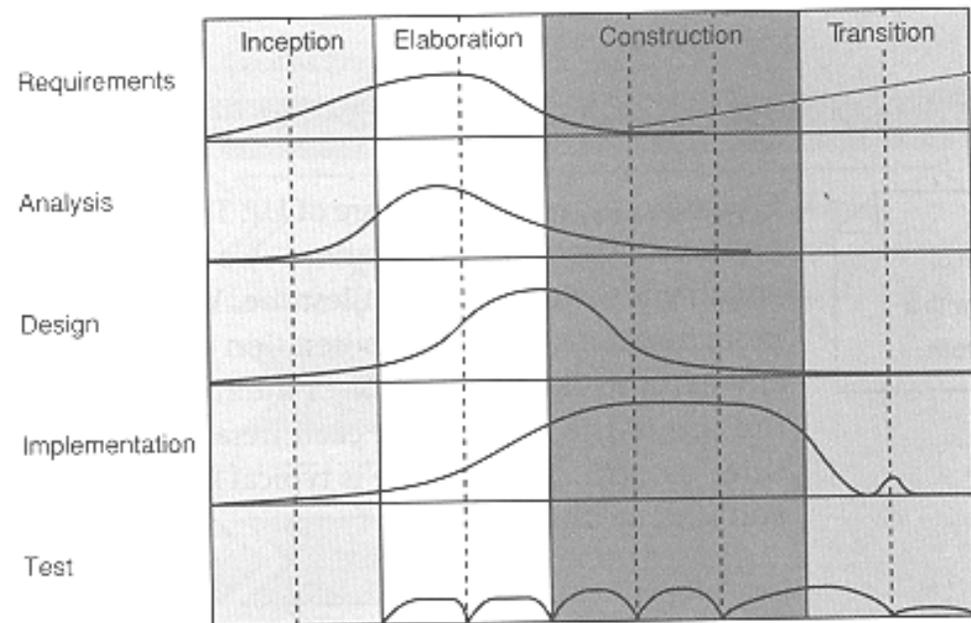


Montréal, le 7 mars 2008

Réponses de l'étudiante 100/100 aux questions
de l'examen intra de INF 5151 groupe 50

NOTE Il ne s'agit pas des seules réponses possibles. FIN DE LA NOTE

1. (30 %) Votre ami François est un garçon très gentil mais le génie logiciel (GL) est loin d'être son domaine préféré. En particulier il n'a rien compris à la représentation sous forme de matrice des phases de UP. Expliquez-lui clairement la figure suivante et mettez en évidence les différences avec un cycle de vie en cascade (« classique »). Vous supposerez que, malgré ses faiblesses en GL, il connaît très bien le cycle de vie « classique ». (Je vous conseille de ne pas dédier plus de 30 minutes à cette question)



R 1.

Avant tout j'aimerais préciser que l'acronyme UP est l'abréviation de *Unified Process*. UP est le processus du génie logiciel (GL) défini par les créateurs de la notation UML.

UP est un processus itératif et incrémentiel fondé sur l'architecture et piloté par les exigences et les risques. Itératif signifie que le projet est divisé en plusieurs petits projets d'une durée de l'ordre de quelques semaines : chaque sous-projet est une

itération et « livre » à l'interne ou à l'extérieur un « produit partiel ». La différence entre deux livraisons est un incrément.

J'imagine, cher François, que tu commences déjà à voir les différences avec le cycle de vie classique (CdVC), mais avant de te parler des différences, je vais t'expliquer la figure.

1. Chaque courbe représente l'évolution dans le temps de la quantité de travail. Le temps s'écoule de gauche vers la droite.

2. Les quatre colonnes représentent les quatre phases et les cinq lignes représentent les cinq flux de travail essentiels.

Une phase est un intervalle de temps caractérisé par des objectifs, organisé autour de certaines activités et ayant des jalons.

Objectifs des phases :

Inception : montrer la faisabilité et mettre en place tout ce qui est nécessaire pour le projet

Élaboration : créer une architecture exécutable, définir les exigences de qualité et faire un plan détaillé pour la phase suivante.

Construction : compléter l'analyse et la conception et avoir l'architecture finale.

Transition : corriger les erreurs et aider les utilisateurs

Activités centrales des phases :

Inception : Exigences et analyse avec des débuts de conception.

Élaboration : pratiquement tous les flux de travail avec des légers décalages

Construction : surtout la mise en œuvre.

Transition : surtout les tests d'acceptation.

Je ne vais pas présenter les jalons mais il est important de comprendre que contrairement au CdVC qui est fondé sur l'échéancier et les livraisons des artefacts, UP est fondé sur une approche pilotée par les objectifs. L'artefact est un moyen pour satisfaire un objectif et non vice versa !

La dernière ligne de la figure (test) montre les différents incréments.

Avant de parler des différences j'aimerais que te dire que les cinq flux de travail de UP sont l'équivalent des « phases » du CdVC (comme quoi le mot « phase » change de signification).

Différences :

1. À tout moment (dans toute phase) on peut travailler dans tous les flux de travail. Donc, par exemple, on ne finit pas une analyse avant de commencer la conception comme dans le CdVC. C'est la quantité de travail relative qui change. On est dans une espèce de spirale.

2. Les incréments et les itérations rendent le processus plus agile et le risque que le produit final ne soit pas accepté plus faible. Le CdVC (au moins dans sa version la plus simple) n'a pas d'itération et s'il en a c'est pour corriger des erreurs.

3) Contrairement au CdVC l'échéancier n'est pas au centre du projet mais il est un moyen pour maîtriser le projet.

2. (20 %) Françoise ne comprend pas l'utilité et le but des champs *Der*, *Ea*, *Sta*, des fiches des exigences. Expliquez-lui clairement :

1. La fonction des « fiches » dans les artefacts du GL ;
2. Le but et l'utilité de *Der*, *Ea* et *Sta*.

R2

Fiches

Les fiches sont un moyen pour faciliter la structuration des informations et pour éviter d'oublier des éléments importants. En particulier les fiches des problèmes, des besoins et des exigences sont reliées entre elles, ce qui facilite la traçabilité. Les fiches (stockées dans une BD) permettent aussi de présenter les informations organisées en fonctions des besoins des parties prenantes. Un gestionnaire de projet, par exemple, sera à un certain moment intéressé par toutes des exigences ayant un certain degré de stabilité.

Der

Ce champ contient la liste des exigences dérivées. Les exigences dérivées étant des composants de l'exigence principale. Ceci permet la structuration des exigences complexes dans une hiérarchie. Ce champ peut être vide

Ea

Ce champ contient de liens avec d'autres éléments d'analyse : diagrammes conceptuels, autres exigences, prototypes, etc. Ce champ favorise la compréhension de l'exigence décrite dans la fiche.

Sta

Ce champ contient une valeur numérique qui représente le niveau de stabilité de l'exigence. Utile pour la gestion de projet car, par exemple, il est préférable de mettre en œuvre les exigences très stables (pour lesquelles les parties prenantes n'ont jamais changé d'avis) que celles très instables (changements d'avis continuels)

3. (10 %) François et Françoise, deux membres de l'équipe KOBOLD, ne réussissent pas à se mettre d'accord sur les antécédents de la fiche d'exigences titrée « demande de révision de note ». François propose : « *Insatisfaction de la part de l'étudiant suite à une modification de sa note finale* » tandis que Françoise propose : « *L'étudiante a reçu le formulaire de modification de note après la signature de l'enseignante.* ». Vous êtes l'arbitre. Quelle formulation, selon vous, est la plus correcte ? Pourquoi ? Est-ce que vous en avez une meilleure ? Si oui, écrivez-là.

R3

Je trouve la formulation de Françoise plus correcte que celle de François. Celle de François contient un élément « psychologique » pratiquement impossible à évaluer.

Je n'en ai pas de meilleure.

4. (10 %) François ne comprend pas pourquoi il ne faut pas employer des synonymes dans un document décrivant les exigences. Expliquez-lui clairement pourquoi.

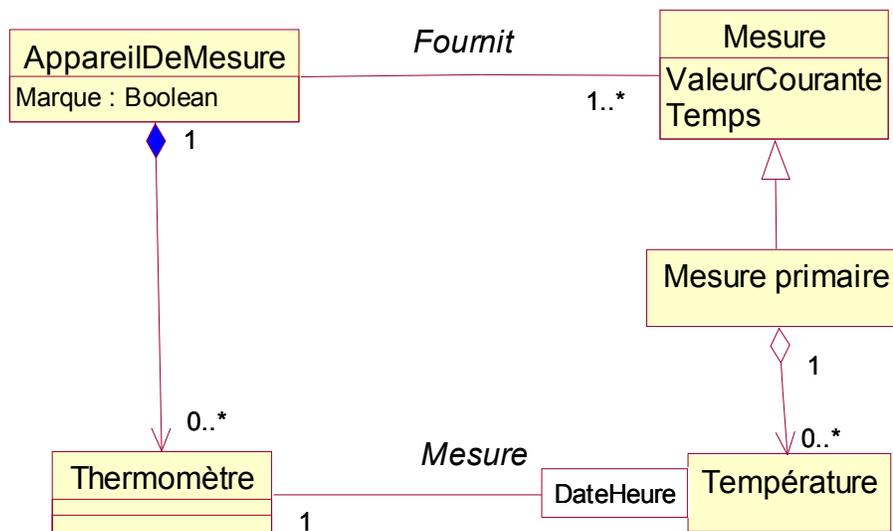
R4

Rappelons qu'un mot est synonyme d'un autre quand les deux ont des formes différentes mais le même sens (ou un sens proche). La synonymie, très appréciée en littérature, est dangereuse dans les documents techniques car elle « force » le lecteur à essayer de comprendre si le rédacteur a voulu dire la même chose avec les deux mots. Dans un document technique, le premier mouvement quand on voit des noms différents est de penser que leur signification est différente : que le rédacteur veut mettre en évidence un détail. Deux synonymes du langage courant se transforment souvent en deux concepts différents dans des documents techniques où la précision est très importante (souvent une association de type

spécialisation) : c'est-à-dire qu'ils cessent d'être des synonymes car ils ont au moins un attribut ou le type d'un attribut qui les différencie.

5. (30 %) François — encore lui ! — pour vous montrer ses progrès en UML a préparé le diagramme suivant comme une traduction d'une partie de la conceptualisation du module 11 (celui où l'on parle de *Température* dans le but de montrer comment « on cherche les concepts »).

1. « Traduisez-le » en français,
2. Si vous pensez qu'il y a des erreurs corrigez-les en les expliquant.
3. Dessinez un diagramme qui, selon vous, est correct dans le contexte du problème énoncé dans le module 11.



R5

R5.1 : traduction

Un *AppareilDeMesure* **fournit** une ou plusieurs *Mesure*. Un *AppareilDeMesure* est doté d'une *Marque* qui peut être Vraie ou Fausse. Une *Mesure* a comme attributs une *ValeurCourante* et un *Temps*.

Un *AppareilDeMesure* est une *composition* de 0 ou plusieurs *Thermomètre* et un *Thermomètre* **fait partie** de un et un seul *AppareilDeMesure*.

Une *Mesureprimaire* est de type *Mesure*.

Une *Mesureprimaire* est un *agrégat* de 0 ou plusieurs *Température*. Une *Température* **fait partie** d'une et une seule *Mesureprimaire*.

Un *Thermomètre* mesure la *Température*. Le qualificateur *DateHeure* permet d'avoir un seul *Thermomètre* qui mesure une *Température*.

R5.2 : Erreurs.

Après chaque erreur je vais mettre entre parenthèses un niveau de gravité (5 = max et 1 = min).

1. Un *AppareilDeMesure* n'est pas une composition de *Thermomètre*. (5) Un *thermomètre* est de type *AppareilDeMesure*.
2. L'attribut *Marque* ne doit pas être un Booléen (3). Il s'agit plutôt d'une chaîne de caractères,
3. *Température* n'est pas une partie de *Mesureprimaire* mais une spécialisation. (5)
4. Le qualificateur doit être sur l'autre bout de l'association pour indiquer qu'à un moment donné il y a une seule valeur de température. (4)

R5.3

Voici mon schéma

