

Plan de Développement



SOMMAIRE

1. Préambule	3
2. Planning prévisionnel.....	4
3. Organisation de l'équipe projet.....	5
4. Plan qualité	6
4.1. La démarche qualité	6
4.2. Moyens mis en place	6
4.3. Méthodes logicielles	6
4.4. Contrôle prévus et planifiés	7
4.5. Conventions de codage	7
4.5.1. En-têtes	7
4.5.2. Règles de codage	8

1. Préambule

La deuxième année de Master de l'Institut Universitaire Professionnalisé mention Systèmes Intelligents intègre dans son programme un Projet de Grande Envergure (PGE). Celui-ci mettant en œuvre un bras haptique, il est demandé à la promotion de développer un système interactif multimodal incluant ce périphérique.

Ce document présente les 3 parties spécifiques au plan de développement du PGE 2005 :

- Le planning prévisionnel
- L'organisation de l'équipe projet
- Le plan qualité

Le planning prévisionnel correspond aux délais prévus pour chaque tâche du projet. L'organisation de l'équipe présente les moyens humains prévus pour la bonne conduite du projet. Enfin le plan qualité aborde les points suivants : les objectifs à atteindre, les moyens mis en œuvre, les contrôles à effectuer ainsi que les règles à respecter.

2. Planning prévisionnel

Le planning prévisionnel a été fixé lors de la présentation du projet. Nous y avons apporté quelques modifications, notamment en ce qui concerne l'avancement du projet. (Voir planning ci-joint).

Voici la liste des livrables à fournir au client avec les échéances :

A la réunion d'initiation du projet (11/10/2005) :

- Plan de développement
- Cahier des charges (1 semaine après)

A la réunion de spécifications (07/11/2005) :

- Analyse fonctionnelle
- Bilan organisationnel

A la réunion de conception générale (02/12/2005) :

- Dossier de conception générale
- Bilan organisationnel
- Prototype

A la réunion de conception détaillée (20/12/2005) :

- Dossier de conception détaillée
- Bilan organisationnel
- Prototype

A la réunion de pré validation (début 02/2006) :

- Bilans des tests par équipe
- Bilan organisationnel

A la réunion de la recette (21/02/2006) :

- Bilan des tests d'intégration
- Bilan organisationnel

A la livraison (24/02/2006) :

- CD contenant tous les documents, le code source, un exécutable, les manuels utilisateur et programmeur.

3. Organisation de l'équipe projet

Nous savons d'ores et déjà que nous allons nous partager le travail en équipe mais la répartition des personnes dans chacune d'elles n'est pas encore définie. Voici une ébauche des groupes de travail que nous sommes susceptibles de former :

- 3 à 4 personnes sont en charge de la gestion du projet : chef de projet, gestion des communications entre les équipes, communication également entre les équipes et les experts/clients, gestion de l'organisation logistique (prévoir les réunions, réservation de salle en cas de besoin, ...). Ces personnes sont aussi en charge de la qualité.
- Une équipe pour le vocal,
- Une équipe pour la synthèse d'image,
- Une équipe pour le bras haptique,
- Une équipe pour le réseau,
- Une équipe pour la fusion/fission multimodale,

N'ayant pas encore réalisé l'analyse fonctionnelle, nous ne pouvons dire combien de personnes composent ces équipes. Cela dépend des fonctionnalités définies (donc la charge de travail prévue) pour chacun des modules. Chaque équipe aura un représentant responsable de son groupe et de la communication avec les autres équipes.

Des réunions hebdomadaires sont prévues entre les chefs d'équipe et l'équipe gestion de projet afin de veiller à la bonne conduite du projet, au respect des délais ainsi qu'au respect des règles définies par le plan qualité.

Les chefs d'équipes sont chargés d'organiser des réunions internes pour discuter des problèmes à résoudre et, le cas échéant, faire remonter les problèmes mettant en cause le travail d'autres équipes.

4. Plan qualité

4.1. La démarche qualité

Le but de notre projet est de fournir aux promotions SI futures une bibliothèque robuste, générique et évolutive autour du thème du bras haptique et de l'environnement 3D virtuel.

Notre objectif principal et implicite est d'aller au « bout » du projet. C'est-à-dire de développer un maximum de fonctionnalités. Cependant la qualité et la robustesse de ces dernières primeront sur leur nombre. Ainsi nous viserons au plus juste quant au nombre de fonctionnalités initialement prévues, quitte à en ajouter si le planning nous le permet.

4.2. Moyens mis en place

Les moyens humains sont composés de la totalité de notre promotion, c'est-à-dire 24 personnes. Certaines assureront une double fonction (chef d'équipe, de projet, ...). Ces personnes auront à charge la gestion de la communication aussi bien avec l'équipe gestion de projet qu'avec les experts/clients. Nous bénéficions également de l'aide d'experts pour chacun des modules.

Les ressources dont nous souhaitons disposer sont les locaux de l'AIP et des salles de réunion. Nous utiliserons les ordinateurs de l'AIP, de la salle 112 en U3 ainsi que des ordinateurs personnels. Nous demandons un accès total au bras haptique fourni par les clients.

Concernant la maintenance, nous nous engageons à mettre à jour régulièrement le site mis à notre disposition (<http://www.desssi.ups-tlse.fr>). Tous les documents y seront directement téléchargeables, au format pdf. Nous voulons également utiliser le site à des fins de communication avec la mise en place d'un forum.

4.3. Méthodes logicielles

Le déroulement de ce projet suit un cycle en cascade par prototypage (voir planning). Des prototypes concernant notamment les parties réseau, interface et parole sont envisagés. Nous ne sommes pas encore sûrs à ce jour d'utiliser un logiciel de gestion de projet tel que MSProject ou GanttProject.

Il est également possible que nous utilisions un logiciel de gestion de version tel que SubVersion. Ce logiciel serait installé sur un serveur personnel qui hébergera également des sauvegardes de tous les documents téléchargeables sur le site mis à notre disposition.

Nous imposons aux développeurs une certaine mise en page de leur code source afin d'assurer aux clients une certaine lisibilité et une meilleure compréhension. De plus, cela rendra plus aisée la fusion des différents modules (cf. partie 4.5). Pour cela, nous nous baserons sur l'utilisation du logiciel « Doxygen ».

4.4. Contrôle prévus et planifiés

La personne responsable de la qualité ainsi que les chefs d'équipe doivent réaliser les contrôles nécessaires au respect des règles de codage et du planning. Ils rendront compte à l'équipe de gestion de projet, et notamment au responsable de la qualité, du résultat de leur contrôles. Ces derniers prendront alors les décisions nécessaires en cas de non respect de certaines règles.

Les dates de ces contrôles seront laissées libres aux chefs d'équipe. Cependant trois dates seront fixées pendant la phase de codage afin de vérifier l'homogénéité de la qualité du codage dans chaque équipe.

4.5. Conventions de codage

4.5.1. En-têtes

→ En-tête de fichier :

L'en-tête d'un fichier doit contenir le nom du projet (= « PGE 2005 MOVE IT »), le nom du fichier, les auteurs ayant participé au codage du fichier, la date de dernière modification, ainsi qu'une description rapide du contenu du fichier.

Avec le logiciel Doxygen :

```
/*! PGE 2005 : MOVE IT
 * \file nom du fichier
 * \author nom des auteurs
 * \date 16/10/05
 * \brief description rapide du contenu du fichier
 */
```

→ En-tête de classe :

L'en-tête d'une classe doit contenir le nom de cette dernière, la description du rôle de la classe, ainsi qu'un listing de chacun de ses attributs avec leur rôle.

Avec le logiciel Doxygen :

```
/*! \class nom de la classe
 * \brief description du rôle de la classe
 * Attributs :
 * - Attribut 1 : servant à ...
 * ...
 * - Attribut n : servant à ...
 */
```

→ En-tête de fonction :

L'en-tête de chaque fonction contient son prototype, la description de son rôle, la liste de ses paramètres, et spécifier le retour de la fonction.

Avec le logiciel Doxygen :

```

/*! \fn prototype de la fonction
* \brief description du rôle de la fonction
* \param premier paramètre + son type
...
* \param dernier paramètre + son type
* \return le retour de la fonction
*/

```

4.5.2. Règles de codage

→ Le nom des fonctions et des variables (en anglais) doit être clair et spécifique à leur utilisation. Dans un soucis d'uniformisation et si plusieurs mots sont nécessaires, nous les « collons » les uns aux autres et nous les séparons par une majuscule.

Ex : nomFonction, nomVariable, ...

→ L'ouverture et la fermeture d'une accolade se fait après un retour à la ligne (fonction, classe, bloc de code).

→ Une limite de 80 à 100 lignes de code est fixée pour chaque fonction.

→ Le nom d'une classe doit obligatoirement commencer par une majuscule.

→ Afin d'améliorer la lisibilité du code, nous identifions les variables par leur type.

```

Entier      : iNomInstance
Caractère   : cNomInstance
Booléen     : bNomInstance
String      : sNomInstance
Flottant    : fNomInstance
Double      : dNomInstance
Void        : vNomInstance

```

```

Unsigned    : uTypeNomInstance
Ex : unsigned int uiEntier ;

```

```

Variable Globale : GTypeNomInstance
Ex : int GiEntier ;

```

```

Pointeur : ptrTypeNomInstance
Ex : int * ptriPointeurEntier ;

```

```

Classe : claNomInstance (cla = trois premières lettres du nom de la classe)
Ex : class Matrice ;

```

```

...
Matrice MatCarree ;

```


- Le code doit contenir des commentaires devant toutes les parties importantes de l'algorithme.