



## **LIVRABLES DU PROJET « COMPOSANT AUTONOME »** *Développement d'un composant incluant son auto-test*

*Le produit logiciel que l'on vous demande de développer est un composant qui comprend deux services : i) le service de vérification syntaxique tel qu'il est défini dans le cahier des charges que nous avons finalisé en cours et ii) le service d'auto-test du composant.*

**Attention :** *le service de vérification syntaxique devra pouvoir être exécuté en mode interactif et en mode batch.*

*Les contraintes données pour la production de ce composant sont de développer en équipe et en parallèle le service d'auto-test. Ce développement devra se faire, indépendamment du service de vérification syntaxique, suivant le cycle en V donné en cours. C'est seulement lorsque ce service sera développé que le service de vérification syntaxique devra être développé suivant les phases du cycle en V. Vous utiliserez le logiciel subversion pour le développement réparti et pour gérer la configuration du logiciel. Vous installerez le Plugin Tigris qui permet d'intégrer Subversion dans la plateforme Eclipse*

*Organisation des équipes pour le développement en parallèle : huit équipes seront constituées sur la base de trois monômes par équipe.*

Un rapport de projet est à constituer par équipe. A titre indicatif, voici ce que je souhaiterais trouver dans ce rapport. Chaque rubrique doit être signée par son/ses auteur(s) et relecteur(s).

- 1. Contexte et objectif de la production logicielle demandée**
- 2. Spécification du produit logiciel**  
Il s'agit ici du composant final incluant les deux services publics (interface) cités en introduction.
- 3. Plan de validation du produit logiciel**  
Le plan de test permettant de valider le produit logiciel.  
Le jeu de test doit être représentatif de l'analyse partitionnelle et des tests aux limites.
- 4. Développement du service d'auto-test**
  - 4.1. Spécification du service
  - 4.2. Plan de validation du service
  - 4.3. Architecture logicielle du service
  - 4.4. Plan d'intégration du service
  - 4.5. Développement en parallèle
    - 4.5.1. Objectifs
    - 4.5.2. Composition de l'équipe et organisation
    - 4.5.3. Répartition des tâches pour un développement parallèle
    - 4.5.4. Planning prévisionnel (diagramme de Gantt)

- 4.5.5. Mode de suivi des activités de l'équipe
- 4.6. Par unité de programme constituant le service d'auto-test
  - 4.6.1. La spécification de l'unité de programme
  - 4.6.2. Une fiche de test pour la validation de l'unité  
Intitulé, descriptif, conditions d'activation (données en entrée), procédures de test et d'évaluation (mode)
  - 4.6.3. Bilan de validation : 0-défaut ou non (suite au *run*)
  - 4.6.4. En annexe (identifiée) : les sources de programmes, les fichiers d'entrée et de sortie (résultat du run), (le fichier d'évaluation), les stubs éventuels.
- Remarque :** Chaque programme devra contenir une cartouche permettant de l'identifier (fichier physique, auteur(s), rôle, date de création et de dernière modification, ...) et être commenté suivant le formalisme Javadoc. Chaque programme devra faire l'objet d'une revue de code par un autre collègue de l'équipe et devra tracer le relecteur dans sa documentation.
- 4.7. Bilan des tests d'intégration
- 4.8. Bilan de validation du service

## 5. Développement du service de vérification syntaxique

A vous de définir la rédaction de cette partie. Elle devra refléter les étapes du cycle de développement en V dont les bilans de validation nécessaires.

## 6. Bilan du projet

- 6.1. Analyse des plannings prévisionnel et effectif (problèmes rencontrés)
- 6.2. Retour d'expérience
  - Ce qui a été réussi
  - Ce qui peut être amélioré.
  - Les enseignements du projet

**Annexe A.** Planning prévisionnel et planning effectif (diagramme de Gantt/Pert)

**Annexe B.** Suivi d'activités (réunions et rapports)

**Annexe C.** Sources de programmes (commentés), tous fichiers impliqués dans les diverses validations.

**Deadline :** **i) 25/05/09 (0h00)** pour l'**archive électronique (zip) des sources du projet**

(tous les sources du produit logiciel, fiches de tests, JDT...),

**e-mail :** Marie-Jose.Caraty@ParisDescartes.fr,

**ii) 01/06/09** au plus tard **remise de la version papier** à l'IUT ou par courrier :

Marie-José Caraty – Université René Descartes – UFR de Mathématiques et Informatique – 45, rue des Saints Pères – 75006 Paris

# *Recette du composant de vérification syntaxique*

Le composant de vérification syntaxique devra faire l'objet d'une recette le 26/05/09

## 1. Organisation de la recette

Chaque équipe devra valider en ma présence le produit logiciel demandé  
Pour être recevable, le composant devra subir les tests croisés. Il devra être 0-défaut pour les JDT élaborés par les autres équipes ainsi que sur un JDT que je préparerai à votre égard.

## 2. Livrable à ajouter à votre rapport (deadline du 01/06/09) : le rapport de recevabilité du produit logiciel faisant suite à la recette.