L3 INFO : Projet du module “Génie logiciel”

Spécification et test de l’application “Skaïpeuh”

Mihaela Sighireanu

sighirea@liafa.univ-paris-diderot.fr

Année 2013-2014

1 Objectif et rendu

Le but de ce projet est de vous faire pratiquer la spécification (en UML) et le test de logiciel.

L’objet de l’étude est l’application “Skaïpeuh” que vous devez développer pour le module “Programmation réseau”. *Les étudiants qui suivent ce module sans suivre le module “Programmation réseau” doivent contacter par mail avant le 2 Mai 2014 afin de vous fixer le sujet du projet à spécifier.*

Le rendu de ce projet sera un document qui décrit l’application “Skaïpeuh” pour un utilisateur et pour un spécialiste du domaine (développeur). Ces deux niveaux d’abstraction seront répartis sur les chapitres suivants du document :

Chapitre 1 : contient le cahier de charge (voir Section 2),

Chapitre 2 : contient la spécification interne (voir Section 3), et

Chapitre 4 : contient le plan de test (voir Section 4).

Le **Chapitre 3** du document doit lister des fonctionnalités implémentées dans votre projet.

2 Cahier de charges

Un cahier de charge est rédigé à la suite d’une analyse des besoins des utilisateurs. Il doit indiquer les problèmes (besoins) que le produit doit résoudre (satisfaire) de point de vue des utilisateurs, sans donner une solution technique. Il s’agit donc de synthétiser toutes les contraintes des utilisateurs (fonctionnelles, non-fonctionnelles, d’environnement, etc.) pour le produit à réaliser tout en restant non technique (sans détails d’implémentation).

Pour cela, vous devez utiliser le sujet du projet “Skaïpeuh”, le formaliser et le compléter. *Attention, cette partie spécifie les besoins et non l’implémentation faite. Autrement dit, toutes les fonctionnalités demandées par le sujet du projet “Skaïpeuh” doivent être spécifiés.*

Un exemple de plan pour le contenu du chapitre “Cahier de charges” est le suivant :

1. Description générale du produit avec les cas d’utilisation.
   L’emploi des diagrammes de cas d’utilisation est obligatoire.
2. Besoins (contraintes) du produit :
   (a) Besoins fonctionnels : détail non-technique sur les fonctionnalités de l’application pour chaque cas d’utilisation identifié. Vous pouvez y inclure des exemples d’interface utilisateur qui illustrent chaque fonctionnalité.
   L’emploi des diagrammes de séquence est obligatoire pour décrire les fonctionnalités de chaque cas d’utilisation. Vous pouvez compléter ces diagrammes avec des fiches semi-structurées et, si besoin, avec des diagrammes d’état transition ou de diagrammes

d'activités. Toutefois, veiller à rester non-technique et à ne pas dévoiler la structure interne du produit dans ces diagrammes.

Cette section doit détailler les fonctionnalités de l'application pour :
— les clients "avertis" du produit,
— chaque développeur de l'application pour qu'il sache quel produit programmer,
— pour l'équipe de test (voire l'équipe enseignante) pour définir les tests système à exécuter.

(b) Contraintes sur l'utilisateur du produit : quelles sont les conditions que l'utilisateur doit respecter pour utiliser le produit.

(c) Besoins techniques et d'environnement à respecter pour installer et utiliser le produit (par exemple : le type d'OS à utiliser, le type de réseau, la sécurité du système, etc.). *Pas de diagrammes pour ces deux derniers points.*

3 Specification interne

La spécification interne du produit est un document qui aide les développeurs à comprendre les choix de conception faits et le travail qui est à accomplir. Elle inclut au moins deux parties :

1. La spécification de l'architecture du produit, c'est à dire modules (paquetages, classes, interfaces pour Java) qui composent le produit.

   Si le langage de programmation est Java, un diagramme de classes doit être présent. Pour le langage C, le diagramme de composants doit être présent.

2. La spécification détaillée des modules.

   Les diagrammes de séquence, d'état/transition ou d'activités doivent être utilisées en impliquant les différentes composantes (méthodes, fonctions, états) d'un module. Si possible, des spécification en logique de Hoare peuvent être incluses pour chaque méthode/fonction.

4 Plan de test

Cette partie du document explicite les tests que vous avez appliqués pour vous assurer du respect du cahier de charges par l'application que vous avez programmée. La suite de tests écrite doit inclure :
— des tests unitaires pour les méthodes/fonctions développées,
— des tests de système pour les fonctionnalités demandés.
Une automatisation de l'exécution de ces tests en utilisant un système de type XUnit est un plus.

5 Organisation

Vous devez rendre le document (sur Didel, dans la section "Travaux") une semaine avant la soutenance du projet "Skaïpeuh". Le dépôt du document est fait par chaque étudiant, mais le document doit préciser les étudiants du groupe avec lequels vous avez travaillé.

Lors de la soutenance du projet "Skaïpeuh", des questions vous seront posés sur ce document.

Le projet est *obligatoire* pour les étudiants inscrits à ce module. Il compte pour 50% de la note finale.