

Processus de développement du projet de programmation

ANALYSE

- 1) Ecrire une description de quelques lignes sur le projet, l'objectif final et les objectifs intermédiaires et les fonctionnalités principales. Proposer un exemple d'utilisation ou **scénario d'utilisation** pour chaque objectif intermédiaire ou final
→ créer le livrable « *Analyse de l'utilisation* »
- 2) Réaliser un organigramme représentant **les enchaînements des activités du programme**. Il couvre évidemment les scénarios identifiés dans 1) mais ne doit pas être trop détaillé : il faut que les principales actions (générales) apparaissent.
→ créer le livrable « *Organigramme général* »
- 3) À partir des scénarios identifiés dans 1), et de l'organigramme 2), identifier et lister les **fonctionnalités principales du programme**.
→ créer le livrable « *Liste des fonctionnalités* »
- 4) Identifier les **données manipulées et la manière de les structurer** : il pourra s'agir de variables isolées, de tableaux, de grilles à deux dimensions, de constantes....
→ créer le livrable « *Liste des types de données* »
- 5) À partir des étapes précédentes, identifier les différents **sous-programmes, ainsi que les paramètres** d'entrée et/ou de sortie, nécessaires à la réalisation **des fonctionnalités**. Vous donnerez l'entête ainsi qu'une phrase explicative de l'objectif de chaque sous-programme ainsi que la vue externe.
→ créer le livrable « *Liste des sous-programmes* »
- 6) En fonction des **priorités des fonctionnalités**, des **dépendances entre les sous-programmes** pour les réaliser, établir un échéancier de programmation de ces sous-programmes.
→ créer le livrable « *Échéancier* »

Une première version des 6 livrables doit être validée par un enseignant.

RÉALISATION

- 1) Pour chaque sous-programme selon l'échéancier : établir un jeu de test, écrire le sous-programme en notation algorithmique, puis en langage C++, tester chaque sous-programme.
- 2) Tester de manière globale en regardant si l'exécution suit les exemples d'utilisation de 1). Faire les fiches de tests unitaires au fur et à mesure du codage des Sous-Programmes
- 3) Réaliser les sous-programmes qui répondent aux fonctionnalités secondaires ou optionnelles selon la même démarche logique. Idéalement, cette partie ne doit pas amener à retoucher les entêtes des sous-programmes mis au point en 1) mais

seulement leur corps (méthode plus efficace, affichage amélioré...)

- 4) Compléter au fur et à mesure le cahier de conception avec l'analyse, les fiches de tests unitaires, etc...

CRITÈRES POUR LA NOTATION

- 1) Les différents livrables de l'analyse seront évalués par les enseignants de programmation (respect des consignes des livrables, qualité, etc.)
- 2) Respect de l'échéancier
- 3) Qualité de programmation : choix des noms explicites pour les sous-programmes, les variables, les types ; présence de commentaires ; respect des indentations ; structuration en sous programmes ; utilisation des structures de contrôle ; robustesse ; ergonomie ; fonctionnalités avancées...