

Fiche récapitulative F-Safe

Aurélien Deharbe Gary Benattar

Résumé

Complément des comptes rendus, pour les révisions de l'examen 1.

1 Roles

- Vendredi 25 Novembre
 - Chef de projet : Gary Benattar
 - Chef de projet adjoint : Thierry Paulin
 - Directeur technique : Aurelien Deharbe
 - Directeur technique adjoint : Yoann Couillec
- Lundi 28 Novembre (I_CR_1_2011-11-28.pdf)
 - Adminstrateur système : Jérémie Salvucci
 - Secretaire : Arnaud Sipasseuth
- Vendredi 9 décembre (I_CR_3_2011-12-09.pdf)
 - Chercheurs :
 - Arnaud Sipasseuth
 - Thierry Paulin
 - Jérémie Salvucci
 - Gary Benattar
 - Ugo Ditaranto
 - Joseph Afari
 - Codeurs :
 - Aurélien Deharbe
 - Yoann Couillec
 - Vincent Membré
 - Antoine Hamdi
 - Adele Djoudi
 - Sergey Vakulenko

2 Methodologie

- Sergey : Présentation sur les méthodes agiles (PR_methodes_agiles.pdf)
- Grandes lignes de la methode adoptée (I_CR_2_2011-12-02.pdf)
- Basée sur *Scrum* :
 - Echelle de temps : 1 jour scrum = 1 semaine *F-Safe*

- Daily meeting (1 fois par semaine)
- *Scrum* Planning
- Retrospective reviews
- Avec une touche d'*XP* :
 - Pair programming
 - Code review par le binome ainsi que par le directeur technique
 - Unit Test (à débattre)
- Documents partagés :
 - Taches avec estimation de temps
 - Diagramme de Gantt
 - Diagramme de Temps
 - Histogramme de répartition du travail
 - Camembert code vs biblio
 - ... (Cf F-Safe Planning - Google Doc)

3 Environnement technique

- Source control (I_CR_2_2011-12-02.pdf)
 - Choix retenu : Git / Github (wiki, branching, ...)
 - Présentation et formation par Jérémie
- Langage (I_CR_2_2011-12-02.pdf)
 - Premier choix entre paradigmes ; choix retenu : programmation fonctionnelle (adaptée à un compilateur)
 - Choix entre Scala et OCaml ; choix retenu : OCaml (“prudence”)
 - Etablissement de règles de codage (E_conventions_code.pdf)

4 Algorithmique de la terminaison

- Principalement basé sur l’algorithme de Foetus
- Etablissement d’un graphe d’appels et construction d’un ordre sur les arguments
- Etapes de l’algorithme :
 - Construction de l’AST
 - Construction du graphe d’appels
 - Détection de cycles dans le graphe d’appels (A_detection_de_cycles.pdf)
 - Complétion partielle du graphe (A_complétion_partielle.pdf)
- A approfondir : détection des ordres pour la construction des matrices

5 Code

- Grammaire (C_grammaire.pdf)
- Parsing des déclarations de type (90%)
- Parsing des expressions (90%)
- Découpage en tâches pour le prochain sprint *Scrum*