Projet Programmation JEE

ESIGELEC

Problématique :

Les logiciels sont souvent générateurs

d'erreurs, de bugs dus à des spécifications

insuffisantes et/ou des applications peu/mal testées....

Voici quelques exemples ...

Bourse de Tokyo (01/11/05)

Bug qui a empêché l'ouverture à 9h00 GMT de la bourse en bloquant toutes les cotations. (…)

Il s'agirait du plus important chaos boursier qu'ait connu la bourse depuis août 1997.

(…)

Le marché financier de Tokyo représente un volume quotidien moyen de transactions estimé à 13,2 milliards de dollars.

CAUSE : Trop gros volume de transactions à traiter

source http://www.generation-nt.com

Terminaux des gares (04/12/04)

Panne informatique qui a paralysé vendredi, à la veille du week-end, 800 terminaux de vente aux guichets des gares, qui ne pouvaient émettre de billets.

CAUSE : Algorithme défectueux qui a progressivement contaminé les terminaux de vente en gare

source : http://www.mines.inpl-nancy.fr/~tisseran/cours/qualite-logiciel/qualite_logiciel.html

Mission Vénus

Passage à 5 000 000 de Km de la planète, au lieu de 5 000 Km prévus

CAUSE : Une virgule a été remplacée par un point (au format US des nombres)

Le bogue de l'an 2000

La lutte contre le bogue de l'an 2000 a coûté à la France 500 milliards de francs.

CAUSE : L'année était codée sur deux caractères, pour gagner de la place

Ariane V (04/06/96)

Premier lancement de Ariane V, explosion en vol

Coût du programme d'étude d'Ariane V : 38 milliards de Francs.

CAUSE : Logiciel de plate forme inertielle repris tel quel d'Ariane IV sans nouvelle validation.

Ariane V ayant des moteurs plus puissants s'incline plus rapidement que Ariane IV, pour récupérer l'accélération due à la rotation de la Terre.

Les capteurs ont bien détecté cette inclinaison d'Ariane V, mais le logiciel l'a jugée non conforme au plan de tir (d'Ariane IV), et a provoqué l'ordre d'auto destruction.

En fait tout se passait bien !

(source : http://www.mines.inpl-nancy.fr/~tisseran/cours/qualite-logiciel/qualite_logiciel.html)

Sécurité Sociale (2001)

Un bug informatique aurait coûté 10 millions d'euros à la Sécurité Sociale, selon le Parisien. De nombreuses cliniques ont été remboursées deux fois.

CAUSE : bug dans le nouveau logiciel de télétransmissions des demandes de remboursement.

«la plupart des cliniques ont maintenant accepté de rembourser». Seule une dizaine d'entre elles contestent le remboursement et ont porté l'affaire devant les tribunaux des affaires sociales.

(source : http://www.liberation.fr)

FDJeux (27/02/07)

Un bug informatique joue un vilain tour aux joueurs de La Française des Jeux
Environ 25 000 parieurs qui ont joué en ligne au Joker Plus vont être remboursés. Le numéro 9 ne sortait jamais dans les combinaisons.

Le chiffre 9 n'apparaissait jamais parmi les 7 numéros attribués automatiquement par un système Flash aux joueurs en ligne.

Les 25 000 joueurs lésés séduits par le Joker+ sur Internet seront dédommagés.

CAUSE : oubli du chiffre 9 dans la matrice de tirage aléatoire du logiciel

(source : http://www.01net.com)

BNP Paribas (26 février 2009)

Des milliers de clients, particuliers et entreprises, ont été débités plusieurs fois d'un même chèque, virement ou prélèvement.

Ce bug concerne près de 586.000 opérations.

CAUSE : BUG informatique

(source : http://www.01net.com)

BlackBerry (10 octobre 2011)

Paralysie du réseau mondial du téléphone BlackBerry

Messagerie texte inactive, messagerie instantanée bloquée, navigateur internet en rade, environ 70 millions d'utilisateurs sont touchés à travers le monde. Le tout en pleine campagne de lancement de son concurrent, l'I-Phone 4S…

CAUSE : problème d'engorgement

(source : Yahoo Actualités)

Facebook (21 octobre 2013)

Facebook victime d'un bug mondial : des centaines de millions d'usagers paralysés

Impossible de "mettre à jour son statut" (autrement dit publier un message, une photo ou partager un article), de commenter ou de cliquer "J'aime"

environ 1 milliard d'utilisateurs potentiellement touchés

CAUSE : inconnue

bug de l'An 2038 (19/01/2038)

Sur beaucoup de systèmes d'exploitation actuels :

temps = nb de secondes depuis le 01/01/1970 et codé sur 32 bits signé

maxi = 2 147 483 647 secondes

La limite sera atteinte le 19/01/2038 à 3h14mn7s

on passera alors au 13 décembre 1901 ...

Qualité

Nécessité d'améliorer la qualité logicielle

Mais Comment ?

Qualité

Définition :

Degré auquel un système, un composant ou un processus satisfait les besoins spécifiés

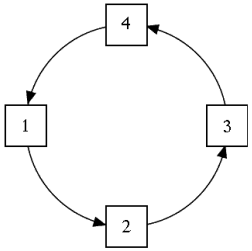
Degré auquel un système, un composant ou un processus satisfait les attentes du client ou les besoins des utilisateurs

Afin d'améliorer au maximum la qualité logicielle, il est nécessaire d'insister sur les phases de spécification/conception et de tests.

Assurance qualité et cycle de vie sont fortement liés

Le génie logiciel désigne l'ensemble des méthodes, des techniques et outils concourant à la production d'un logiciel

Cycle de vie

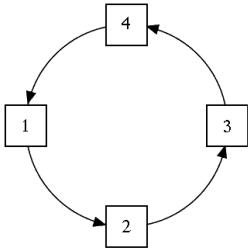


Un projet logiciel passe par plusieurs étapes :

- Expression du besoin
- Conception
- Réalisation
- Tests
- Validation
- Mise en production
- (mort éventuelle)

Pourquoi utiliser un cycle de développement ?

- permet de définir un cadre de référence pour le processus logiciel
- définit des liens entre les étapes de conception et de tests
- permet de ne pas oublier les phases de tests et de maintenance en les intégrant dans le cycle
- permet d'améliorer la Qualité du logiciel



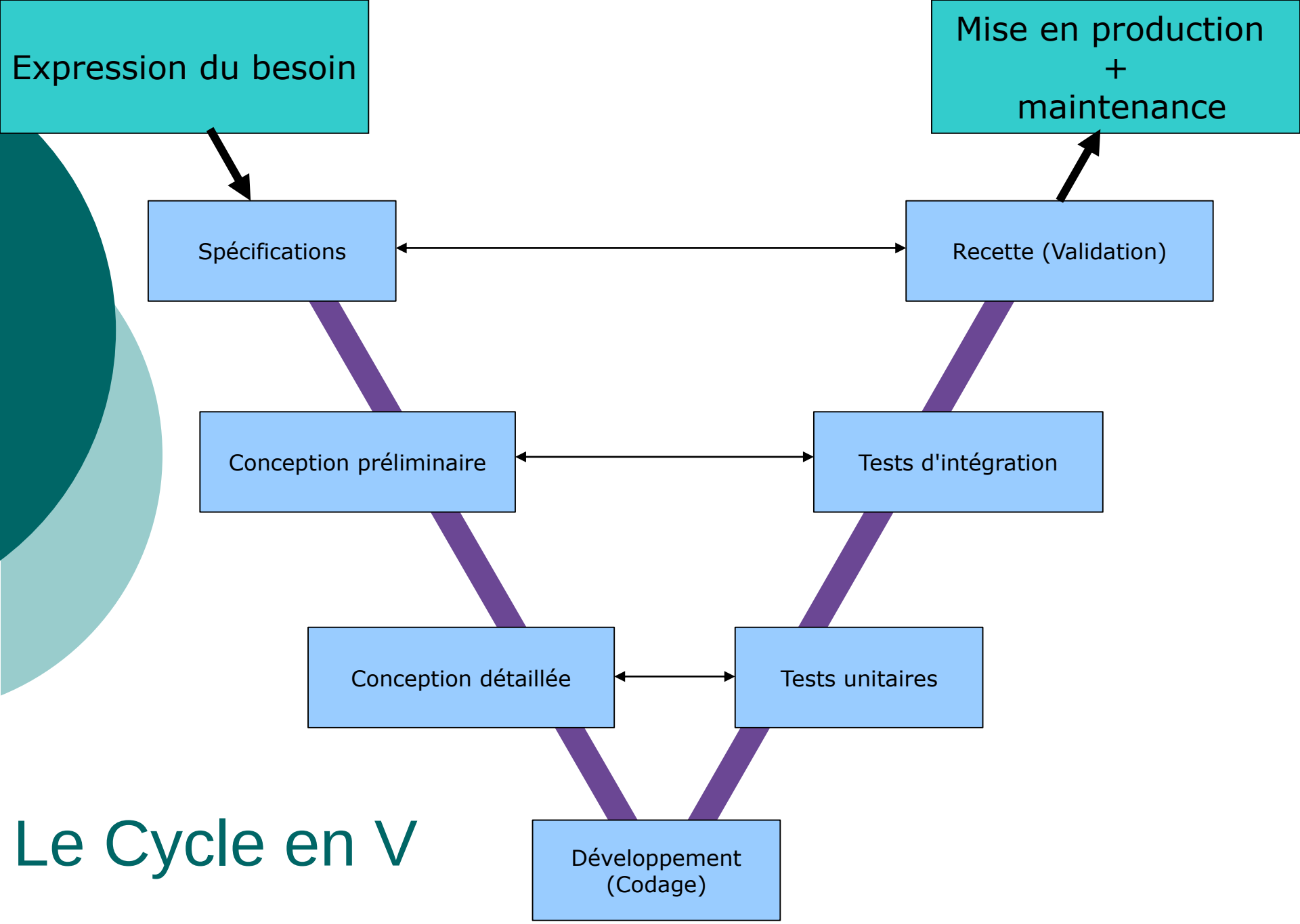
Les cycles de vie

Cycle en V
Cycle en cascade
Cycle incrémental
Modèle du prototypage rapide
Cycle en spirale
...

Le cycle en V

- Le plus connu des cycles de développement
- Modèle industriel bien éprouvé
- Validation intégrée et planifiée au cycle de développement
- Présence de points de contrôle (insiste sur les tests)
- Activités et documents précisément définis
- Se prête bien à l'Assurance Qualité et au Management de Projets

Le Cycle en V


Expression du besoin
Mise en production + maintenance
Spécifications
Recette (Validation)
Conception préliminaire
Tests d'intégration
Conception détaillée
Tests unitaires
Développement (Codage)

Les documents

Afin de faciliter la transition et la communication entre les différentes phases/personnes, il est nécessaire de rédiger des documents

Spécifications

La phase de Spécifications aboutit à l'écriture :

- Du dossier de spécification du logiciel (DSL)
- Du plan de validation du logiciel (PVL) qui permettra de faire la validation

Spécifications

Le DSL formalise la liste des fonctionnalités que le logiciel devra remplir pour répondre aux besoins des clients :

Contexte :

- Objectifs du projet
- Hypothèses

Description générale :

- Fonctionnalités principales
- Futurs utilisateurs
- Définition des IHM
- Contraintes générales
- Temps de réponse
- Contraintes mémoire
- ...

Spécifications

PVL (Plan de Validation du Logiciel) :

-> document qui permet de définir le processus de validation finale du logiciel (scénarii de tests,…)

Exemple de scénario de test

Action	Résultats attendus	OK	KO	Commentaires
Double cliquer sur le fichier appli.exe	Le programme se lance			
Cliquer sur le bouton "gérer les utilisateurs"	La fenêtre de gestion des utilisateurs s'ouvre			
…….				
Cliquer sur le menu "quitter"	Le programme se termine			

Conception préliminaire

La phase de Conception préliminaire aboutit à l'écriture :

- Du dossier de conception préliminaire (DCP)

permet de définir la liste des « modules » de haut niveau du logiciel et leurs interactions (entrées/sorties, protocoles,...)

- Du plan d'intégration du logiciel (PIL) qui permettra de tester l'intégration

Conception détaillée

La phase de Conception détaillée aboutit à l'écriture :

- Du dossier de conception détaillée (DCD)

Etape qui consiste à détailler chacun des modules définis lors de la conception préliminaire

- Du dossier de tests unitaires du logiciel (DTUL) qui permettra de faire les tests unitaires

Développement (codage)

La phase de Développement

Traduction des algorithmes écrits lors de la phase de conception détaillée dans le langage de développement choisi (C, C++, Java, C#, JavaScript, PHP, Python...)

Pourquoi définir des règles ?

Faciliter la maintenance

- le développeur n'est pas forcément celui qui assurera la maintenance et les évolutions futures.

Permettre une meilleure réutilisation du code

- cela permettra d'intégrer certaines fonctionnalités dans un autre projet

Les règles de codage

- commenter le code
- choix des noms de fichiers
- choix des noms des fonctions
- choix des noms de variables
- découpage en modules
- variables globales autorisées ou non
- …

Les tests unitaires

Qu'est-ce qu'un test ?

« Le test est l'exécution ou l'évaluation d'un système ou d'un composant, par des moyens automatiques ou manuels, pour vérifier qu'il répond à ses spécifications ou identifier les différences entre les résultats attendus et les résultats obtenus »

(IEEE- IEEE STD 729, 1983 : Standard glossary of software engeneering terminology)

Les tests unitaires

Tester unitairement les fonctions du logiciel

Les fonctions sont alors validées une par une.

Si le test unitaire échoue :

- → revenir à l'étape de conception détaillée
- → re-modifier le code
- → re-tester la "fonction"

Choix des valeurs de test

Pour tester, choisir des valeurs judicieuses :

- les valeurs limites
- les valeurs non conformes
- les valeurs valides

Objectif des tests

Trouver les erreurs

L'objectif n'est pas de prouver qu'il n'y en a pas !

Tests unitaires réalisés par le codeur

Les tests se terminent lorsqu'il n'y a plus d'erreurs décelées

La phase d'Intégration et de Tests

Intégrer, assembler tous les modules (fonctions) et tester si leur assemblage fonctionne

Tests Non réalisés par le codeur

Les tests s'arrêtent quand il n'y a plus d'erreurs détectées

Si les tests d'intégration échouent :

→revenir à l'étape de conception préliminaire

→Faire une étude d'impact

→recommencer le cycle en V pour le périmètre impacté par ce changement

La phase d'Intégration et de Tests

Une erreur en phase de conception préliminaire engendre un surcoût plus important qu'une erreur en phase de conception détaillée

La phase de Validation

Permet de "valider" l'application complète (fonctionnement, interfaces, etc…) en s'appuyant sur les documents rédigés lors de la phase de spécification (DSL et PVL)

Réalisée par une personne qui n'a pas développé de préférence

S'arrête quand tous les cas d'utilisation ont été validés

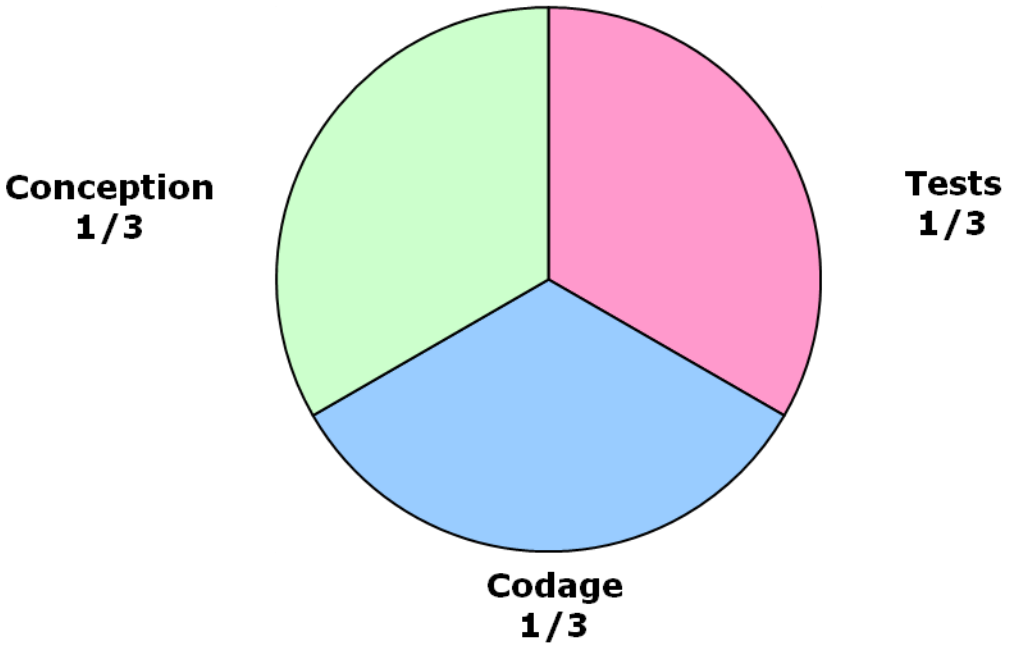
La phase de Validation

La validation se réalise le plus souvent en deux étapes :

Qualification : validation en interne (étape préalable à la recette)

Recette : tests, remontée et levée des éventuelles réserves, puis validation finale

On peut considérer que la répartition approximative des coûts selon les phases est la suivante :



Inconvénients du cycle en V

Difficile de prendre en compte des évolutions du cahier des charges

Visibilité tardive des résultats

Peu ou pas de maquettage et/ou de prototypage

Parfois difficile à appliquer rigoureusement

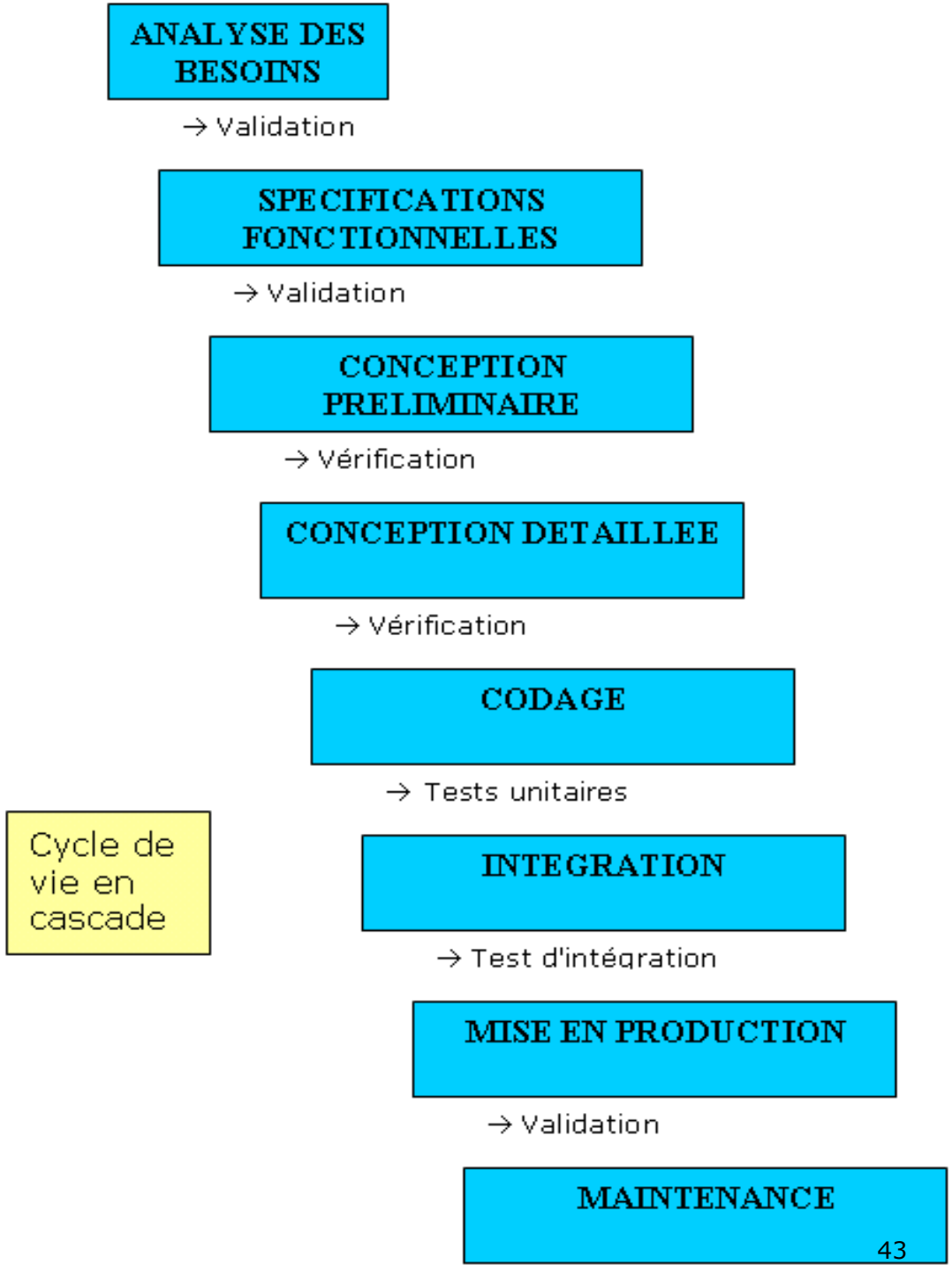
Cycle de vie assez long en général

Le cycle en cascade

S'appuie sur l'hypothèse que l'avenir est prévisible avec certitude

On ne passe à l'étape suivante qu'après vérification de l'étape courante

Le cycle en cascade



Le cycle en cascade

Inconvénient :

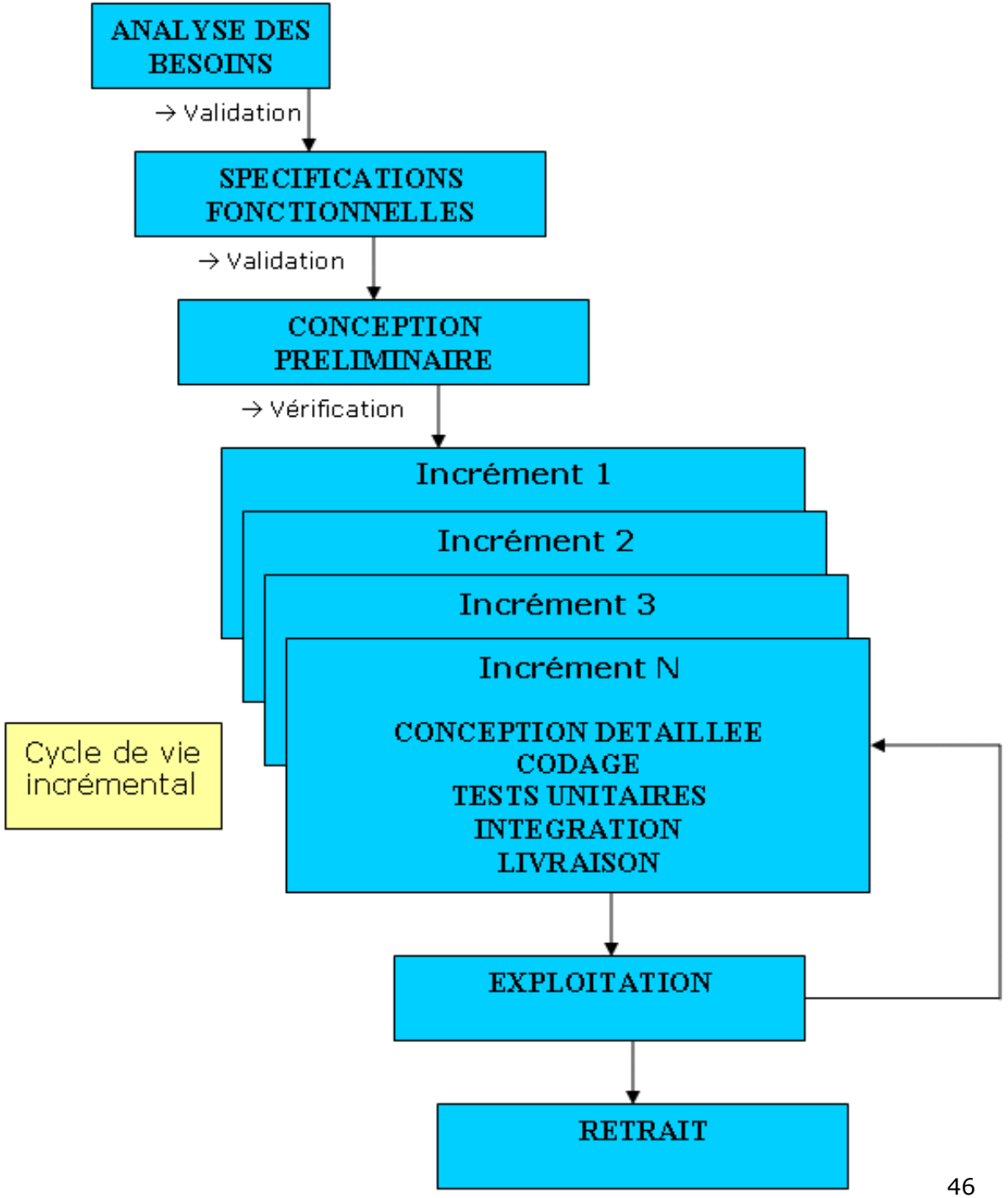
Une modification en début de cycle engendre de forts coûts en fin de cycle

Le modèle incrémental

Idée: construire le logiciel étape par étape

- Spécifié et conçu dans son ensemble
- Réalisation par incréments de fonctionnalités
- A chaque étape le produit est testé, exploité et maintenu dans son ensemble
- Ce cycle de vie permet de faire accepter progressivement un logiciel par les utilisateurs plutôt que de faire un changement brutal des habitudes

Le modèle incrémental



Le modèle incrémental

Avantages :

- chaque développement est moins complexe
- les intégrations sont progressives
- possibilité de livraisons et de mises en service après chaque incrément

Le modèle incrémental

Inconvénients

- Difficultés en cas de remise en cause du noyau ou des incréments précédents
- Risques de ne pas pouvoir intégrer de nouveaux incréments
- Les incréments doivent être aussi indépendants que possibles
- La durée des itérations dépend de l'implication du client → risque de dépassement des délais

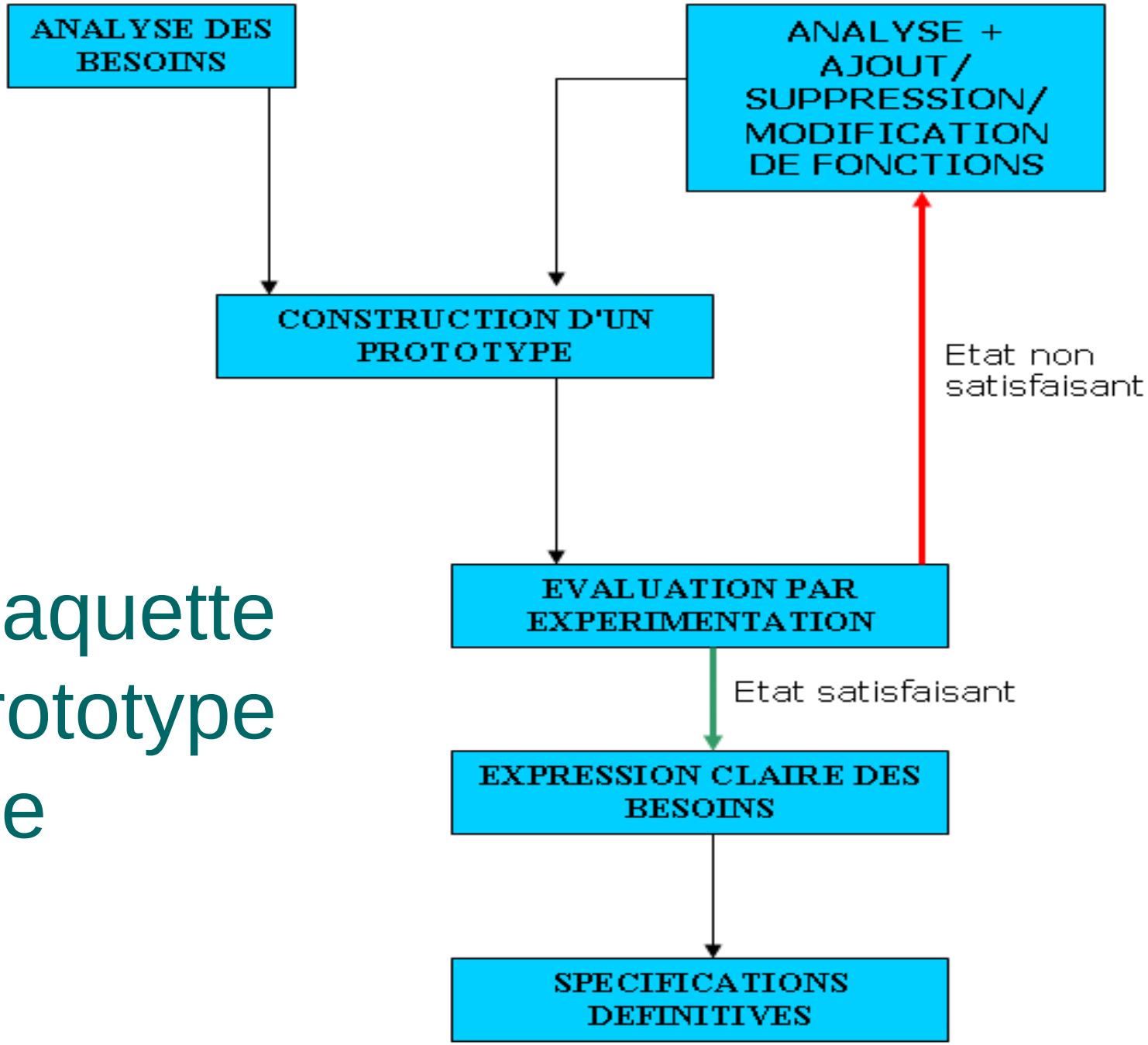
La maquette ou prototype rapide

La maquette ou prototype rapide est utilisée en amont du cycle de développement :

- Analyse des besoins
- Spécifications fonctionnelles

Permet d'affiner l'expression des besoins par expérimentation en construisant un prototype

La maquette ou prototype rapide



La maquette ou prototype rapide

Permet au client et au développeur de bien se mettre d'accord sur la nature du produit à réaliser (interfaces, fonctionnalités,...)

Permet de raccourcir la durée en économisant des allers/retours client/développeur pendant la phase d'analyse des besoins

Les méthodes agiles

SCRUM est probablement la méthode Agile la plus utilisée.

SCRUM est basée sur les notions de sprint, daily scrum, planning poker, sprint review…

Adaptée à des équipes de petite taille (moins de 8)

Cette méthode sera pratiquée rigoureusement dans le projet technologique

Conclusion

Faire le bon choix du cycle de vie dès le départ

Cycle en V : bonne idée mais difficile à respecter

Cycles incrémentaux : permettent un changement en douceur pour le client

Les maquettes/prototypes rapides permettent d'affiner au mieux les besoins du client/utilisateurs → améliorer la Qualité

Déroulement du projet

Pour se rapprocher au maximum de la vie en entreprise

- Travail en équipe (binôme)
- Constitution aléatoire des binômes
- 1 projet = 3 équipes
- Sujets tirés au hasard

Déroulement du projet

1 projet = 3 binômes successifs

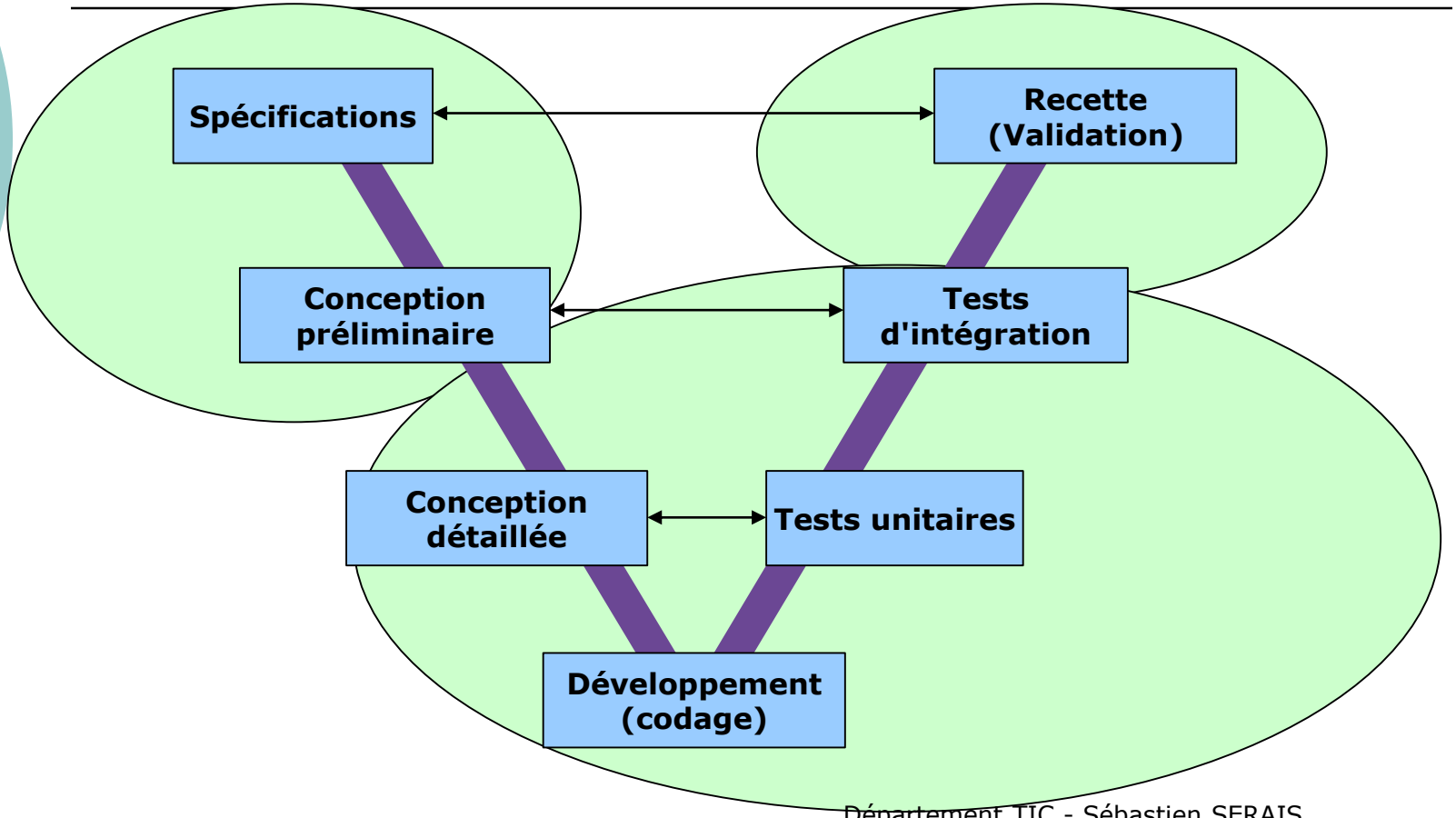
1er binôme : phases de spécification et de conception préliminaire

2è binôme : phases de conception détaillée, de codage et de tests unitaires, intégration

3è binôme : validation du logiciel

→ documents à rendre à la fin de chacune des 3 phases

Découpage


Spécifications
Recette (Validation)
Conception préliminaire
Tests d'intégration
Conception détaillée
Tests unitaires
Développement (codage)

binôme 1
binôme 2
binôme 3

Découpage

Séance	1	2	3-6	7	8
Durée	4h	4h	4*4h	1*4h	2h
Salle	TP	TP	TP	TP	TP
Phase	Présentation + Groupes + Spécifications	Spécifications + Conception préliminaire	Conception détaillée + codage + tests unitaires +Tests d'intégration	Soutenances + corrections bug	Recette + Validation
Equipe		1	2	2	3
Documents à rendre		DSL + PVL + DCP (format papier + ent)		DCD +Tableaux de bord + code commenté + .sql (ent)	PVL complété + Bilan (ent)

Documents à réaliser par équipe 1

- DSL (environ 5-6 pages):
  - Objectifs du projet
  - Fonctionnalités principales
  - Futurs utilisateurs (Use Case)
  - Données à stocker
  - Définition des IHM (Maquettes)
  - Contraintes générales
  - Temps de réponse
  - Contraintes mémoire
  - Fonctionnalités détaillées
  - Délais
  - Coûts
- PVL

Documents à réaliser par équipe 1

- DCP :
  - Schéma de l'Architecture générale du projet

Documents à réaliser par équipe 2

- DCD
  - diagramme de classes Serveur
- Tableaux de bord
  - Format Excel
  - 1 mise à jour à la fin de chaque séance=1 nouvel onglet Excel

Exemple de tableau de bord

(unité : heures.homme)

Date : 21/03/2011						
Tâches	Qui	Budget temps (fixé au départ)	Consommé	RAF (Reste à faire)	Produit (Budget-RAF)	Temps non facturé= Delta (consommé-produit)
Modéliser la BDD	Mickaël	2	1	1	1	0
Créer fichier .sql	Nicolas	3	2	0	3	-1
MessageDAO	Mickaël	2	2	3	-1	3
TOTAL		7	5	4	3	2

Documents à réaliser par équipe 3

- Rendre le PVL initial complété (format papier ou numérique)
- Compte rendu des tests
  - % des objectifs atteints ?
  - respect des normes (W3C) ?
  - respect des IHM ?
  - votre avis sur le logiciel (simplicité d'utilisation, IHM,...)
  - commentaires dans le code? Javadoc générée ?
  - code cohérent avec le diagramme de classes ?
  - selon vous, l'application peut-elle être mise en production en l'état ? Pourquoi ?
  - combien de temps faudrait-il pour finaliser l'application ? quel en serait le surcoût ?
  - autres remarques si nécessaire

Documents à réaliser par équipe 3

- Bilan général sur les 3 phases de projet du binôme (1 page)
  - Quelles ont été vos erreurs ?
  - Que faut-il retenir de cette expérience ?

Le projet

- Durée : 30h
- Coût du projet ?

Questions ?