Analyse - R3.03 *IUT Montpellier-Sète*
BUT2 - Semestre 3 (2022/2023) *Département Informatique*
- **M. Gasquet, G. Hisler, N. Lazaar, N. Palleja, S. Robillard, B. Rouchon**

VERSION ENSEIGNANT

TD0 : Génie Logiciel

Exercice 1 (généralités)

Question 1 • Quelle est la différence entre un programme, un ensemble de programmes et un logiciel ?

Cours

Question 2 • C'est quoi le génie logiciel ? est-il nécessaire pour développer une solution informatique ?

Cours

Question 3 • Nous avons vu en cours les activités principales dans un processus de développement logiciel.. Donnez les entrées/sorties ainsi qu'une description de chaque activité.

Quelques éléments en complément du support de cours :

— **Spécification** :
DESC : Comprendre les besoins du client, les objectifs généraux, l'environnement du futur système, ressources disponibles, contraintes de performance...
IN : Informations fournies par le client (Expert du domaine d'application, futur utilisateur)
OUT : Cahier des charges + manuel d'utilisation préliminaire

— **Conception** :
DESC : Élaborer une solution concrète réalisant la spécification. Description architecturale en composants (avec interface et fonctionnalités). Réalisation des fonctionnalités par les composants (algorithmes, organisation des données). Réalisation des exigences non-fonctionnelles (performance, sécurité...)
IN : Cahier des charges fonctionnel
OUT : Dossier de conception

— **Programmation** :
DESC : Implantation de la solution conçue. Choix de l'environnement de développement, du/des langage(s) de programmation, de normes de développement...
IN : Dossier de conception
OUT : Code documenté + manuel d'utilisation

— **Programmation** :
DESC : Implantation de la solution conçue. Choix de l'environnement de développement, du/des langage(s) de programmation, de normes de développement...
IN : Dossier de conception
OUT : Code documenté + manuel d'utilisation

— **Vérification & validation** :

DESC : Vérification : assurer que le logiciel satisfait sa spécification. Validation : assurer que les besoins du client sont satisfaits (au niveau de la spécification, du produit fini...).
IN : système
OUT : rapports de tests, certificats, preuves...

— **Déploiement** :
DESC : Installation, formation, assistance.
IN : Code exécutable installable, manuels d'installation et d'utilisation.
OUT : Utilisation du logiciel, feedback utilisateurs validation

— **Maintenance** :
DESC : Installation, formation, assistance.
IN : nouveaux besoins, bugs constatés, remarques générales, version actuelle
OUT : nouvelle version, manuel d'utilisation à jour

— **Documentation** :
DESC : une activité qui accompagne chaque étape

Question 4 • Donnez la liste des qualités requises dans un cahier de charges.

Quelques éléments en complément du support de cours :

— Bon niveau de généralité ;
— Formulation adéquate des besoins. Problème bien décrit ;
— Etre précis, non ambigu malgré l'usage d'un langage informel (ou semi-formel) ;
— Etre complet (pas d'omission involontaire) ;
— Etre cohérent (pas d'inférence de fonctionnalités) ;
— Etre vérifiable : critères de validation définis. Evaluer la faisabilité des besoins. faire éventuellement une maquette, une simulation ;
— Etre modifiable : facilité à exprimer un changement ou ajout de besoins.

Exercice 2(critères de qualité)

Question 1 • Donnez la liste des principaux facteurs de qualité d'un logiciel.

Question 2 • Quel est le seuil minimal (entre min, moyen et max) à avoir pour chaque facteur des systèmes suivants :

— Contrôleur de machine à laver
— Jeu vidéo
— Client de messagerie
— Application mobile d'une banque en ligne
— Simulateur en génétique

	A	B	C	D	E	F
machine à laver	max	max	min	min	min	min
Jeu	moy	max	moy	max	min	max
Mail	max	moy	moy	min	max	moy
BankApp	max	max	moy	min	max	max
SimGen	min	min	max	max	min	min

A : Portabilité, B : facilité d'utilisation, C : validité, D : performance, E : sécurité/fiabilité, F : maintenabilité

Exercice 3 (cycle de vie)

Question 1 • Reprenez le modèle en cascade vu en cours. Donnez les caractéristiques et les inconvénients d'un tel modèle.

Quelques éléments en complément du support de cours :

— Cycle de vie linéaire sans aucune évaluation entre le début du projet et la validation ;
— Le projet est découpé en phases successives dans le temps ;
— A chaque phase correspond une activité principale bien précise produisant un certain nombre de livrables ;
— Chaque phase ne peut remettre en cause que la phase précédente.

Inconvénients :

— Les vrais projets suivent rarement un développement séquentiel.
— Etablir tous les besoins au début d'un projet est difficile.
— Aucune validation intermédiaire (aucune préparation des phases de vérification) ;
— Sensibilité à l'arrivée de nouvelles exigences : refaire toutes les étapes ;

Question 2 • Reprenez maintenant le modèle en V et donnez les avantages et les inconvénients d'un tel modèle.

Quelques éléments en complément du support de cours :

— La préparation des dernières phases (validation-vérification) par les premières (construction du logiciel), permet d'éviter d'énoncer une propriété qu'il est impossible de vérifier objectivement après la réalisation.
— Idéal quand les besoins sont bien connus, quand l'analyse et la conception sont claires.

Inconvénients :

— Construit-on le bon logiciel ? le logiciel est utilisé très (trop) tard
— Il faut attendre longtemps pour savoir si on a construit le bon logiciel ;
— Difficile d'impliquer les utilisateurs lorsqu'un logiciel utilisable n'est disponible qu'à la dernière phase.

Question 3 • Donnez une comparaison détaillée entre un prototypage jetable et un prototypage évolutif.

Quelques éléments en complément du support de cours :

Jetable :

— Cette approche consiste à produire rapidement une maquette (prototype jetable) qui constitue une ébauche du futur système.
— Avec cette approche, on est capable de définir plus explicitement et de manière plus cohérente les besoins des usagers, de détecter les fonctions manquantes et identifier et améliorer les fonctions complexes, comme elle permet de démontrer la faisabilité et l'utilité de l'application.
— Le prototype est ensuite abandonné et le système réel est construit.

évolutif :

— Un prototype répond à des objectifs différents : on cherche à construire un système éventuellement très incomplet, mais dans son dimensionnement réel de façon à pouvoir faire des essais en vraie grandeur. On détermine d'abord un ensemble minimum de fonctions elles-mêmes incomplètes de façon à réaliser un premier incrément du logiciel dont on se sert pour analyser le comportement du logiciel. L'utilisateur fournit en retour des informations au concepteur qui modifie le prototype. Ce processus d'évolution de prototypes continue jusqu'à ce que l'utilisateur soit satisfait du système livré.