Université de Cergy-Pontoise

Projet Mouvements de Patterns

Plan Assurance Qualité

David Barouh
Marc Teyssier
David Liebgott

Tuteurs :
Mr Philippe Laroque
Mr Pierre Andry

Année 2007-2008
Projet de Master 2
V1.42

Table des matières

1 Introduction

Le Plan d'assurance qualité a pour but de définir l'ensemble des moyens mis en oeuvre pour garantir la cohérence du projet dans sa globalité, mais aussi de s'assurer que le produit réalisé répond aux attentes du client. Tous les membres de l'équipe devront se tenir informé de chacun des points de ce document.

2 Responsabilité de l'assurance qualité

2.1 Désignation du responsable

L'ensemble des membres de l'équipe sera responsable de la qualité du projet.

2.2 Procédures d'application et de non application

Ce plan sera en application dès le début du projet pour tous les membres de l'équipe sans exception. Chacun peut vérifier auprès d'un autre le respect des dispositions prises dans ce plan à condition que tous les autres membres soient tenu informé. Suite à une anomalie, un accord commun devra être trouvé par n'importe quel moyen de communication. Une fois l'accord établi, la personne la plus apte à appliquer les modifications sera en charge de les réaliser.
Si on ne peut appliquer cette procédures, il faudra dans un premier temps faire une réunion afin d'étudier l'impact sur l'ensemble du projet. Il engendrera automatiquement une mise à jour du PAQ qui devra répondre à ce problème. De plus, les documentations incriminées devront être revues afin d'apporter les modifications nécessaires. En fonction de la gravité du manque, une réunion avec le client peut être programmée pour l'avertir au plus vite.

3 Suivi de l'exécution du plan

3.1 Ensemble des dispositions à mettre en œuvre pour la bonne application du plan qualité

Le suivit du projet est obligatoire pour garantir que chaque membre de l'équipe ait toutes les connaissances sur tous les développements logiciels et matériels. Chacun peut suivre le planning d'avancement du travail et des réunions sur Google Agenda. De plus, afin de faciliter les tâches à effectuer et de tenir les objectifs fixés, un tracker de bug (Mantis) a été installé sur le site Web. Il sera fournit aux tuteurs un compte en lecture seule pour suivre l'avancement.
Pour augmenter le partage des connaissances, des pages Internet de type Wiki ont été mis à la disposition des membres de l'équipe sur le site Web. Elles permettent notamment d'enregistrer des liens vers des sites intéressants ou les idées trouvées hors d'une réunion ou de rendez-vous fixe.
Pour que tous les membres de l'équipe soient au courant de tous les éléments, une adresse e-mail unique a été créé. Celle-ci redirige tous les messages vers la boite e-mail de chacun des membres. Grâce à ce système aucune personne n'est mise à l'écart lors de la communication interne ou externe (que ce soit pour une question ou autre).
Un gestionnaire de versions (SVN) nous a été fournit par l'université afin de pouvoir stocker notre projet (que ce soit du code ou des documentations). Toute implémentation de code ou d'une documentation sera obligatoirement "versionnée", ce qui garantira que chaque membre travail sur la dernière version.

3.2 Modalité de déroulement des activités de contrôle qualité

Tout document devra être relu et approuvé par l'ensemble des membres de l'équipe. Il est de l'obligation de chacun de vérifier que le présent plan est correctement respecté. Lors de toute réunion il faudra suivre le procédure suivante :

- En début, contrôler l'avancement de chacun.
- Pendant, s'assurer de la compréhension de chaque membre (que ce soit techniquement ou fonctionnellement).
- En fin, la répartition équitable des tâches.

Lors de la clôture d'une version, l'équipe devra valider les modalités de la suivante.

Durant le projet, si un membre est en désaccord avec les dispositions prises, il devra informer l'équipe entière en formulant clairement le point de divergence ainsi que les arguments qui le soutient. S'il ne propose pas de solution alternative, une réunion extraordinaire sera réalisée, par n'importe quel moyen de communication, pour résoudre ce problème le plus rapidement possible.

3.3 Qualité des documents applicables

Une charte graphique a été mis en place pour tous les documents produits par l'équipe DMD. Elle est principalement composée d'un modèle de document et d'un logo. Cette charte graphique devra être utilisée pour tous les documents

produits lors de ce projet, ainsi que pour toute communication externe, exception faite du site Web qui aura une mise en page spéciale.
Tous les documents devront être rédigé aux formats LaTeX ou OpenOffice pour assurer la compatibilité quelque soit le système d'exploitation.

3.4 Terminologie

Tout mot spécifique devra être expliqué afin que les documents soient accessibles au plus grand nombre Toute abréviation devra être développée dans sa première utilisation. Si le mot ou l'abréviation revient souvent, il devra être ajouté dans un lexique.

4 Exigences, risques et contraintes du projet

4.1 Référence aux éléments mis en évidence par le cahier des charges

Avoir une Webcam reliée à un ordinateur ou un PDA. Transmettre le flux vidéo à un serveur qui devra traiter les images et renvoyer un code (se référer au protocole réseau). Des applications récupéreront ce message et le convertiront en action.

4.2 Exigences fonctionnelles

Le système devra pouvoir fonctionner quelque soit le nombre d'applications qui récupèreront les données envoyées avec le protocole défini. Le système devra offrir un fonctionnement simple où l'utilisateur est au centre de toutes les questions de réalisation afin d'avoir un outil suffisamment intuitif.

4.2.1 Matériels

- Une Webcam USB Logitech QuickCam Pro 4000 (et une seconde pour pouvoir développer à plusieurs)
- Un ordinateur qui devra récupérer le flux vidéo et l'envoyer sur le réseau. Il est définit qu'il peut être remplacé par un PDA HP hx4700, selon les versions du projet
- Un ordinateur qui aura le rôle de serveur (traitement de l'image et envoi sur le réseau). Une version future pourra le remplacer par un routeur Wifi pour avoir une version compacte de la solution.
- Un routeur Wifi
- Des servos-moteurs qui serviront d'actionneur.

Les ordinateurs auront comme système d'exploitation une distribution de Linux, Ubuntu, ou sa dérivé Kubuntu. Une utilisation de Windows n'est pas exclu.

4.2.2 Interface de communication

- Ethernet : Type de réseau local, le débit utilisé sera 100Mbps
- Liaison RS232 : pour les servos-moteurs débit 8600Bauds
- Wifi : Réseau de communication sans fil. Sera utilisé en mode b (11Mbps) ou g (54Mbps)

4.3 Exigences organisationnelles

Lors de la création du groupe DMD, d'un commun accord, la décision a été prise de ne pas avoir d'organisation hiérarchiques au sein du groupe. Ainsi chaque membre est l'égal de tous les autres, aussi bien au niveau des droits (de décision, de paroles...), que des devoirs. Ceci implique que chacun des membres a une connaissance du projet dans ses moindres détails, et est donc apte a prendre seul une décision pour le projet, à écrire une documentation complète, voir même à présenter le projet.

Cette organisation implique une prise de décision différente des autres types de hiérarchies : une décision est prise lorsque tous les membres du groupe sont

d'accord. Si ce n'est pas le cas, un débat d'arguments constructifs a lieu jusqu'à ce que, soit la personne ayant proposé la décision change d'avis, soit la personne qui était en désaccord change d'avis, soit, ce qui arrivera le plus souvent, une meilleure idée soit trouvée.

4.4 Exigences de contraintes

4.4.1 Contraintes majeures

Il devra être fournit une version du projet parfaitement opérationnelle quelque soit l'avancement finale. Les différentes documentations et le site bilingue devront être finalisés.
Un guide utilisateur devra être fournit, permettant l'installation de la solution, et sa prise en main.
La possibilité de l'ajout simple d'une application devra être présente.

4.4.2 Analyse des risques et des mesures préventives

Traitement de l'image Les membres du groupe ont très peu de notions de traitement d'image. C'est pourquoi ils ont décidé d'utiliser une librairie Open Source pour simplifier cette partie : OpenCV. Cette librairie est accompagnée d'exemple avec les différentes utilisations fondamentales. De plus, le nombre de technique de traitement de l'image sera réduit au maximum afin de minimiser les risques et le temps d'étude de principes trop compliqués vis à vis des buts fixés.

Flux vidéo Définir le protocole de communication utilisé pour transmettre le flux vidéo. En effet, c'est un point sensible du projet, car le traitement sera déporté. Si le protocole est mal choisit, le réseau sera saturé. L'équipe se basera soit sur une utilisation d'un logiciel libre comme VLC (VideoLAN Client) ou sur leur connaissance en C/C++ pour optimiser cette communication.

Le langage Afin de ne pas être bloqué sur un problème de syntaxe ou de fonction, le langage utilisé sera le C++ qui possède une communauté suffisamment importante pour minimiser les ralentissements dus au langage.

L'ensemble du projet Pour avoir un système opérationnel à la date du 12 juin 2008, le projet est découpé en plusieurs versions. Celles-ci seront validées par l'équipe et par le client. Après quoi il n'est plus possible de la modifier. Ceci garanti qu'au moins une version du projet fonctionnera, même si la dernière évolution n'est pas finalisée.

5 Organisation et communication du projet

5.1 Identification des intervenants

Intervenant	Fonction
M. Pierre ANDRY	Tuteur Université
M. Philippe LAROQUE	Tuteur Université
M. David BAROUH	Membre de l'équipe
M. Marc TEYSSIER	Membre de l'équipe
M. David LIEBGOTT	Membre de l'équipe

FIG. 1 – Identification des intervenants

5.2 Définition des responsabilités

Le fonctionnement de l'équipe est atypique, en effet, il n'y a pas de responsable déclaré, chaque membre est en droit de prendre des décisions si et seulement si tous les autres membres donnent leur accord. De plus, chacun devra veiller à ce que l'ensemble des documents soit respectés, que ce soit le plan d'assurance qualité ou le cahier des charges. La responsabilité de l'ensemble du projet repose donc sur toute l'équipe, sans exceptions.

5.3 Définition et organisation des moyens

5.3.1 Communication

Différents outils sont à disposition de l'équipe pour faciliter les communications :

- Un email unique pour tout le groupe : dmdm2p@gmail.com
- Un espace Web : http ://dmdm2.free.fr
- Un serveur SVN (mis à disposition par l'université)
- Un tracker de bugs
- Un Wiki
- Différentes messageries instantanées
- Le téléphone

5.3.2 Matériel

Chaque membre du groupe possède son propre ordinateur portable afin de pouvoir développer ou rédiger des documents.
De plus, l'université nous fournit du matériel, voir la partie Matériels au paragraphe 4.2.1, page 5.

5.4 Définition de la formation

Les membres de l'équipe devront apprendre par eux même à manipuler les librairies utilisées et se former sur les notions de traitement d'image, le traitement en temps réel et approfondir leur connaissance des protocoles réseaux.
De plus, de par l'architecture du projet, de nombreux domaines devront être abordés, qui permettront à l'équipe d'expérimenter de nouvelles choses.

6 Méthode et processus d'ingénierie

6.1 Les phases du processus

Les phases du processus d'ingénierie sont les suivantes :

1. Réunion de lancement
   - Étude des orientations techniques et des choix fonctionnels
2. Création du cahier des charges
   (a) Besoins du client
   (b) Phases du projet
   (c) Planning
   (d) Charte graphiques
   (e) Plan d'Assurance Qualité
3. Analyse technique approfondie
4. Développement d'une version
5. Recette d'une version
6. Étape 4 tant que le planning est tenu
7. Documentation

6.2 Livrables

6.2.1 Sur support papier

L'ensemble de la documentation : le cahier des charges, le plan d'assurance qualité, le manuel d'installation et le manuel d'utilisation.

6.2.2 Sur support multimédia

Le code source du projet.
L'exécutable du projet, pour le système d'exploitation Linux, au minimum.
L'ensemble des documentations au format PDF.
Le site Web du projet sous forme d'archive.

6.3 Activités de production

6.3.1 Analyse

Avant tout développement, une analyse des besoins a été nécessaire, pour permettre de répondre au mieux aux besoins des clients.
De plus une analyse de l'existant fut essentielle pour partir sur de bonnes bases.

Par la suite, il a été nécessaire d'analyser en détail chacune des parties à effectuer pour permettre de créer un planning prévisionnel collant le plus possible à ce que sera la réalité.

6.3.2 Développement de l'application

Le découpage du développement sera effectué en quatre phases principales :

Serveur de traitement Cette phase est la partie centrale de l'application. En effet, elle comprend toutes les fonctionnalités de traitements d'images et de traduction en code qui seront utilisées par le reste de l'application.

Flux vidéo Cette partie représente la déportation de la capture du flux vidéo. Un flux vidéo en entrée étant essentiel à l'application, le fait de pouvoir capturer la vidéo sur un système différent que le serveur est critique.

Interface Homme-Machine Dans cette phase de développement aura lieu la mise au point de l'interface graphique de l'application. Etant donnée que l'interface représente le point d'entrée de l'utilisateur dans l'application, elle est essentielle.

Gestion des applications externes qui géreront des actionneurs Seront contenu dans cette phase, de multiples développements. En effet, les applications externes (ou Binap, Binap Is Not A Plug-in) devront être développées pour chaque actionneur. Ces Binaps représentent l'utilité réelle de l'application. Plus ils seront nombreux et fonctionnels, plus l'application aura un grand intérêt dans son utilisation quotidienne.

7 Démarche

7.1 Cycle de vie

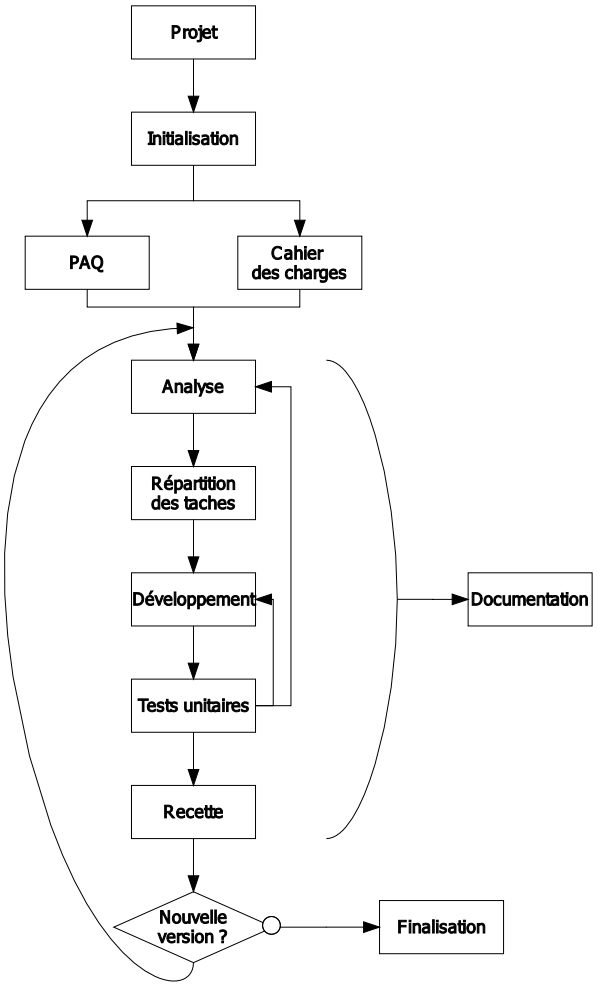

FIG. 2 – Cycle de vie

7.1.1 Description détaillée des différentes phases

– Initialisation : Cette phase consiste à avoir un ensemble de réunion avec les tuteurs et l'équipe puis seulement l'équipe afin de définir les objectifs réalisables ou non. Il s'en suit une étude fonctionnelle et technique globale (sans entrer dans les détails) et de les faire valider par les tuteurs qui ont une meilleure expérience des réalisations pratiques.

- PAQ : Réalisation d'un ensemble de dispositions qui garantiront l'intégrité du projet tout au long de son cycle de vie et auxquelles l'équipe devra se conformer durant tout le projet.

- Cahier des charges : Devra cerner le projet dans son ensemble afin d'y apposer des limites fonctionnelles et techniques. Il devra aussi proposer tout un ensemble de solutions fonctionnelles et techniques retenues pour la réalisation.

- Analyse : analyse générale puis détaillé de la version à réaliser. Elle devra assurer les choix fonctionnels, techniques, mais aussi ergonomiques.

- Répartition des tâches : Suite à l'analyse, l'objectif à réaliser sera découpé en un ensemble de fonction unitaire en vu d'être codée par les différents membres de l'équipe.

- Développement : Chaque membre devra développer sa ou ses fonctions et réaliser les tests unitaires appropriés.

- Test : Regroupement des fonctions par rapport à leur but (IHM, traitement de l'image...) et tests.

- Recette : Finalisation d'une version donnée en fonction du cahier des charges.

- Documentation : Réalisation des documentations d'installation et d'utilisation de chaque version.

- Finalisation : Compte rendu final avec la livraison des éléments définis dans la partie Livrables 6.2, page 8.

7.2 Activités de vérification

Chaque version devra subir un ensemble de test de validations qui devront répondre au cahier des charges. Une fiche répertoriant l'ensemble des tests devra figurer pour chaque version. Une relecture avec les tuteurs devra être prévu pour valider l'application.

7.3 Les contrôles qualités applicables

Chaque membre de l'équipe devra s'assurer du respect de la qualité. C'est pourquoi il est obligatoire que tous les membres de l'équipe contrôlent l'ensemble des documents et du code réalisés.

7.4 Activités de gestion de projet

Des réunions au sein de l'équipe devront être réalisé régulièrement pour valider le travail effectué par chacun et programmer la suite des orientations à prendre. De plus, une réunion avec les tuteurs devra avoir lieu après chaque phase majeure du projet (notamment les changements de version) mais aussi si

l'équipe rencontre des difficultés ou des interrogations auxquelles elle ne trouve pas de solutions.

8 Règles pour le code source

Les règles de codage dans ce document devront être respecté durant tout le projet par l'ensemble des membres de l'équipe DMD.
Le langage utilisé étant le C++, l'ensemble des fonctions implémentant une facette particulière du programme devra être regroupé dans une classe. Les fichiers devront avoir le même nom que la classe, amputé du préfixe 'C'. Pour une classe CTEST le fichier sera *test.cpp* et les headers seront regroupés dans un fichier *test.h*. Le fichier *test.cpp* ne devra inclure que le header qui lui est associé. Toute autre déclaration se fera dans le fichier *.h*.

Chaque classe et chaque fonction devront être commentées à la manière d'une JavaDoc :

```
/** Fonctionnalités générales de la classe
  *
  * Explications détaillées
  */
```

Les fonctions devront réaliser des calculs les plus élémentaires possibles afin d'augmenter la possibilité de les réutiliser. Elles devront posséder des noms les plus explicites possibles. Les commentaires seront de la forme :

```
/** Fonctionnalités de la fonction
  * @param nom_du_paramètre1 Détails du paramètre 1
  * @param nom_du_paramètre2 Détails du paramètre 2
  * @return Détails de l'élément retourné
  */
```

Les noms des variables devront être de la forme :

$$[in\_|out\_|io\_|g\_|m\_|l\_]\,[type]\,NomDeLaVariable$$

Si la variable est un paramètre passée à une fonction elle aura au début du nom soit "in_", si elle représente un paramètre qui n'est utilisé qu'en entrée, soit "out_", si elle représente un paramètre utilisé uniquement en sortie, soit "io_", si elle est utilisée en entrée/sortie.
S'il s'agit d'un attribut de la classe (variable membre) ce sera "m_", pour une variable globale ce sera "g_", et pour une variable locale ce sera "l_".
Ensuite on aura le type :

Type	Abréviation
int	i
boolean	b
Enumeration	e
char*	str
Pointeur	p
Objet	o

FIG. 3 – Abbréviations

Si le type est une structure, l'abréviation devra être facilement identifiable. Les types peuvent se combiner (l'abréviation pour un pointeur d'entiers est

donc “pi”).
Le nom devra représenter au mieux la fonction de la variable. Elle est écrite avec une majuscule à chaque début de mot.

Quelques variables échappent à ces règles, ce sont les variables utilisées pour les boucles. N’étant représentée que par une lettre, il n’y a pas lieu de préciser le type.

9 Gestion des modifications / Corrections

9.1 Gestion des non-conformités

En cas d'une application non conforme au plan d'assurance qualité, un rapport de quelques lignes avec l'application incriminée ainsi que les raisons devra être rédigé et envoyé par mail avec l'adresse électronique du groupe. Il s'en suivra une réunion par n'importe quel moyen (physique ou par internet) pour évaluer les conséquences que cela produira sur la suite du projet aussi bien sur le développement que sur les retards que cela pourrait impliquer.
Puis un des membres de l'équipe devra corriger l'élément incriminé.

9.2 Gestion des demandes d'évolution

Les demandes d'évolution devront être formulées par mail. Une réunion avec le client et l'équipe devra être mise en place pour évaluer les modifications à apporter aux différentes analyses mais aussi au planning.
Ensuite il faudra modifier le cahier des charges en conséquence, si l'évolution à lieu d'être ou pas.

9.3 Reproduction, sécurité, livraison

La reproduction de tous documents que ce soit en partie ou dans l'intégralité ne peut ce faire qu'avec l'accord de l'ensemble des membres de l'équipe.
Chaque membre de l'équipe aura l'ensemble du projet (documentation et code source) sur support multimédia (CD-ROM).
Une version sera également fournit au client avec un format papier des différents documents (Cf Livrable 6.2, page 8)
.

10 Gestion de la documentation

10.1 Page de garde

Cf Annexe 11.1, page 17

10.2 Gestion de la documentation

10.2.1 Identification

Chaque document devra être référencé avec :

- Le nom du projet
- Le nom du document
- Le numéro de version

Cette notation permettra de faciliter l'archivage des différents documents. Ainsi chaque membre de l'équipe devra avoir accès à l'ensemble des documents versionnés.

10.2.2 Statuts des documents

Un document ne peut être distribué que si son statut est "validé". Sinon il est "en cours".

11 Annexes

11.1 Page de garde pour chaque document relatif au projet

Université de Cergy-Pontoise

Projet Mouvements de Patterns

TITRE DU DOCUMENT

David Barouh
Marc Teyssier
David Liebgott

Tuteurs :
Mr Philippe Laroque
Mr Pierre Andry

Année 2007-2008
Projet de Master 2
VERSION