateur de centre de contrôle et de un ou plusieurs PC simulant chacun une borne interactive.

Les bornes interactives sont prévues avec un écran tactile pour les dialogues avec les utilisateurs. Cet écran tactile est simulé par un écran de PC, et la pression sur l'écran tactile simulée par un clic de souris sur la zone. Le clavier du PC ne sera plus utilisé une fois que le système aura été initialisé.

Le calculateur du poste de contrôle est un PC standard.

Les calculateurs du démonstrateur sont reliés par un réseau Ethernet.

Un schéma du système est montré en annexe.

3.3 Principes de fonctionnement

3.3.1 Le réseau de navettes

Le système est destiné aux aéroports où sont organisés des navettes d'autobus entre l'aéroport, le centre ville, et les villes importantes voisines, pour amener les passagers à l'aéroport ou les ramener.

Le réseau de lignes d'autobus est en étoile entre l'aéroport et des terminus situés en ville, avec des arrêts intermédiaires. Cependant les seuls trajets prévus sont entre l'aéroport et un arrêt (intermédiaire ou terminus).

Une *ligne* relie l'aéroport et un terminus dans les deux sens, et se caractérise par :

- son terminus,
- des arrêts et des points de passages,
- une liste d'horaires de passages journaliers dans les deux sens.

Les horaires sont les mêmes tous les jours. Cependant ils pourront être modifiés pour la journée en cours : annulation de circulation d'autobus, retards.

Pour l'information des clients, des bornes interactives sont prévues au moins dans l'aéroport et à chaque arrêt. Ces bornes, en fonctionnement permanent affichent les heures prévues des

prochains départs, et permettent au client de demander des informations et d'acheter des billets.

Un centre de contrôle permet de surveiller le fonctionnement du système, faire des synthèses, mettre à jour les informations, et diffuser aux bornes les informations sur les retards ou annulations.

3.3.2 Vente de billets et tarifs

Les clients doivent acheter des billets pour les navettes. Ils pourront le faire à toutes les bornes interactives.

Les *tarifs* des billets comprennent un prix unique prédéfini pour chaque zone tarifaire. Pour chaque arrêt est défini une zone tarifaire. Le prix est le même pour les trajets aéroport - arrêt et arrêt - aéroport.

Pour les achats de 10 billets ou plus une ristourne est faite, de 5 % pour 10 billets et proportionnelle au logarithme du nombre au delà.

Chaque borne interactive entretient un *cumul des ventes*, il comprend le nombre de billets vendus et le montant payé par type de carte de paiement pour chaque zone tarifaire de chaque ligne (ligne pour laquelle les billets sont demandés).

Le centre de contrôle fait une *synthèse des ventes* qui comprend :

- le montant des ventes, en nombre de billets et en euros, pour chaque borne interactive,
- le montant des ventes, en nombre de billets ou en euros, pour chaque ligne (ligne pour lesquels les billets sont demandés), et chaque zone tarifaire de la ligne,
- les nombres de billets et montants totaux (par type de carte de paiement et global).

3.3.3 La borne interactive opérationnelle

Une ou plusieurs bornes interactives seront placées aux différents endroits fréquentés par les utilisateurs : aéroport, terminus des lignes, arrêts des lignes, offices de tourisme...

Une borne, en fonctionnement permanent, comprend un écran tactile et un terminal de paiement pour carte de crédit. Elle affiche les heures prévues des prochains départs, et une zone de dialogue permet au client de demander des informations sur les trajets, les horaires et les prix, et d'acheter des billets payés par carte bancaire ou carte privative prépayée.

La borne s'initialise automatiquement et est reliée par un réseau au centre de contrôle qui lui envoie les modifications de prévisions de circulation des autobus et les modifications de lignes ou horaires. La borne envoie au centre de contrôle le cumul de ses ventes de billet.

Les informations affichées en permanence sont :

- la date et l'heure,
- les destinations et heures prévues des prochains départs (départs de l'aéroport pour les bornes d'aéroport, départ vers l'aéroport pour les bornes situées dans des terminus de ligne ou des arrêts) et le temps restant jusqu'au prochain départ,
- le dernier message d'information envoyé par le centre de contrôle (si le centre de contrôle a une information générale à donner).

La borne affiche aussi des boutons permettant à l'utilisateur d'obtenir des informations sur le réseau de navettes ou d'acheter des billets.

Lorsque l'utilisateur demande des informations sur le réseau de navettes, une partie de l'écran est réservée au dialogue avec cet utilisateur en vue de lui permettre de préciser sa demande et afficher les informations désirées.

Lorsque l'utilisateur veut acheter des billets, la même zone de dialogue sera utilisée pour préciser le nombre et le type de billet, indiquer le prix et piloter le paiement. Afin d'éviter que les bornes soient fracturées, le paiement se fera uniquement par carte de crédit ou carte privée prépayée du système de navette. La borne dispose pour cela d'un lecteur de cartes de crédit et d'un terminal cartes de crédit.

3.3.4 La simulation de la borne interactive

Le simulateur simule la borne sur un PC. L'écran du PC simule l'écran tactile, et la désignation sur l'écran tactile est simulée avec la souris. Les autres moyens de dialogue du PC ne seront plus utilisés une fois que le programme aura été lancé et se sera initialisé.

Le lecteur et le terminal carte de crédit ne sont pas simulés, on se limite à montrer la séquence des dialogues de paiement sur l'écran tactile et à demander au centre de contrôle l'autorisation de paiement dans le cas d'une carte bancaire.

Le réseau d'échanges d'information n'est pas non plus simulé, on se limite à échanger les informations avec le centre de contrôle par une liaison Ethernet.

3.3.5 Le centre de contrôle opérationnel

...

3.3.6 Simulation du centre de contrôle

...

3.3.7 Simulation du réseau

Le système opérationnel devra pouvoir être adapté à divers types de réseau avec des vitesses de transmission variées en fonction de la localisation des bornes.

On ne cherchera pas à simuler ces divers types de réseau dans le démonstrateur. On se limitera à relier les simulateurs de borne au centre de contrôle par un réseau Ethernet.

Cependant le système sera conçu pour limiter au maximum les volumes d'échange sur le réseau et pouvoir être utilisé avec des liaisons basse vitesse. En particulier on implantera une base de données sur chaque borne.

3.4 Services offerts

La liste des services prévus est la suivante :

Affichage des prochains départs

L'écran de chaque borne affiche en permanence des informations à destination de tous les utilisateurs :

- situation de la borne, date et heure, fête du jour
- durée restante jusqu'au prochain départ,
- une liste des prochains départs : terminus et heure,
- le message d'information envoyé par le centre de contrôle.

Informations sur le réseau

...

Vente de billets

L'usager peut acheter des billets sur la borne (de 1 à 99 billets). Un billet est valable pour tous les trajets dans les deux sens entre l'aéroport et un arrêt de la zone tarifaire.

Le paiement pourra se faire par carte de crédit ou carte de paiement privative, mais on se bornera à simuler les dialogues dans les deux cas et à demander au centre de contrôle l'autorisation pour une carte bancaire

Opérations courantes au centre de contrôle

...

Création ou modification de lignes

...

Mise à jour des tarifs

...

Synthèse des ventes

...

Création et modification de mots de passe

...

Sauvegarde et restauration des données

...

Initialisation et arrêt du système

...

Initialisation d'une borne

La simulation d'une borne est lancée en lançant l'exécution du programme correspondant. Un fichier d'initialisation lui fournira ses paramètres d'initialisation, en particulier sa situation. La borne s'initialise, se connecte au centre de contrôle, puis rafraîchit toutes les 10 secondes son affichage permanent et attend les demandes de service de l'utilisateur pour les exécuter.

3.5 - Présentation générale de l'architecture

Le logiciel est découpé en deux programmes :

- un programme pour la borne,
- un programme pour les fonctions centre de contrôle.

Un seul processus s'exécute sur la borne. Il crée deux threads pour les échanges réseau et un thread pour la simulation du paiement.

Un seul processus s'exécute au centre de contrôle. Il assure la fonction de serveur de réseau et crée pour cela deux threads pour chaque borne qui se connecte.

3.6 La base de données

Les données nécessaires pour le fonctionnement du système sont stockées au centre de contrôle dans une base de données. Un sous ensemble de cette base sera implanté dans chaque borne.

Le démonstrateur sera prévu pour pouvoir utiliser au centre de contrôle tout type de SGBD : SQL Server, Oracle, MySql, PostgreSQL, Access, SQLite...

La borne pourrait aussi utiliser tout type de SGBD, mais il sera plus simple de se limiter aux SGBD qui ne nécessitent pas de fonction serveur et où la base est contenue dans un fichier (Access ou SQLite).

La base de données comprend :

- la table des lignes donnant pour chaque ligne le code, le sens et le terminus,
- la table des stations donnant pour chaque station son nom, sa localisation et sa zone tarifaire,
- la table ligne - stations fournissant les associations ligne - station -station suivante,
- la table des navettes fournissant les associations lignes - navettes,
- la table des horaires fournissant les associations navette - station - heure de passage
- la table des tarifs indiquant pour chaque tarif la période de validité et le prix pour chaque zone tarifaire,
- la table des bornes indiquant pour chaque borne le nom, la station d'implantation et son état,
- la table calendrier indiquant le saint du jour pour chaque jour de l'année
- la table des messages textuels, qui enregistre pour chaque message émis, la date et heure d'émission, l'émetteur, les destinataires et le contenu du message,
- la table des cumuls qui enregistre borne par borne pour chaque jour le nombre de billets vendus par zone tarifaire et type de carte de paiement, et les montants correspondants,
- la table des mots de passe qui contient les identificateurs et mots de passe (sous forme codée) des opérateurs et administrateurs du centre de contrôle.

Pour dimensionner ces tables, on retient les valeurs suivantes :

Nombre de lignes possibles	10
Nombre maximum d'arrêts par ligne	10
Nombre maximum de zones tarifaires	10
Nombre maximum de circulations journalières sur une ligne	10 dans chaque sens
Nombre maximum de bornes	50
Nom de station, d'opérateur, d'administrateur	20 caractères maximum
Code circulation	4 chiffres
Taille maxima d'un message textuel	40 caractères
Taille des mots de passe	Entre 8 et 16 caractères

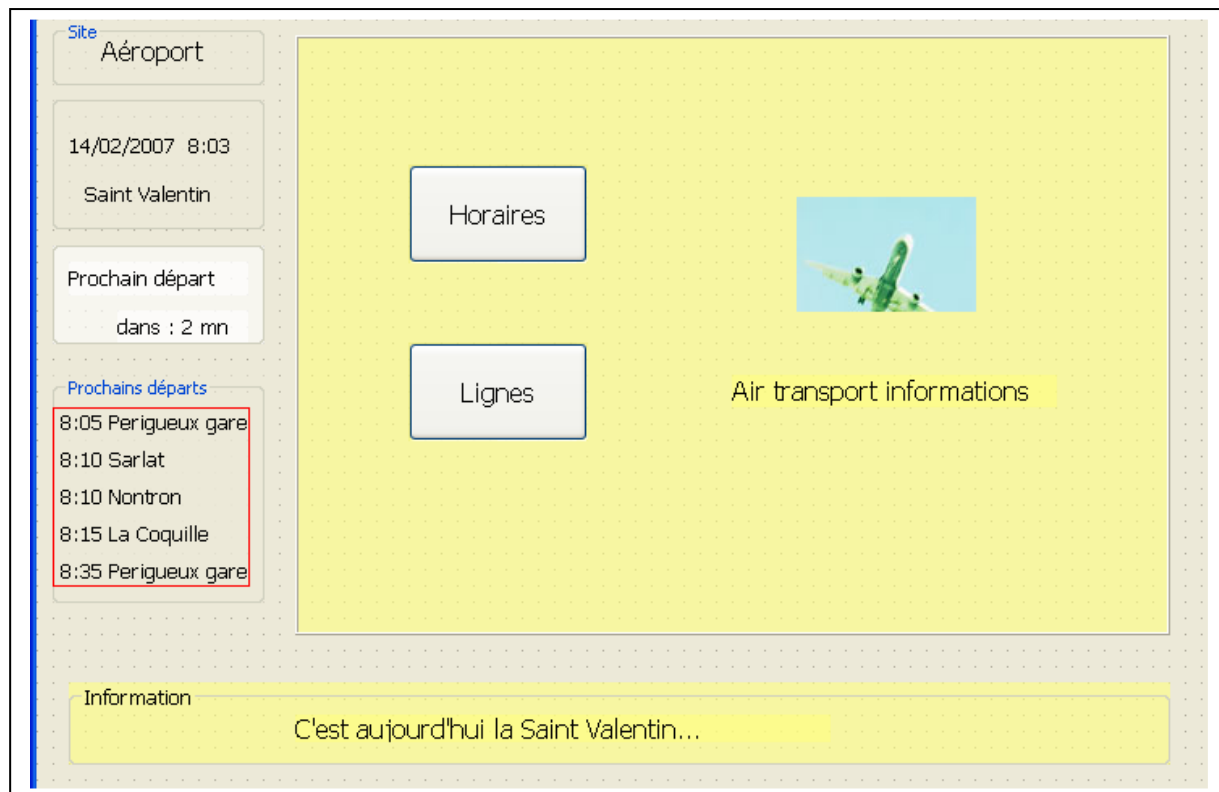
Les utilitaires du SGBD seront utilisés pour la création de la base, les sauvegardes et restauration, ainsi que pour les opérations de type suppression de lignes, de stations, de messages...

3.7 Interface opérateur

3.7.1 Borne

L'interface opérateur de la borne réelle est un écran tactile, et on prendra en compte les contraintes de cet écran pour le démonstrateur. L'écran sera divisé en deux zones comme présenté dans la maquette ci-dessous. La partie gauche contient les informations permanentes : nom de la station, date et heure, temps restant jusqu'au prochain départ, heures des prochains départ, et en bas le dernier message du centre de contrôle.

La partie droite de l'écran est réservée pour les services à la demande. En l'absence de service demandé elle présente deux pavés de sélection. Lorsqu'un service est demandé elle présente des boutons de sélection et des informations. Les boutons de sélection ont une taille suffisante pour permettre une sélection au doigt.



3.7.2 Centre de contrôle

....

3.8 Techniques de réalisation

La conception des programmes sera de type objet, et on utilisera UML au niveau de la conception.

Les programmes seront codés et mis au point sous Windows 7 en C++ avec l'atelier Qt Creator et la bibliothèque QT de Nokia.

Pour bien maîtriser les évolutions de code, on utilisera l'outil de gestion des sources SVN.

La base de données sera accédée par la bibliothèque ODBC et on utilisera une technique de programmation indépendante du SGBD. Cela permettra de changer simplement de base de données si besoin.

Pour faire les essais on utilisera SQL Server pour le centre de contrôle et Access pour les bornes.

4 - Méthode de développement

4.1 Méthode

On utilisera une méthode agile. Deux releases sont prévues, la première au bout de 6 mois, et la seconde au bout de un an.

Pour chaque release, il est prévu une étape de définition de release de 4 semaines suivie de 5 itérations de réalisation de 4 semaines (soit 48 semaines au total - un an avec 4 semaines d'interruption pour vacances).

4.2 Release 1

Cette première release disponible au bout d'environ 6 mois, contiendra les fonctions les plus prioritaires de chacun des deux postes qui seront reliés en réseau.

On prévoit donc les fonctions suivantes :

- sur le simulateur de borne : initialisation et dialogues avec le centre de contrôle, affichage des prochains départ, vente de billets,
- au centre de contrôle : surveillance de l'état des bornes, dialogue avec les bornes, base de données.

Cette première release pourra être livrée aux équipes de démonstration où à quelques clients privilégiés pour des travaux d'évaluation accompagnée de la documentation suivante :

- la spécification générale de besoin du produit,
- le document de définition de release,
- la procédure de recette de la release,
- le manuel d'utilisation.

4.3 Release 2

Cette release doit couvrir l'ensemble des besoins du cahier des charges.

Elle sera disponible au bout d'un an pour réaliser des démonstrations, et pourra être livrée à quelques clients privilégiés pour des travaux d'évaluation.

Elle sera accompagnée de la documentation suivante :

- la spécification générale de besoin du produit,
- le document de définition de release,
- la procédure de recette de la release,
- le manuel d'utilisation,

- des notes de release (problèmes identifiés, restrictions d'emploi, notes d'installation...).

4.4 Fournitures

En plus de la fourniture des deux releases et de leur documentation d'accompagnement, on prévoit les fournitures suivantes :

- plan qualité logiciel,
- dossier de test de la release 1,
- dossier de test de la release 2,
- documentation intégrée au code de la release 2,
- bilan de projet.

5 - Organisation proposée

Le logiciel est développé par une équipe projet placée dans le service Recherche et Développement de la société NAVFORCE.

L'équipe projet est prévue composée :

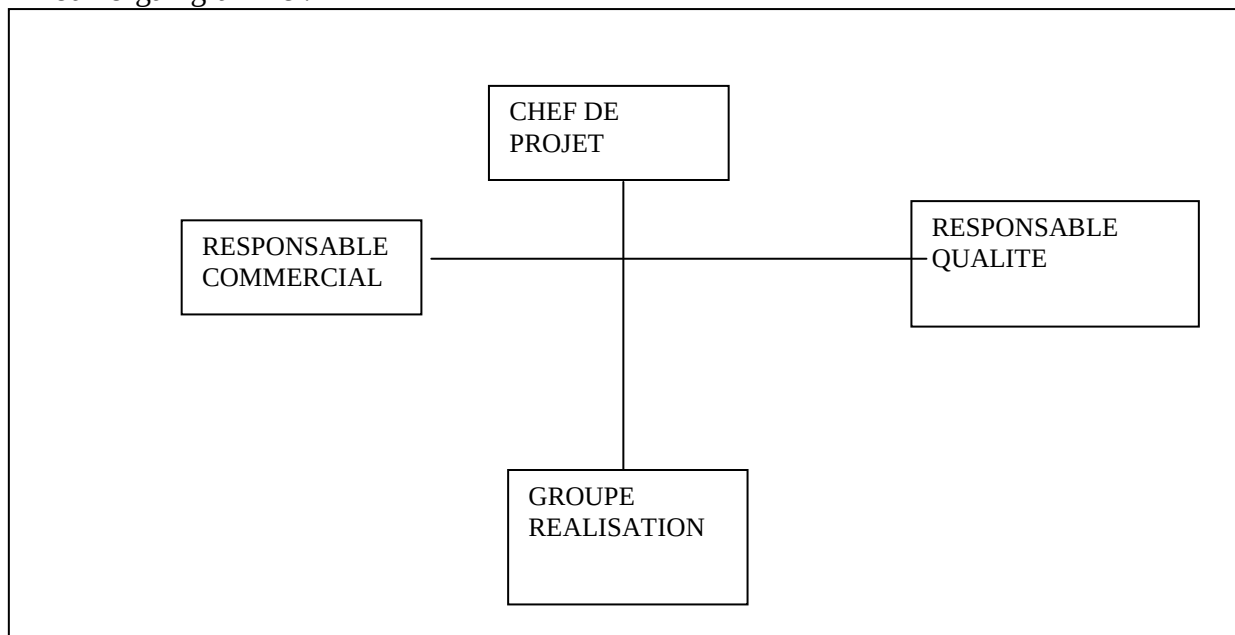
- d'un chef de projet très expérimenté,
- d'un ingénieur confirmé,
- et d'un jeune ingénieur.

Ils effectueront toutes les activités de développement.

Le chef de projet assurera le rôle de responsable commercial en liaison avec le service commercial de la société.

L'ingénieur confirmé assurera le rôle de responsable qualité en liaison avec le service qualité de la société.

D'où l'organigramme :



6 - Moyens nécessaires

Les moyens nécessaires sont :

- 3 PC sous Windows 7 (disponibles),
- les logiciels MS Project, Qt Creator et bibliothèque QT, SVN, SQL Server, Access (disponibles)
- un outil pour la modélisation UML (à choisir).

7 - Identification des tâches

Les tâches suivantes sont identifiées :

WBS	Tache
11	Organisation et gestion de projet release 1
12	Définition release 1
13	Réalisation release 1 - itération 1
14	Réalisation release 1 - itération 2
15	Réalisation release 1 - itération 3
16	Réalisation release 1 - itération 4
17	Réalisation release 1 - itération 5
18	Documents de release
21	Gestion de projet release 2
22	Définition release 2
23	Réalisation release 2 - itération 1
24	Réalisation release 2 - itération 2
25	Réalisation release 2 - itération 3
26	Réalisation release 2 - itération 4
27	Réalisation release 2 - itération 5
28	Documents finaux

8 - Calendrier proposé

L'objectif du projet est de pouvoir fournir un logiciel pour organiser des démonstrations au bout d'un an.

Un objectif intermédiaire est fixé, la fourniture d'une première release au bout de 6 mois.

Le calendrier général prévisionnel est fourni en annexe dans l'hypothèse d'un démarrage du projet le 10 février :

9 - Estimation des prix de revient

9.1 Taux horaires

Les taux horaires estimés (en euros) pour les catégories d'ingénieurs du projet sont :

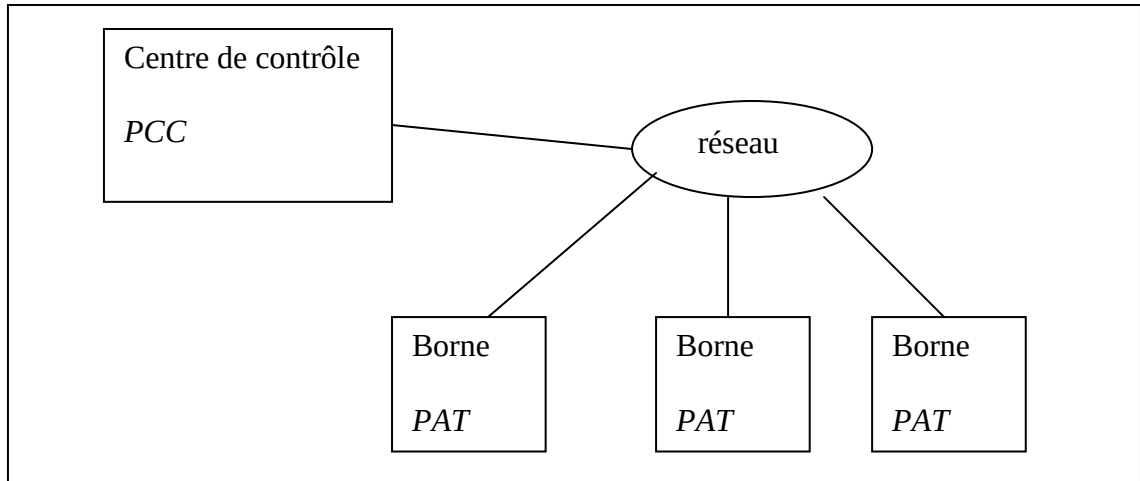
	Salaire brut (k€)	Taux CS	Taux FHP	Total annuel (k€)	Nb heures	Taux horaire HT (€)
CP	50	1,62	1,41	114	1200	95
IC	40	1,62	1,41	91	1400	65
ID	30	1,62	1,41	68	1500	45

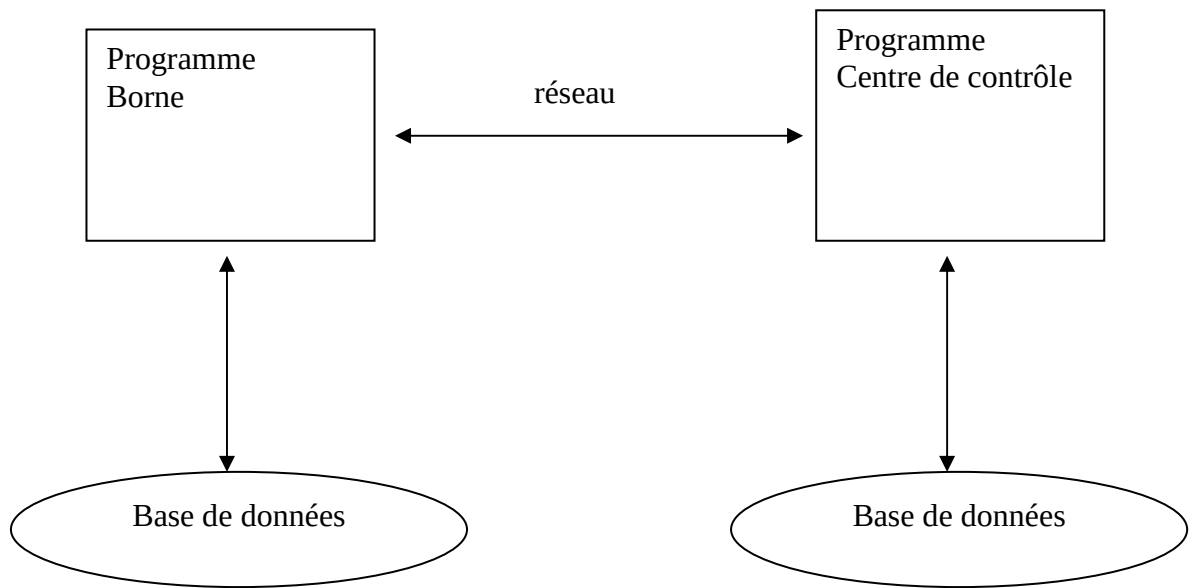
9.2 Prix de revient (euros)

WBS	Tache	Heures CP	Heures IC	Heures ID	Achats HT	Prix revient HT
11	Organisation et gestion de projet release 1	40	50	0	1000	8050
12	Définition release 1	60	100	120		17600
13	Réalisation release 1 - itération 1	90	100	120		20450
14	Réalisation release 1 - itération 2	90	100	120		20450
15	Réalisation release 1 - itération 3	90	100	120		20450
16	Réalisation release 1 - itération 4	90	100	120		20450
17	Réalisation release 1 - itération 5	90	100	120		20450
18	Documents de release	10	50	50	100	6550
21	Gestion de projet release 2	20	50	0		5150
22	Définition release 2	90	100	120		20450
23	Réalisation release 2 - itération 1	90	100	120		20450
24	Réalisation release 2 - itération 2	90	100	120		20450
25	Réalisation release 2 - itération 3	90	100	120		20450
26	Réalisation release 2 - itération 4	90	100	120		20450
27	Réalisation release 2 - itération 5	90	100	120		20450
28	Documents finaux	10	50	50	100	6550
totaux		1130	1400	1540	1200	268850

Annexe 1 - Architecture du produit

Architecture matérielle



Architecture logicielle

Annexe 2 - Planning général

