

# Projet d'étude et de recherche

Jean-Michel Couvreur

Janvier 2013

## Les projets TER ?

- Directement inspirés des Projets de programmation, Master 1, Université Bordeaux
- Un projet TER est un exercice en continu sur un semestre :
  - Réaliser/Augmenter un logiciel.
  - Transmettre, expliquer et justifier votre travail.

## Buts des projets TER

- **Application des connaissances acquises :**
  - **Pratiques** : compilation, systèmes d'exploitation, réseaux, etc.
  - **Théoriques** : complexité, calculabilité, mathématiques numériques, etc.
- **Acquisition de nouvelles connaissances ou perfectionnement** dans des domaines variés.
- **Application d'une pratique scientifique :**
  - Documentation, bibliographie, analyse de l'existant.
  - Rigueur dans les affirmations, justifications des choix, tests, utilisation d'outils de vérification.

## Buts des projets TER (suite)

- Expériences concrètes :
  - **Génie logiciel** : développement en groupe autour d'un programme de taille relativement importante.
  - **Rédaction** de documents écrits.
  - **Présentation orale** (soutenance).

## Synopsis du déroulement d'un projet

- Documentation, compréhension du sujet.
- Analyse du problème, conception de prototypes papiers, spécifications, analyse des besoins, tests.
- Rédaction du mémoire intermédiaire.
- Architecture, conception.
- Gros oeuvre de l'implémentation du logiciel.
- Rédaction du mémoire final.
- Soutenance.

## Cours et TDs

- **TDs** Techniques de communication.
- **Cours** magistraux sur :
  - Génie logiciel.
  - Recherche documentaire.
  - Principes de développement, modularité.
  - Rédaction de documents et présentation orale.

## Calendrier

- **Phase I** : 28 janvier - 4 mars (mémoire intermédiaire et architecture).
- Congés d'hiver : 23 février - 3 mars.
- **Phase II** : 4 mars - 17 mai (codage, rédaction des documents).
- Congés de printemps : 20 avril - 27 avril
- **Phase III** : 21 mai (soutenances)

## Quoi réaliser ?

- Mémoire intermédiaire du projet.
- Maquette/Analyse de l'architecture.
- Implémentation du logiciel.
- Mémoire final du projet.
- Supports de soutenance.

Détaillons-les . . .



## Contenu du mémoire intermédiaire

- 1 Un **résumé** du projet.
- 2 Une **introduction** au domaine.
- 3 Une **analyse** de l'existant.
- 4 Une liste et analyse des **besoins non fonctionnels** (+ risques, probl. techniques, tests, essais).
- 5 Une liste et analyse des **besoins fonctionnels** (+ prototype papier, probl. techniques, tests, essais).
- 6 Une description de **prototypes et des résultats de tests préparatoires**.
- 7 Un **planning**, affectations des tâches.
- 8 Une **bibliographie**.

## Mémoire intermédiaire : partie documentaire

(Résumé, introduction, analyse de l'existant, et biblio.)

- Démontre la connaissance du sujet du projet à partir de :
  - Recherches bibliographiques.
  - Lecture d'articles
  - Utilisation du réseau et des outils de la BU sciences.
  - Client du projet, autres enseignants, chercheurs du domaine.
- Décrit et analyse le ou les codes existants (et possibilités de réutilisations et/ou modifications).

## Mémoire intermédiaire : cahier des charges

(Définition des besoins, planning)

- Définit et **organise le travail à effectuer**.
- Assure une **entente préalable** entre le client et le groupe qui réalise le projet.
- Enonce les **besoins non fonctionnels**, puis **fonctionnels**.

## Besoins non fonctionnels

- **Qualités globales** que l'on attend du logiciel.
- Par exemple : domaine d'action ; temps de réponse ; fiabilité ; facilité d'utilisation, convivialité, esthétique des interfaces ; utilisation d'algorithmes particuliers ; langages d'implémentation ; portabilité.
- **Argumentés** avec :
  - Une **analyse de risques** et **leurs parades**.
  - Une description des **tests de validation** prévus, des **seuils quantifiés** à dépasser ou à ne pas dépasser.
  - Une description de **problèmes techniques associés** (avec, le cas échéant, des essais pratiques).

# Besoins fonctionnels

- **Fonctionnalités** nécessaires au logiciel.
- **Argumentés** avec :
  - Un **prototype papier** (par exemple, dessin de l'aspect de l'interface).
  - Une description des **tests de validation** prévus sur certaines des fonctionnalités.

## Mémoire intermédiaire : partie prototypes, essais et tests

- La description des **prototypes, essais et tests** :
  - **Tests effectifs de certains points techniques** liés à des outils, des systèmes, des machines, etc.
  - **Tests effectifs de faisabilité, de complexité**, etc.
  - Description d'**essais implémentés** et de **prototypes** qui permettent d'affiner la compréhension du domaine.

## Apropos de la rédaction des documents

- Tous les documents devront être rédigés en  $\text{\LaTeX}$ .
- **Désavantages :**
  - Apprentissage plus long.
  - WYSINWYG.
- **Avantages :**
  - Gestion des **références bibliographiques** facilitée.
  - Permet de se concentrer sur **le fond** sans la forme.
  - **Bien adapté** au type de documents à réaliser

## Le contenu de la maquette de l'architecture

- L' **architecture du logiciel sous forme graphique** et qui décrit l'agencement de ses principaux composants :
  - **paquetages**,
  - **modules** et/ou **classes** et/ou **types**.
- **Pas de représentation particulière requise**, car dépend du projet et du langage (par ex. UML pour Java et C++, graphe de liens pour C, automates pour OCaml, etc.)



## Logiciel : ses qualités requises

- Bon choix du (des) **langage(s)**.
- **Réutilisation** de logiciels existants.
- Utilisation des **outils de développement** adéquats : (makefiles, automake, gdb, lex, yacc, antlr, rcs, cvs, subversion, javadoc, doxygen, bibliothèques graphiques, bibliothèques numériques, gestion mémoire, etc.).
- Bon choix des **structures de données**.
- Bon choix des **algorithmes**.
- **Cloisonnement** : abstraction modulaire, masquage d'implémentation, séparation interface/implémentation, composants réutilisables.
- **Lisibilité**, commentaires, pas de duplications.
- **Robustesse**, traitement des erreurs (assertions/exceptions).

## Logiciel : son analyse

- Utilisation des **outils de d'analyse** adéquats de **performance** et gestion **mémoire** (si nécessaire) : gprof, gcov, lint, valgrind, etc.
- Etude de la **complexité en temps et en espace** des algorithmes principaux.
- **Description et résultats des tests** :
  - **Tests de validation** (cf. besoins non fonctionnels et fonctionnels).
  - **Tests de fonctionnement et de robustesse** (par exemple : entrées aléatoires, imprévues, trop grandes, trop petites, trop comme-ci et trop comme-ça)

## Contenu du mémoire final

- 1 Les **six premières rubriques du mémoire intermédiaire** (améliorées si nécessaire).
- 2 Des **exemples de fonctionnement** du logiciel final.
- 3 L'**architecture**, le **découpage modulaire**.
- 4 Une description, commentaires techniques, justifications sur **les algorithmes et les structures de données utilisés**.
- 5 Une **analyse en complexité** des algorithmes.
- 6 Les **résultats des tests** de validation et de fonctionnement, expliqués et justifiés.
- 7 La description technique des **extensions** possibles du logiciels.
- 8 Une **bibliographie**.
- 9 Le cas échéant : un **appendice** en forme de manuel

## La soutenance

- S'adresse à un jury et à tous les étudiants du Master.
- **Durée** : 20 minutes + 5/10 minutes de questions.
- **Règle stricte** : tous les membres d'un groupe doivent prendre la parole pendant 5 minutes.
- **Moyens** : vidéo-projection.

## Contenu de la soutenance

- 1 Description du domaine.
- 2 Description générale du logiciel.
- 3 Architecture du logiciel.
- 4 Description de quelques points techniques.
- 5 Exemples de fonctionnement.
- 6 Résultats des tests.

## le client

Le **client** (celui qui a proposé le sujet) :

- Connaît le domaine d'application du projet.
- Oriente l'élaboration du logiciel.
- Vous reçoit régulièrement.
- Vérifie la qualité des rapports (le respect du plan)
- Juge le travail et le résultat final.
- Attribue une note à la maquette de l'architecture et au projet global.

## Le rapporteur

Le **rapporteur du mémoire final** :

- Est anonyme jusqu'à la soutenance (c'est un client d'un autre projet).
- Attribue une note aux deux mémoires (intermédiaire et final).

# Les jurys

- Le **jury de soutenance** :
  - Juge la soutenance.
  - Est composé d'enseignants de TER et de clients qui ont assisté à la soutenance.
  - Attribue une note à la soutenance seule.
- Le **jury de TER** :
  - Juge le travail complet.
  - Est composé de tous enseignants de TER et des clients.
  - Attribue la note finale à partir de toutes les notes partielles.



## La note finale

- **Mémoire intermédiaire** (rapporteur) + maquette de l'architecture (client) : 25 %
- **Mémoire final** (rapporteur) : 40 %
- **Satisfaction du client** : 15 %
- **Soutenance** (jury) : 20 %

## Dates principales

- remise du **mémoire intermédiaire** : 4 mars
- remise de la **maquette de l'architecture** : 4 mars
- remise du **mémoire final** et **code** : 17 mai
- **soutenances** : 24 mai ( à confirmer).