

« Il semble que l'on n'ait jamais assez de temps pour faire correctement le travail dès la première fois, mais que l'on trouve toujours le temps de le refaire »

« Posséder un marteau ne fait pas de vous un architecte »

Table des matières

A Épreuve E6.2 : projet technique	3
A.1 Objectifs	3
A.2 Travail demandé	3
A.3 Soutenance	4
B L'expression du besoin	4
C Cahier des charges	4
D Processus de développement logiciel	5
D.1 Définition	5
D.2 OpenUP	5
D.3 Principe	6
D.4 Glossaire	6
E Exigences	7
E.1 Le produit à réaliser	7
E.2 Le développement	7
E.2.1 La documentation à produire	8
E.2.2 La gestion des modifications	8
E.2.3 La livraison	9
E.3 Les ressources	10
E.3.1 Les ressources logicielles	10
E.3.2 Les ressources matérielles	10
E.3.3 Les ressources documentaires	10
E.4 Dossier de suivi de projet	10

F	Revue de projet	10
F.1	Ressources	11
F.2	Revue n°1	11
F.3	Revue n°2	11
F.4	Revue n°3	12
F.5	Revue finale	13
G	Production attendue	14
H	Tâches à réaliser	15

Lien : <http://tvaira.free.fr/>

A Épreuve E6.2 : projet technique

A.1 Objectifs

Le projet technique a notamment pour objectifs de placer les étudiants en situation :

- de résoudre un problème technique en respectant une démarche cohérente et conforme aux pratiques rencontrées dans les entreprises :
 - démarche de projet ;
 - environnement collaboratif, travail d'équipe ;
 - contexte spécifié : contraintes et moyens d'entreprise, contraintes réglementaires et normatives, démarche qualité, environnement.
- de mobiliser et d'acquérir des connaissances scientifiques, programmatiques et techniques, notamment méthodologiques.

Lors du démarrage du projet, au début du mois de février de la deuxième année, le dossier de projet est remis à l'équipe d'étudiants :

- La première partie fixe les contraintes générales du projet pour l'ensemble de l'équipe ;
- les parties suivantes s'adressent plus spécifiquement aux différents membres composant l'équipe et précisent les Tâches professionnelles à réaliser de façon individuelle contribuant à la réussite du projet.

L'épreuve a pour objectif de valider l'acquisition des compétences terminales :

- C2.1 Maintenir les informations
- C2.2 Formaliser l'expression d'un besoin
- C2.3 Organiser et/ou respecter la planification d'un projet
- C2.4 Assumer le rôle total ou partiel de chef de projet
- C2.5 Travailler en équipe
- C3.1 Analyser un cahier des charges
- C3.3 Définir l'architecture globale prototype ou d'un système
- C3.5 Contribuer à la définition des éléments de recette au regard des contraintes du cahier des charges
- C3.6 Recenser les solutions existantes répondant au cahier des charges
- C4.1 Câbler et/ou intégrer un matériel
- C4.2 Adapter et/ou configurer un matériel
- C4.3 Installer et configurer une chaîne de développement
- C4.4 Développer un module logiciel
- C4.5 Tester et valider un module logiciel
- C4.6 Intégrer un module logiciel
- C4.7 Documenter une réalisation matérielle/logicielle

A.2 Travail demandé

Le travail demandé consiste à la conception voire à une évolution d'une partie d'un produit ou d'un service informatique en liaison avec une problématique authentique liée à un cahier des charges.

Les éléments de description et de modélisation s'appuient sur les outils de modélisation SysML et/ou UML.

Les étudiants travaillent par groupes de 3 à 4. Ils devront réaliser le projet en se répartissant les problématiques décomposées au préalable par l'équipe pédagogique. Ils ont la possibilité de proposer une organisation modifiée par rapport à celle proposée initialement.

On demandera aux étudiants d'assembler les différentes structures matérielles et logicielles qu'ils auront réalisées et d'effectuer les essais de la maquette ou du prototype assemblé, dans une situation permettant de tester, à minima, toutes les fonctionnalités demandées dans le cahier des charges.

La durée minimale d'un projet pour un étudiant IR est de **200 heures**.

A.3 Soutenance

La soutenance dure **1 heure**. Le jury est composé de professeurs de spécialité, d'un professeur de SPC (Sciences Physiques) et éventuellement d'un industriel.

La soutenance est organisée en trois phases. Une phase de présentation de 20 minutes, durant laquelle le candidat expose à la fois la globalité du projet et son action personnelle dans le projet puis une phase de mise en œuvre du projet de 20 minutes et une phase d'interrogation de 20 minutes durant laquelle le jury questionne le candidat.

Lors de l'épreuve de soutenance, le jury doit disposer :

- du sujet initial remis à l'équipe, le dossier contrat de projet comportant les éléments de validation, des avenants éventuels ;
- du dossier technique de projet constitué par les étudiants ;
- des éléments de suivi constitués au moins de la fiche de synthèses concernant les trois revues de projet.

Le dossier technique de projet est établi en un seul exemplaire papier et un exemplaire numérique pour les membres du jury.

Pour arrêter la note finale du candidat à cette sous-épreuve, la commission d'interrogation prend en compte les deux notes proposées lors des revues de projet et la note de soutenance de projet :

- 3 points de coefficient pour la moyenne des notes attribuées par les professeurs de la section, lors des revues de projet ;
- 3 points de coefficient pour la note attribuée par la commission d'interrogation, à l'issue de la soutenance du projet.

B L'expression du besoin

Il faut dire les choses en quelques phrases courtes et précises. L'expression des besoins (EdB) doit en dire le plus possible avec le moins de mots possibles. Des schémas et des illustrations peuvent apporter plus de précision tout en restant concis et synthétique. L'important d'une bonne EdB est qu'elle soit comprise par tout le monde, aussi bien les utilisateurs que les techniciens.

L'expression des besoins est le **point de départ de tout projet informatique**. Elle se situe avant l'étude de faisabilité et du cahier des charges.

Les cas d'utilisation (UML) permettent d'illustrer ces besoins. Ils détectent puis décrivent les besoins fonctionnels (du point de vue de l'utilisateur), et leur ensemble constitue le modèle de cas d'utilisation qui dicte les fonctionnalités complètes du système.

C Cahier des charges

Le but d'un projet est de **satisfaire le besoin**. Il faut exprimer clairement les objectifs à atteindre d'un projet, afin d'éviter toute confusion entre vous et le demandeur.

Le CDCF (Cahier Des Charges Fonctionnel) est un document qui permet de formaliser avec précision le besoin du demandeur. En effet, le CDCF est un tableau de bord qui définit le projet et détaille les conditions dans lesquelles il doit être réalisé ; il décrit l'ensemble des caractéristiques attendues des fonctions de service. C'est le lien de compréhension entre l'entreprise et le client.

Les besoins devront être exprimés sous forme de fonctions (d'où le nom d'analyse fonctionnelle) et non de solutions pour permettre un choix lors de l'étude technique. Ces fonctions doivent donc être listées, classées et explicitées.

Le cahier des charges constitue un élément contractuel.

Chaque projet s'appuiera sur un cahier des charges fonctionnel. Il précisera le contexte du projet avec les contraintes associées, la nature des Tâches professionnelles à réaliser, leur chronologie et leur enchaînement, les ressources et les moyens à disposition.

Lors du démarrage du projet, au début du mois de février de la deuxième année, le dossier de projet est remis à l'équipe d'étudiants :

- La première partie fixe les contraintes générales du projet pour l'ensemble de l'équipe ;
- les parties suivantes s'adressent plus spécifiquement aux différents membres composant l'équipe et précisent les Tâches professionnelles à réaliser de façon individuelle contribuant à la réussite du projet.

D Processus de développement logiciel

D.1 Définition

Un processus de développement décrit une **méthode** qui permet de construire, déployer et éventuellement maintenir un logiciel. Un processus de développement définit une séquence d'étapes, partiellement ordonnées, qui permettent d'obtenir un système logiciel ou faire évoluer un système existant.

« Le processus unifié (UP) est un processus de développement logiciel, c'est-à-dire qu'il regroupe les activités à mener pour transformer les besoins d'un utilisateur en système logiciel » (Jacobson, Booch, Rumbaugh 1999)

On utilisera la méthodologie **OpenUP** qui permettra de réaliser un développement de manière itérative et incrémentale.

D.2 OpenUP

OpenUP (*Open Unified Process*) fait partie du projet *Eclipse Process Framework* (EPF) et il est basé sur RUP (*Rational Unified Process*) et UP (*Unified Process*).

OpenUP préserve les caractéristiques essentielles d'UP et RUP, qui inclut le **développement itératif et incrémental**, le **développement piloté par les cas d'utilisation et les scénarios**, la gestion du risque et l'approche centrée sur l'architecture. Les parties les plus optionnelles du RUP ont été exclues et plusieurs éléments ont été fusionnés. Le résultat est un processus plus simple respectant malgré tout les principes du RUP. **OpenUP** cible les petites équipes constituées de 3 à 6 personnes et les petits projets durant 3 à 6 mois d'effort de développement.

Il y a environ 35 **tâches** et 30 **artefacts** répartis dans 7 catégories (Architecture, Déploiement, Développement, Environnement, Gestion de projet, Exigences, Test).

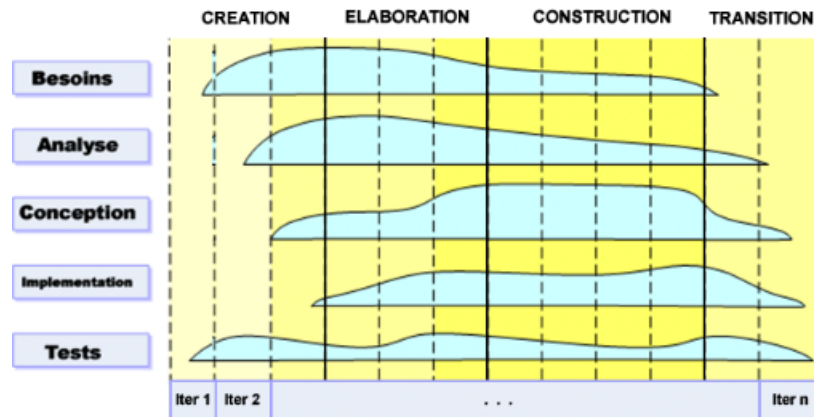
Liens :

- Site officiel : epf.eclipse.org/wikis/openup/
- Livre blanc : xavier.mehaut.free.fr/publications/OpenUp2007.pdf
- *Unified Process* : sabricole.developpez.com/uml/tutoriel/unifiedProcess/
- Gestion de projet : tvaira.free.fr

D.3 Principe

Le **développement itératif** s'organise en une série de développements très courts de durée fixe nommée **itérations**. Le résultat de chaque itération est un système partiel exécutable, testé et intégré (mais incomplet). Le résultat d'une itération n'est pas un prototype expérimental ou « jetable ».

Chaque itération comprend ses propres activités : analyse des besoins, conception, implémentation et tests.



D.4 Glossaire

Artefact : simplement le terme général qui désigne tout chose produite ou consommée par une étape du processus de développement. Cela peut désigner un diagramme UML, du texte, du code, ... etc .

Build : Une version opérationnelle d'un système ou d'une partie d'un système qui démontre un sous-ensemble des capacités à fournir dans le produit final.

Besoins : conditions auxquelles un système et plus généralement un projet doivent satisfaire.

Analyse : accéder à la compréhension des besoins et des exigences du client (« Quoi faire ? »). Il s'agit de livrer des spécifications pour permettre de choisir ensuite la conception de la solution. Un modèle d'analyse livre une spécification complète des besoins issus des cas d'utilisation et les structure sous une forme qui facilite la compréhension (scénarios), la préparation (définition de l'architecture), la modification et la maintenance du futur système.

Conception : élaboration d'une solution répondant aux besoins (« Comment faire ? ») plutôt que la mise en oeuvre de cette solution. La conception orientée objet fait apparaître des objets « informatique ». La conception constitue un point de départ à l'implémentation.

Implémentation : résultat de la conception pour implémenter le système (« Le faire ! ») sous formes de composants, c'est-à-dire, de code source, de scripts, de binaires, d'exécutables et d'autres éléments du même type. On peut aussi parler de fabrication ou de construction concrète du système.

Test : les tests permettent de détecter des erreurs dans l'implémentation. Pour mener à bien ces tests, il faut les définir pour chaque itération, les implémenter en créant des cas de tests, effectuer ces tests et prendre en compte le résultat de chacun.

Itératif : une itération est un cycle de développement complet

Incrémental : chaque développement s'ajoute et enrichit (écrase) l'existant. Un incrément est donc une avancée dans le développement.

Glossaire : artefact qui définit les termes importants utilisés par le projet. Cela clarifie le vocabulaire utilisé sur le projet.

E Exigences

E.1 Le produit à réaliser

Le produit à réaliser devra répondre aux facteurs de qualité suivants :

<i>Facteurs liés à l'environnement d'exploitation et d'utilisation</i>	
Facteur	Signification
Couplage	capacité de liaison avec un autre logiciel
Efficacité	optimisation de l'utilisation des ressources
Maniabilité	facilité d'emploi pour l'utilisateur
Robustesse	conservation d'un fonctionnement conforme aux besoins exprimés, en présence d'événements non prévus ou non souhaités (arrêt normal, intempestif ou d'urgence)
Sécurité	disponibilité assurant la continuité des traitements

<i>Facteurs liés à l'environnement de maintenance et de suivi</i>	
Facteur	Signification
Adaptabilité	facilité de suppression, d'évolution de fonctionnalités existantes ou d'ajout de nouvelles fonctionnalités
Maintenabilité	facilité de localisation et de correction des erreurs résiduelles
Portabilité	minimisation des répercussions d'un changement d'environnement logiciel et matériel

E.2 Le développement

En ce qui concerne les exigences qualité du développement :

- le développement se fera de manière itérative et incrémentale (méthodologie **OpenUP**) ;
- la modélisation du système devra être réalisée avec le langage de modélisation **UML 2** (logiciel **bouml**) ;
- l'architecture du logiciel sera « orientée objet » et multitâche si besoin ;
- les langages de programmation à utiliser sont le **C++**, **Java**, **PHP** ;
- les IHMs seront développées avec l'API **Qt** de *Qt Development Frameworks*, filiale de *Digia*.
- l'implémentation des structures de données doit privilégier les structures de données de la **STL** ou équivalent dans **Qt/Java**.
- le codage doit respecter le standard de codage en cours dans la section ;
- la chaîne de production des exécutable doit être réalisée avec un gestionnaire de type **make** ;
- le gestionnaire de gestion de versions utilisé sera **subversion** ;
- le logiciel devrait posséder une suite de tests unitaires **cppunit/Junit/PHPUnit** ;
- la documentation du code sera générée à partir de **doxygen** ;
- la documentation du projet (**README**, **TODO**, **Changelog** et **BUGS**) respectera le langage de balisage **Markdown** et sera générée avec l'outil **pandoc** (formats **ODT**, **PDF** ou **HTML** acceptés).

Le dépôt **subversion** sera organisé de la façon suivante :

- un répertoire **trunk** pour le développement courant,
- un répertoire **tags** pour le stockage des versions identifiées,
- un répertoire **branches** pour l'expérimentations, les tests et la correction des bugs.



Le développement en cours est enregistré dans un répertoire appelé **trunk**. Les revues portent sur la dernière version qui est copiée dans un sous-répertoire nommé **revue-j**, où **j** est le numéro de la revue, du répertoire appelé **tags**.

Il est recommandé de ne pas mettre dans le dépôt :

- les fichiers personnels (.user, ...)
- les fichiers générés (.o ou .obj, moc_* ou ui_*, ...)
- les fichiers de sauvegarde (~, .bak, ...)
- les fichiers binaires (.exe, ...)



Cas des fichiers ressources (images, ...) : vous pouvez utiliser le dépôt comme espace de stockage mais il est limité en taille (50MO). Bien évidemment, ces ressources ne sont pas des fichiers qui seront gérés par le gestionnaire de versions.

E.2.1 La documentation à produire

La suite bureautique libre LibreOffice ou Apache OpenOffice sera utilisée pour le dossier technique et les diaporamas.

Chaque page du dossier technique devra être clairement identifiée par :

- un entête comprenant : le nom de l'établissement, la session du BTS et le nom du projet.
- un pied de page comprenant : l'identifiant du document, la date de mise à jour, le nom du ou des auteurs (les initiales sont acceptées), le numéro de page du nombre total de pages.

La documentation du projet sera rédigée en respectant le langage de balisage Markdown. On pourra ensuite la générer avec l'outil pandoc dans les formats ODT, PDF ou HTML. Elle comprendra les fichiers textes suivants :

- **README** : contient des informations sur le projet (nom, objectifs, date de début), l'équipe de développement (nom, prénom et courriel de chaque membre), la version de l'application, la description détaillée des fonctionnalités (ce que fait l'application actuellement), la licence applicable et la procédure d'installation et les instructions d'exploitation ;
- **TODO** : (liste des tâches à réaliser) contient l'énumération de ce que l'équipe se fixe comme tâches à réaliser. Les éléments présents sont biffés une fois réalisés et se retrouveront dans le journal des améliorations et modifications apportées au programme **ChangeLog** ;
- **Changelog** : (journal des améliorations et modifications) contient l'énumération de ce que l'équipe a effectué comme travail sur le projet ;
- **BUGS** : contient la liste des défauts constatés non corrigés.



Ces fichiers textes seront stockés sur le serveur subversion.

Pour la documentation du code, on distinguera :

- les commentaires publics destinés à doxygen (de type `/** */`) ;
- des commentaires internes au code (de type `/* */`).



Le code remis en fin de projet devra être documenté de manière homogène. La notion de versions successives ne doit pas apparaître. Tous les modules logiciels livrés devront apparaître en "version 1". Une "version 0.9" est tolérée. Si votre code évolue entre la remise du dossier technique et l'oral, on passera alors en "version 1.1".

E.2.2 La gestion des modifications

Les modifications liées à des erreurs dans la demande du client (cahier des charges) ou des changements de stratégies liés à l'environnement ou les conditions financières ou des choix technologiques, devront faire l'objet d'une procédure de demande de modification clairement définie et approuvée par le responsable du projet et du client.

La procédure de gestion des modifications se déroule en trois temps :

1. établissement d'une demande de modification (remplir la "Fiche de demande de modification" fournie)
2. prise en compte de la modification (validée par un professeur responsable du projet)
3. réalisation de la modification

Un **avenant** sera alors joint au dossier technique.

E.2.3 La livraison

À l'issue du projet, l'équipe remettra à Thierry Vaira un dossier technique unique représentatif de l'ensemble du projet. Ce dossier comprendra une partie commune à tous les membres de l'équipe et la partie personnelle traitée par chacun d'entre eux. À des fins d'évaluation lors de la soutenance du projet, chaque étudiant devra indiquer explicitement les parties qu'il a personnellement réalisées dans les différents documents.



Un glossaire sera intégré au dossier technique. Des documents annexes peuvent être joints sous forme électronique (annexes techniques, programmes complets, manuel d'utilisation, notice de maintenance, sources complets, ...).

Les produits livrables du projet sont :

Nom	Contenu	Réalisation
Dossier de spécifications	Présentation du projet Dossier de spécification des besoins fonctionnels et techniques Découpage du travail au sein du groupe	Thierry Vaira
Dossier d'étude préliminaire	Analyse métier de l'application Choix ou justification de l'architecture logicielle et matérielle Modélisation métier des besoins	Thierry Vaira et groupe
Dossier de recette	La recette de validation du système	Thierry Vaira et groupe
Dossier d'étude et de réalisation détaillée	Choix ou justification des matériels et des logiciels utilisés Choix de l'organisation générale des tâches : choix des tâches, communications, synchronisations ... Modélisation statique et dynamique de l'application Algorithme de haut niveau décrivant les actions dans leurs grandes lignes sans entrer dans les détails du langage	Individuel
Dossier de tests unitaires	Ensemble des fiches de tests	Individuel
Manuel d'installation	Manuel permettant d'installer l'application	Groupe
Manuel d'utilisation	Manuel pour guider le futur utilisateur	Groupe

Le dossier technique du projet réalisé par un groupe d'étudiants sera constitué comme suit :

- Partie commune : (de 20 à 30 pages)
 - introduction, situation du projet dans son contexte ;
 - dossier de spécifications ;
 - dossier d'étude préliminaire et plan de tests des performances au regard du cahier des charges.
 - éléments nécessaires à la recette de la maquette ou du prototype final ;
 - résultats des essais de la maquette ou du prototype final ;
 - conclusion par rapport au cahier des charges fourni par le donneur d'ordre : test intégration, procédure et résultats de la recette.
- Partie personnelle : (de 20 à 30 pages)
 - situation de la partie personnelle dans l'ensemble du projet ;
 - dossier d'étude et de réalisation détaillée, essais unitaires.

Le dossier technique de projet sera remis sous forme électronique au format PDF.

E.3 Les ressources

E.3.1 Les ressources logicielles

Distinguer les logiciels nécessaires pour l'exploitation et le développement en précisant si ils existent, ou si il faut les acquérir : ateliers logiciels, logiciel de gestion de projets, bibliothèques de composants ...

Une ressource logicielle doit être identifié par son nom et sa version.

E.3.2 Les ressources matérielles

La liste des matériels nécessaires à l'exploitation et au développement en précisant si ils existent, ou si il faut les acquérir : moyens informatiques, outils de diagnostics, outils de tests, analyseurs de réseaux, appareils de mesures ...

E.3.3 Les ressources documentaires

La liste des documents mis à disposition des étudiants notamment la documentation technique, les normes et les notices d'utilisation concernant les moyens informatiques, les ateliers logiciels, les outils de diagnostic, les outils de tests.

Lien : tvaira.free.fr

E.4 Dossier de suivi de projet

Au fur et à mesure du déroulement du projet, il est indispensable que les étudiants consignent les éléments qu'ils réalisent au sein d'**un dossier de suivi de projet**. Ce dossier personnel a plusieurs utilités :

- formaliser l'avancement du travail de l'étudiant (notes, organigrammes, notes de calcul, résultats d'essais, mesure, simulation, modes opératoires, éléments de procédure ;
- compiler les ressources utilisées (notices techniques, document constructeur,) ;
- préparer les revues de projets qui seront au nombre de trois sur la durée du projet ;
- consigner les éléments qui serviront à préparer le dossier technique de projet.

Chaque membre de l'équipe de projet consigne, dans le dossier de suivi, les Tâches professionnelles qu'il réalise pour une période donnée. Les documents relatifs à la vie du projet (devis, bons de commandes, etc.) sont joints au dossier. L'utilisation d'un logiciel de gestion de projet aidera les étudiants dans cette démarche.

Un espace sur le serveur de la section est dédié au dossier de suivi de projet.

F Revues de projet

Après le lancement du projet (à +20 heures), à mi-projet (entre + 50 et + 60 heures), et durant la phase finale du projet (à +100 heures), un bilan doit mettre en évidence :

- ce qui a été réalisé ;
- ce qui reste à réaliser ;
- les ajustements éventuels, techniques ou relatifs au planning.

Les revues de projet permettent de constater le niveau d'implication et l'avancement du projet :

- La **première revue de projet** a pour objectif de vérifier la compréhension du travail demandé et la mise en œuvre du travail par les différents membres de l'équipe. Elle permet d'envisager quelques pistes de solutions. Elle se déroule de manière informelle avec le professeur référent.
- La **deuxième revue de projet** permet de vérifier les solutions retenues ainsi que les essais qui permettent d'atteindre progressivement le fonctionnement désiré de la réalisation. Cette revue fait l'objet d'une présentation orale individuelle (avec support multimédia) et se déroule en présence d'un professeur de spécialité et d'un professeur de SPC.
- La **troisième revue de projet** permet d'évaluer le niveau d'avancement du projet, d'élaborer une procédure de recette globale de la réalisation et l'intégration de sa partie dans ce qui sera présenté, lors de l'épreuve, devant la commission d'interrogation. Cette revue fait l'objet d'une présentation orale individuelle (avec support multimédia) et se déroule en présence d'un professeur de spécialité associé à un autre professeur de spécialité ou un professeur de SPC, en fonction de la spécificité du projet.



L'approche par étapes (avec des jalons) permet de bien structurer le projet dans le temps et d'assurer son suivi. Les jalons permettent de faire le point sur le projet et de n'engager la phase suivante que si tout va bien. Un jalon (*milestone*), dans le cadre de la gestion de projet, est la fin d'une étape, la fin d'un travail. La revue de projet consiste à valider si un jalon est passé, ou pas. Il y a donc une revue de projet à chaque jalon.

De manière générale, les revues permettent de :

- Faire le point sur l'avancement du projet ;
- Vérifier les documents attestant des résultats obtenus ;
- Mettre en commun les informations ;
- Donner une vision semblable du projet et de ses objectifs au sein de l'équipe ;
- Examiner les demandes de prise de décisions ;
- Valider les éléments présentés ;
- Gérer les risques, aléas et problèmes ;
- Cibler des aides ponctuelles éventuelles ;
- Permettre le droit à l'erreur, en appliquant une correction immédiate et collective de l'erreur ;
- Tirer collectivement les leçons de l'expérience ;
- Engager des actions correctives en cas de dérive ;
- Soutenir une coopération efficace entre les membres de l'équipe ;
- Mobiliser et motiver chaque acteur pour la suite du projet.

F.1 Ressources

- Conseils pour l'oral : tvaira.free.fr/projets/conseils.html
- Plan conseillé pour le diaporama : tvaira.free.fr/projets/diaporama.html

F.2 Revue n°1

La première revue de projet a pour objectif de vérifier la compréhension du travail demandé et la mise en œuvre du travail par les différents membres de l'équipe. Elle permet d'envisager quelques pistes de solutions.

F.3 Revue n°2

La deuxième revue de projet permet de vérifier les solutions retenues ainsi que les essais qui permettent d'atteindre progressivement le fonctionnement désiré de la réalisation. Cette revue fait l'objet d'une présentation orale individuelle (avec support multimédia).

L'équipe pédagogique évaluera :

- la compréhension et la présentation du cahier des charges
- la recherche d'informations complémentaires au cahier des charges
- la mise en œuvre des outils et des ressources disponibles (bouml, subversion, planner, ...)
- la conduite de projet (gestion de la planification, respect des délais)
- la rédaction du dossier technique
- la communication (lors de l'exposé)
- la gestion des relations de travail en équipe et l'autonomie

Par rapport à la gestion du projet :

- la prise en compte des tâches attribuées
- la mise en place d'un échancier (planification des tâches)
- la rédaction des premiers documents quant au choix et aux tests déjà réalisés
- l'identification des ressources et moyens communs
- l'organisation des communications et du suivi des documents

Le dossier technique doit contenir les productions suivantes :

- l'identification du travail à réaliser
- l'étude préliminaire
- le prototypage et maquette de l'IHM (diagrammes d'activités si nécessaire)
- la spécification de la base de données et des protocoles de communication
- les plans de test de validation (recette du produit fini)
- le diagramme des cas d'utilisation (commenté), la liste des acteurs
- les contraintes fonctionnelles et techniques
- pour chaque cas d'utilisation, les scénarios
- pour chaque scénario, diagrammes de séquence et de classes du scénario (commentés)
- le diagramme de classe d'analyse de l'application (le rôle de chaque classe est présenté)
- les tests unitaires et le code associé (itération 1)

<i>Revue n°2 : 30 minutes</i>		
Déroulement	Critères	Durée
Présentation orale en utilisant un diaporama	Qualité de la présentation, précision, rigueur, clarté	10 min
Démonstration du fonctionnement d'un sous-ensemble logiciel et/ou matériel	État et qualité de la démonstration Degré d'autonomie	10 min
Entretien avec l'équipe pédagogique sur la justification des choix et des éléments à corriger	Écoute et dialogue Pertinence, précision et exactitude Recherche et exploitation d'une documentation Argumentation et réaction aux objections	10 min

F.4 Revue n°3

La troisième revue de projet permet d'évaluer le niveau d'avancement du projet, d'élaborer une procédure de recette globale de la réalisation et l'intégration de sa partie. Cette revue fait l'objet d'une présentation orale individuelle (avec support multimédia).

L'équipe pédagogique évaluera :

- la réalisation de la conception détaillée et la définition des structures associées
- la fabrication individuelle des modules logiciels
- l'application des procédures de test
- l'interconnexion d'une partie du système

- le suivi, en équipe, des tests d'intégration et de validation de l'application
- la mise en œuvre des outils et ressources disponibles (debugger ...)
- la conduite de projet (gestion de la planification, respect des délais)
- l'application du plan qualité (règles de codage, identification des ressources)
- le dossier technique est complété
- une démonstration de l'application est réalisée
- la communication (lors de l'exposé)
- la gestion des relations de travail en équipe
- l'autonomie

Par rapport à la gestion du projet :

- le suivi de l'état d'avancement
- le respect de la répartition des tâches
- le respect de l'utilisation des ressources communes matérielles et logicielles
- le respect des délais et des plannings de réalisation
- la rédaction des documents d'activités et de suivi

Le dossier technique complété contient :

- l'architecture technique commentée (logicielle et matérielle)
- diagramme de déploiement
- plan des tests unitaires (méthodes, classes)
- diagrammes de composants pour la modélisation du code source et des exécutable, des tables
- algorithmes (limiter cette production aux parties significatives)
- diagrammes de séquence de conception (enrichi)
- diagramme de classe de conception (enrichi)
- conception détaillée des classes (attributs, méthodes)
- diagramme état-transition (si nécessaire)
- le bilan du travail restant à réaliser
- le code associé (itération 2)

<i>Revue n°3 : 1 heure</i>		
Déroulement	Critères	Durée
Présentation orale en utilisant un diaporama	Qualité de la présentation, précision, rigueur, clarté	20 min
Démonstration du fonctionnement d'un sous-ensemble logiciel et/ou matériel	État et qualité de la démonstration Degré d'autonomie	20 min
Entretien avec l'équipe pédagogique sur la justification des choix et des éléments à corriger	Écoute et dialogue Pertinence, précision et exactitude Recherche et exploitation d'une documentation Argumentation et réaction aux objections Prise en compte des conseils	20 min

F.5 Revue finale

Cette épreuve orale doit permettre au candidat de présenter son travail, de montrer son savoir-faire et de justifier ses choix dans les différentes activités de la conduite de projet. Le commission d'interrogation attribue une note à chaque candidat, à l'issue de la soutenance de son projet.

Cette note porte sur : la qualité et la valeur technique du dossier technique, la qualité de son exposé oral, l'état final de la réalisation examinée, la précision et l'exactitude de ses réponses, sa maîtrise des savoir faire définis dans le référentiel de certification.

La soutenance est organisée en trois phases. Une phase de présentation de 20 minutes, durant laquelle le candidat expose à la fois la globalité du projet et son action personnelle dans le projet puis une phase de mise en œuvre du projet de 20 minutes et une phase d'interrogation de 20 minutes durant laquelle le jury questionne le candidat.

Au cours de la soutenance du dossier, le candidat expose sans être interrompu, par la commission, le concept du produit final et la partie du dossier technique correspondant au travail dont il a la charge.

La présentation de la réalisation consiste en la mise en œuvre dûment commentée par le candidat, de façon à vérifier les éléments de recette.

Pendant l'entretien, le candidat doit répondre à des questions qui ont pour but d'évaluer la part de travail réel qu'il a réalisé, son niveau d'implication au sein de l'équipe et le degré de satisfaction des critères d'évaluation.

G Production attendue

- ⇒ Une application informatique fonctionnelle ;
- ⇒ Un modèle UML complet de la partie à développer ;
- ⇒ Le code source commenté de l'application ;
- ⇒ Les documents prévus au paragraphe « Exigences sur la livraison ».

H Tâches à réaliser

<i>Contrat individuel étudiant</i>		
Tâches à réaliser	Production attendue	Estimation horaire
S'approprier le cahier des charges	L'expression du besoin et le cahier des charges sont explicités. Les tâches à réaliser sont identifiées et les ressources sont définies.	16 h
Rechercher et choisir une solution technique	Une étude comparative argumentée des différentes solutions est produite. La commande du matériel et son approvisionnement sont gérés.	6 h (SCP ¹)
Installer et configurer son environnement de développement	Le poste de développement est opérationnel.	2 h
Installer et raccorder le matériel	Les appareils fonctionnent. La procédure d'installation a été respectée. Un compte rendu est rédigé.	2 h (SCP ¹)
Étudier et documenter les caractéristiques techniques	Les capteurs sont caractérisés à partir de l'étude et des mesures réalisées.	10 h (SCP ¹)
Paramétrer les ressources	Les ressources sont correctement paramétrées. Un plan de câblage est réalisé. Le rapport de tests de mise en œuvre est rédigé.	4 h + 4 h (SCP ¹)
Mettre en œuvre les programmes de test fournis ²	Le rapport de tests est renseigné.	12 h + 4 h (SCP ¹)
Finaliser la modélisation UML du module	Les diagrammes UML (diagrammes de séquence, diagramme d'état, ...) sont élaborés et finalisés.	20 h
Produire la maquette de l'IHM du module	La maquette de l'IHM correspond aux exigences du cahier des charges.	8 h
Coder et tester les classes du module	Les classes sont codées et valides.	44 h
Réaliser les tests unitaires	Les tests unitaires des classes sont écrits et archivés.	16 h
Faire la recette du module	Le cahier de recette du module est validé.	4 h + 4 h (SCP ¹)
Intégrer en équipe l'application complète	L'application est intégrée et fonctionnelle.	4 h
Rédiger le dossier technique et les documents relatifs au projet	Le dossier est rédigé en respectant les exigences.	30 h
Produire un guide de mise en route et d'utilisation du module.	Un manuel est fourni.	4 h
Gérer la planification	Le planning prévisionnel est établi. Le planning est actualisé avec une mise en évidence des écarts par rapport au prévisionnel.	10 h
Total		200 heures

1. SCP (Sciences Physiques) nombre d'heures donné à titre indicatif

2. cf. [Activités d'accompagnement](#)