

TER : Outil de recueil d'expérience DES-Application Cahier des charges

Etudiants chargés du projet :

Sébastien FAURE <sebastien.faure3@gmail.com>

Bertrand GROS <gros.bertrand@gmail.com>

Enseignant responsable du projet :

Yannick PRIE

Université Claude Bernard Lyon 1 – M1 Informatique – 2010-2011

Introduction

Dans le cadre du Master 1 Informatique de l'université Claude Bernard Lyon 1, les étudiants doivent réaliser un stage de recherche (TER) d'une durée de cinq semaines à partir du 3 janvier 2011.

Les étudiants Sébastien FAURE et Bertrand GROS ont choisi pour ce stage le sujet proposé par Yannick PRIE, enseignant-chercheur à l'université Lyon 1, de conception et réalisation d'un outil de recueil d'expérience utilisateur, avec l'accord de celui-ci.

I. Problématique

A. DESCRIPTION DES BESOINS

L'objectif du stage est de fabriquer un outil permettant de mener une collecte d'information sur l'expérience vécue d'un utilisateur d'un système informatique classique (systèmes d'exploitation et applications variés).

Le principe retenu est celui du Discrete Experience Sampling (DES), qui consiste à

interrompre un utilisateur de façon apériodique (par exemple 2 ou 3 fois par jour, ou bien 2 ou 3 fois par semaine) pour lui demander de répondre tout de suite à un ensemble de questions portant sur ce qu'il était en train de vivre / faire au moment de l'interruption. L'outil stocke alors les résultats et repart en tâche de fond jusqu'à la prochaine interruption.

Enfin le système mis en place doit être open source, compatible Mac OSX, simple à installer et utiliser et permettre l'intégration future d'extensions.

B. SOLUTION ENVISAGEE

Développement du logiciel en utilisant le langage Java pour utiliser les possibilités de ce dernier pour les interfaces graphiques ainsi que la présence d'API permettant la prise en charges des données qui seront manipulées mais surtout car ce langage permet de se rapprocher de la notion d'applications multiplateformes car les logiciels ainsi développés s'exécutent grâce à la machine virtuelle Java, elle-même indépendante du système d'exploitation de la machine de l'utilisateur.

Sqlite sera utilisé pour la base de données locale sur le poste utilisateur pour sa simplicité et rapidité.

Le projet suivra un modèle en spirale et plusieurs itérations seront développées (cf. partie planification). Nous avons fait ce choix car il est important de rendre un logiciel fonctionnel et le recours à l'itératif nous permet d'avoir des versions stables et testées tout au long du projet contrairement au modèle en cascade où les tests ne se font qu'à la fin.

Divers **outils gratuits** seront utilisés pour mener à bien ce projet.

Un **forum** gratuit destiné à faciliter la communication entre les membres de l'équipe de développeurs a été mis en place (URL : <http://des-application.forumpro.fr>).

Un projet **Google Code** a été créé. Ce sera l'outil principal de gestion de projet. Ses fonctionnalités rendront la conduite de projet plus aisée notamment en se concentrant sur les aspects suivants :

- Hébergement du code et des ressources (Serveur SVN intégré) ;
- Hébergement de fichiers accessibles en téléchargement direct ;
- Documentation du code et de l'application via un wiki interne et publication de comptes rendus d'avancement réguliers permettant de suivre les avancées du projet.
- Retour des développeurs et des utilisateurs via un système d'ouverture de tickets (appelés "issues") permettant de réagir efficacement aux problèmes de développement;
- Gestion des licences libres (le projet sera open source) ;
- Diffusion des différentes versions du logiciel directement via internet.

Son URL est la suivante : <http://code.google.com/p/des-application/> .

Gantt Project sera utilisé pour synthétiser le planning qui a été fixé (voir diagramme qui suit).

Pour l'édition de la documentation des programmes **Doxygen** sera utilisé.

LaTeX permettra de rédiger la documentation utilisateur alors que **Beamer** servira à créer la présentation de fin de projet.

II. Fonctionnalités

A. FONCTIONNALITES PREVUES

Dans le produit final un certain nombre de fonctionnalités seront possibles :

- Poser des questions sous forme de questionnaires à l'utilisateur de manière apériodique ;
- Récupération et stockage des réponses dans une base de données locale ;
- Importation et utilisation de fichiers de configurations pour l'application ;
- Réglages manuels de l'application : augmenter/diminuer la fréquence, temps d'attente avant réponse ;
- Définition de règles représentant les conditions selon lesquelles un questionnaire peut être proposé à l'utilisateur ;
- Consultation des réponses (en lecture seule) ;
- Exportation des réponses pour envoi manuel par mail à l'expérimentateur.

B. FONCTIONNALITES POSSIBLES

En fonction de l'avancement du projet certaines fonctionnalités supplémentaires pourront être ajoutées au logiciel :

- Application totalement multiplateforme : de l'installation à l'utilisation de l'application ;
- Collecte des réponses aux questions à distance grâce à Internet ;
- Autres améliorations/extensions...

III. Planification

Avant le début du stage nous avons dans un premier temps lu des articles sur le recueil d'expérience utilisateur (appelé « Discrete Experience Sampling », D.E.S.) puis nous avons rédigé le **rapport de conception** (complet et terminé) en concertation (disponible en téléchargement sur la page Google Code créée) avec Yannick PRIE, en y faisant apparaître notamment les contraintes d'architectures, les cas d'utilisations et le modèle du domaine de l'application.

Dès le début du stage, le 3 janvier nous pourrons commencer à mener les quatre itérations successives que nous avons planifiées.

A. REPARTITION DES TACHES

Sébastien FAURE et Bertrand GROS travailleront en collaboration constante sur les mêmes itérations et se partageront le développement des fonctionnalités propres à celles-ci pour être efficaces.

En parallèle à la mise en place des itérations successives, ils rédigeront la documentation utilisateur du logiciel.

B. PLANIFICATION ET DELAIS

Le projet sera réalisé à partir du 3 janvier 2011 et sera livré quatre semaines plus tard en version beta puis une semaine plus tard en version finale.

Des réunions quotidiennes entre les membres de l'équipe du projet sont prévues jusqu'au terme de celui-ci. De plus à l'image de ce qui est fait dans la méthode SCRUM (méthodologie itérative et incrémentale de gestion de projets logiciels), tous les matins, quinze minutes seront consacrées à faire un point sur les développements de la journée.

Deux réunions de préparation et de conception ont déjà eu lieu courant décembre avec Yannick PRIE.

Nous allons mener quatre itérations :

Itération 1 : Terminée à la fin de la deuxième semaine. Choix architecturaux qui fonctionnent (base,...). Cas d'utilisation mis en place: Répondre à une question.

Itération 2 : Terminée à la fin de la troisième semaine. Itération 1 avec intégration du système de questionnaires

Itération 3 : Terminée au début de la cinquième semaine. Itération 2 avec système export de données en XML et import d'un fichier de configuration en XML. Livraison en tant que version beta.

Itération 4 : Fin cinquième semaine. Itération 3 et correction/amélioration des points constatés par le client. Documentation (installation/utilisation) et aide interne à l'application. Livraison en tant que version finale.

Le diagramme de Gantt qui suit résume notre planification.

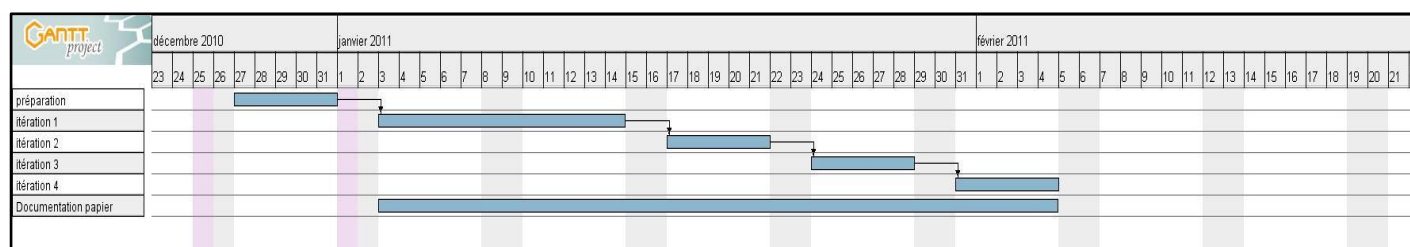


Diagramme de Gantt du projet