

Plan de développement

Synthèse vocale de numéros téléphoniques

Equipe:

CUMMINGS Thibaud
GOLETTA Michael
SOLEYMANKHANI Hossein
DIALLO Négué

Table de Révision

Révision	Date	Auteur	Modifications apportées
1	25 Février 2010	T. CUMMINGS	Version initiale
2	27 Février 2010	N. DIALLO	MAJ de 6.1 et ajout du Glossaire, Références, Index

Table des matières

1	INTRODUCTION	3
1.1	OBJECTIFS ET MÉTHODES	3
1.2	DOCUMENTS DE RÉFÉRENCE	3
2	ORGANISATION.....	4
2.1	DÉCOMPOSITION EN TÂCHES	4
2.2	STRUCTURE DES ÉQUIPES	5
3	PLANIFICATION.....	6
4	CYCLE DE VIE	7
5	METHODES ET OUTILS.....	8
6	DOCUMENTATION.....	9
6.1	PRESENTATION.....	9
6.1.1	Manuel d'utilisation.....	9
6.1.2	Manuel d'installation.....	9
6.1.3	Documentation interne du code	9
6.1.4	Plan des tests.....	10
6.2	STANDARD ET OUTILS	10
7	QUALITE	10
8	GLOSSAIRE	11
9	RÉFÉRENCES.....	11
10	INDEX	12

1 INTRODUCTION

Ce document apporte la description précise de l'organisation à respecter tout au long du déroulement du projet. Il décrit l'approche choisie pour développer la librairie.

1.1 Objectifs et méthodes

Il est demandé de livrer une librairie codé en langage C permettant la synthèse vocale de numéro téléphoniques ainsi qu'une application qui teste cette librairie. La librairie qu'il est demandé de fournir doit comporter une fonction permettant la reconstruction vocale d'un numéro téléphonique. Afin de restituer vocalement des numéros de téléphones, il est nécessaire d'implémenter un algorithme de synthèse vocale. A partir d'une série de nombres en entrée, l'algorithme devra être capable de générer le fichier son correspondant à ces nombres par une voix reconstituée.

1.2 Documents de référence

Le deux documents de référence ici sont le cahier des charges (S1) et le cahier de recette (S2)

2 ORGANISATION

2.1 Décomposition en tâches

L'équipe de développement suit une organisation définie par le chef de projet. La liste des tâches à accomplir est la suivante:

- Analyse besoins
 - Définir les objectifs du projet
 - éliciter besoins du client
- Spécifications
 - Rédiger cahier charges
 - Rédiger cahier de recette
- Conception
 - Concevoir l'architecture, diviser en modules
 - Concevoir en détail chaque module
- Développement
 - Développer les modules
 - Développer les tests unitaires
 - Effectuer les test unitaires
- Enregistrements
- Intégration
- Validation fonctionnelle
- Livraison : recettes avec le client
 - Livrer la librairie et l'application de test
 - Effectuer les tests du cahier de recette
 - Livrer documents: manuels d'installation et d'utilisation, plan des tests, documentation interne code
- Coordination de l'équipe
- Interface client : une réunion par semaine

2.2 Structure des équipes

Chacun des membres de l'équipe s'occupe du développement d'un ou deux modules. Pour s'assurer du bon fonctionnement de la librairie, les tests unitaires sont développés et effectués par un membre autre que celui qui a développé le module.

- CUMMINGS Thibaud
 - Chef de projet: coordination équipe
 - Développement du module "modification de vitesse" et "jouer son"
- GOLETTO Michael
 - Chef de développement: coordination de l'équipe lors du développement
 - Intégrateur
 - Développement du module "jouer son"
- SOLEYMANKHANI Hossein
 - Enregistrements sons
 - Développement du module "analyse syntaxique"
- DIALLO Négué
 - Chef de rédaction de la documentation
 - Développement du module "traitement de la chaîne" et "interface utilisateur"

T. Cummings est le chef de projet et étant le plus à l'aise en mathématiques (traitement du signal) il s'occupe du module qui a pour but de modifier la vitesse de lecture. Il s'occupe également du module "jouer son" en collaboration avec M. Goletto.

M. Goletto est le membre le plus expérimenté en langage C, celui-ci a donc été nommé intégrateur et chef de développement et a pour tâche d'aider les autres membres lors de la phase de développement. En parallèle, il s'occupe du développement du module "jouer son" (la librairie utilisée, SDL demande une certaine rigueur de codage du fait que cette librairie utilise beaucoup de pointeurs).

L'enregistrements des sons a été légué à H. Soleymankhani, en effet celui-ci est habitué des enregistrements, il a de l'expérience dans ce domaine et maîtrise les outils nécessaires. Du fait qu'il s'occupe des enregistrements il a pour tâche de développer le module effectuant l'analyse syntaxique du numéro (segmentation du numéro par rapport au enregistrements).

Pour finir, N. Diallo s'occupe de superviser la rédaction des documents à rendre (manuels d'installation etc..) ainsi que du développement de deux modules. L'un qui va traiter la chaîne en entrée, l'autre qui va s'assurer d'effectuer l'interface avec l'utilisateur.

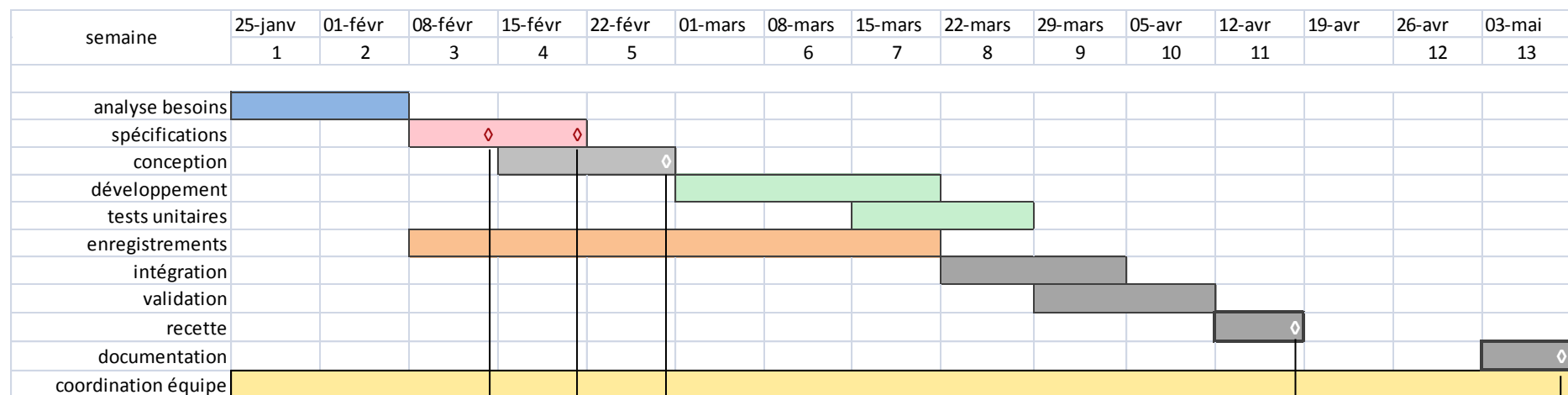
3 PLANIFICATION

Voici un schéma de l'organisation à suivre:

étant donné l'importance portée à la documentation du code source, la phase de développement est longue.

Les phases d'intégration et de développement se chevauchent en cas de problèmes.

Chaque losange correspond à une date butoir pour les documents à rendre.



Plan de développement

Cahier de recette

Rapport

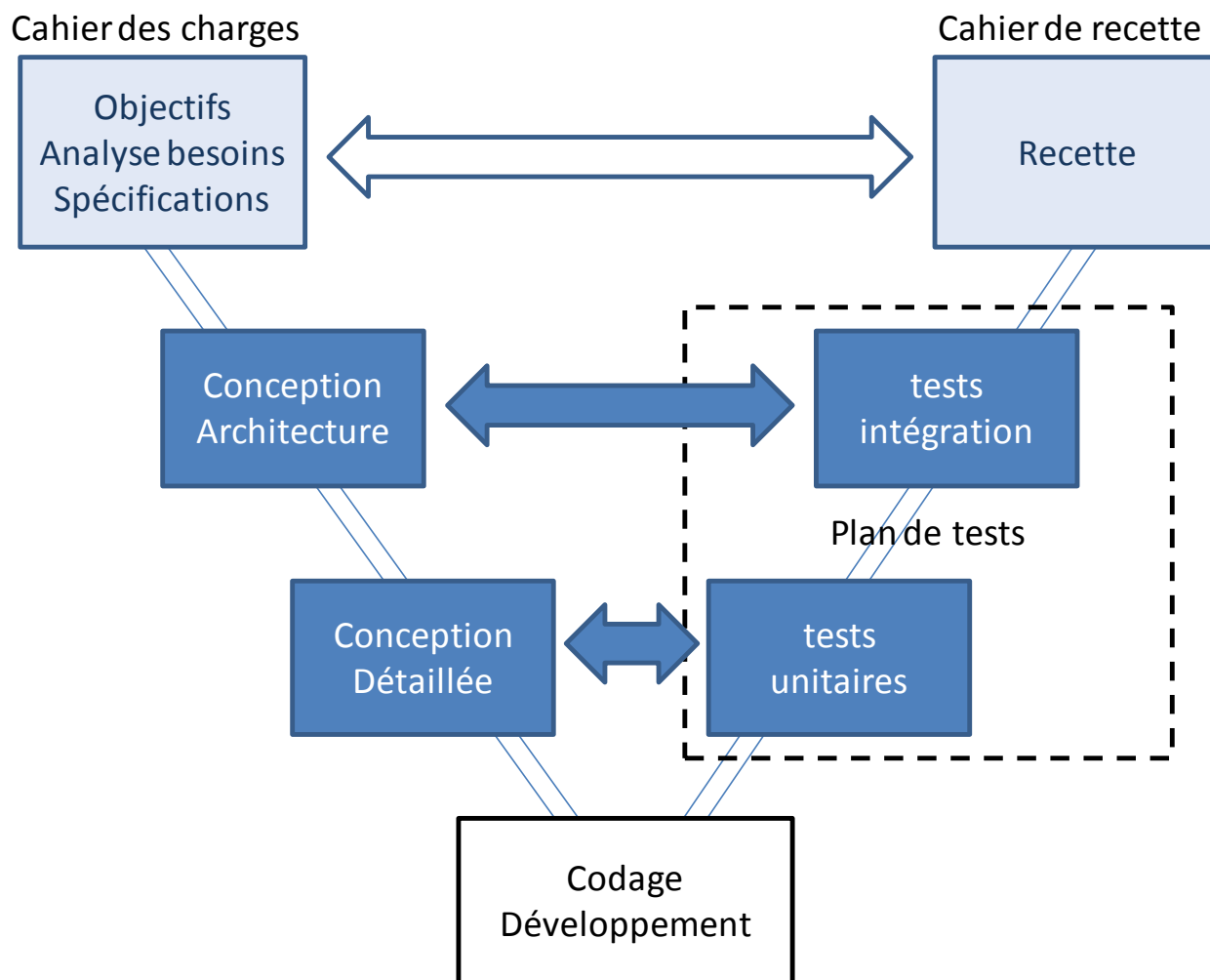
Cahier des charges

Manuel d'utilisation
Manuel d'installation
Plan des tests
Documentation interne du code

4 CYCLE DE VIE

Un développement suivant un cycle en V a été préféré pour ce projet. étant donné la faible complexité du projet, le risque d'erreur est bas; un cycle en V semble donc être adapté.

Le schéma ci-dessous illustre un cycle en V:



Un cycle en V classique est suivi. Chaque phase (rectangle) est vérifiée par celle opposée à sa droite.

Le texte noir en dehors des rectangles correspond aux documents représentant les phases inscrites dans les rectangles adjacents.

Le temps est représenté de la gauche vers la droite, les tâches sont effectuées dans l'ordre chronologique en parcourant le V. Le cycle commence en haut à gauche puis descend jusque la phase de développement pour ensuite remonter vers la phase de recette.

Chaque étape vérifiée par son étape opposée. C'est à dire par exemple que le cahier de recette a pour but de vérifier le cahier des charges, de même les tests unitaires sont développés pour vérifier la bonne conception des modules (conception détaillée) et ainsi de suite...

Le détail est schématisé du haut vers le bas. Soit les phases du haut représente un aspect global le projet et à l'inverse plus la phase est en bas du schéma plus cette phase représente en détails le projet.

5 MÉTHODES ET OUTILS

Les principaux outils utilisés dans le cadre du développement :

- **Développement / code source**

- **Plateforme**: chaque membre dispose d'un micro-ordinateur équipé du système d'exploitation Ubuntu 9.10, dernière version d'une des distribution GNU/Linux
- **IDE**: chaque membre utilise *Eclipse*, un environnement de développement intégré libre, auquel tous avons installer le *CDT (C/C++ Development Toolkit)* pour pouvoir programmer en C.
- **gestion de version**: les membres du groupe utilisent Subversion (*SVN*) un système de gestion de version libre.

- **Enregistrements**

- *Adobe Audition*: le membre de l'équipe s'occupant des enregistrements effectue ceux-ci avec le logiciel *Adobe Audition*. Logiciel qu'il exécute sous *Windows*. Un micro de bonne qualité est utilisé, un *T.Bone SC400* ainsi qu'un interface audio MIDI/USB *M-AUDIO - Fast Track Pro* pour connecter le micro au micro-ordinateur.

Adobe Audition est un logiciel propriétaire: il est chère et non disponible sur les plateformes Linux. Le chef de projet aurait préféré l'utilisation d'un logiciel équivalent tel *Audacity*. En effet *Audacity* est un logiciel open source et de ce fait bien plus accessible. Il est disponible sur toutes les plateformes et est distribué sous une licence libre soit sans frais.

Seulement le membre de l'équipe s'occupant des enregistrements a auparavant utilisé à plusieurs reprises le logiciel *Audition*, notamment pour effectuer des enregistrements donc il est très familiarisé avec celui-ci; cela explique le choix.

6 DOCUMENTATION

6.1 Présentation

Présentation des documents fournis ainsi que leurs plans-type

6.1.1 Manuel d'utilisation

Le manuel d'utilisation fournit des informations sur l'ensemble des fonctions de la librairie et de l'application de test et permet à l'utilisateur final d'exploiter la totalité des fonctionnalités qu'elles offrent. Il décrit aussi les éventuelles erreurs rencontrées en fournissant des informations dans le but les résoudre.

Voici son plan-type:

- **Librairie**
 1. **mise en œuvre** : Indique la manière d'utiliser la librairie pour réaliser les tâches de base
 2. **fonction** : décrit les commandes de la fonction de la librairie
 3. **message d'erreurs** : décrit les erreurs pouvant survenir et les procédures permettant leur résolution
- **Application**

liste des opérations nécessaires pour utiliser l'application de test

6.1.2 Manuel d'installation

Le manuel d'installation énonce les procédures nécessaires à la mise en place de la librairie dans l'environnement de développement et à son bon fonctionnement. Pour l'application de test, le document fournira les étapes à suivre pour une bonne exécution de celle-ci.

Voici son plan-type:

- **Librairie**
 1. **installation sur la plateforme**: liste des opérations nécessaires pour installer la librairie
 2. **inclusion dans le code source**: liste des opérations nécessaires pour inclure la librairie dans le code source d'un programme
- **Application**

liste des opérations nécessaires pour installer l'application de test

6.1.3 Documentation interne du code

La documentation interne du code a pour but de faciliter la compréhension du code source de la librairie: elle fournit des informations techniques sur le fonctionnement de celui-ci. Chaque section de code est commentée afin de définir ses exigences et son rôle.

Plan-type:

1. **architecture**: liste explicite de l'architecture de la librairie
2. **description des modules**: description détaillée de chaque module

6.1.4 Plan des tests

Le plan des tests est un document illustrant les tests effectués pour s'assurer du bon fonctionnement de l'application. Les tests d'intégration et les tests unitaires y sont décrits.

Plan-type:

1. **liste des tests:** liste de l'ensemble des tests ainsi que leur ordonnancement.
2. **description des tests d'intégration:** vérification de la bonne communication entre chaque module.
3. **description des tests unitaires:** vérification complète du bon fonctionnement de chaque module, toutes les possibilités d'exécution y sont testés, cela par un script.

6.2 Standard et outils

La rédaction de la documentation est effectuée à l'aide des suites bureautiques les plus répandues (*OpenOffice.org*, *Microsoft Office*, *Apple iWork*), majoritairement celle de *Microsoft*.

Word pour la mise en forme du texte, *Excel* pour la réalisation de tableaux et *PowerPoint* pour la réalisation des schémas.

La documentation interne du code sera accompagnée d'une documentation générée à l'aide de *Doxygen*.

7 QUALITÉ

les normes de qualité du projet sont les suivantes:

- restituer vocalement les numéros de la façon la plus fluide possible.
- effectuer des enregistrements de bonne qualité.
- tendre vers un minimum de bogues.
- rédiger documentation très claire et accessible de tel sorte que n'importe quel autre développeur puisse reprendre le code.
- commenter le plus possible le code de façon précise et concise.
- concevoir des tests unitaires explicites, illustrant au mieux le fonctionnement de chaque module.

8 GLOSSAIRE

-A-

Adobe Audition : Logiciel de manipulation de données audio numériques et de traitement du signal audio.

Audacity : Logiciel libre pour la manipulation des données audio numériques

-C-

CDT : Acronyme ayant pour référence le plugin d'*Eclipse* (*C/C++ Development Toolkit*)

-E-

Eclipse : Environnement de développement intégré (IDE) libre, extensible et polyvalent (supporte plusieurs langages).

-S-

SVN : *Subversion*, système de gestion de versions au cours du développement

-U-

Ubuntu : Système d'exploitation basé sur *Debian*, une des nombreuses distributions *GNU/Linux*

9 RÉFÉRENCES

Cahier des charges *S1*

Cahier de recette *S2*

10 INDEX

Cycle de vie, 7

Documentation interne du code, 9

Eclipse, 8

Enregistrements, 8

Gestion de versions, 8

Librairie, 3,9

Manuel d'installation, 9

Manuel d'utilisation, 9

Plan des tests, 10

Planification, 6

Synthèse vocale, 3

Tests unitaires, 5

Ubuntu, 8