

Consignes concernant le projet long

7 février 2023

Nous vous proposons de définir vous-mêmes le sujet du projet long sur lequel vous allez travailler. Vous trouverez ici les principales consignes et les jalons de ce projet.

Rappel : Ce projet sert de support à la matière Gestion de projet / Méthodes agiles (SCRUM) de l'UE SHS et sera évalué aussi bien par TOB (Projet Long) que par Gestion de projet. Vous aurez donc 2 notes.

0.0.1 Constitution des équipes

Une équipe (un groupe de projet) doit être constituée au sein d'un même groupe de TD (tous les membres d'une équipe doivent appartenir au même groupe de TD).

Il y aura 4 équipes par groupe de TD. Ceci fera donc des équipes de 7 personnes, plus ou moins une.

Chaque équipe sera suivie par un enseignant qui est son enseignant de TD TOB :

- AB : Xavier Crégut <prenom.nom@toulouse-inp.fr>
- CD : Judicael Bedouet <prenom.nom@onera.fr>
- EF : Meriem Ouederni <prenom.nom@toulouse-inp.fr>
- GH : Xavier Crégut <prenom.nom@toulouse-inp.fr>
- IJ : Judicael Bedouet <prenom.nom@onera.fr>
- KL : Guillaume Dupont <prenom.nom@toulouse-inp.fr>
- MN : Aurélie Hurault <prenom.nom@toulouse-inp.fr>

Pour la partie Méthodes agiles (SCRUM), vous pouvez poser vos questions à Gilles FRANCOIS <prenom.nom.e@thalesdigital.io> (quand le premier TD de la matière aura eu lieu).

Cette année les équipes de projet seront choisies au hasard... Sauf si les groupes de TD arrivent à former des 4 équipes avec comme règle « deux personnes dans la même équipe en PIM ne peuvent pas être dans la même équipe en TOB ».

Merci de signaler toute erreur à Xavier Crégut (prenom.nom@toulouse-inp.fr).

0.0.2 Notation du projet

- Chaque étape du projet (décrite ci-après) sera prise en compte dans la note du projet.
- À la fin du projet, une démonstration, une présentation orale et un compte-rendu concernant l'organisation du projet seront faits. **Trois personnes seront tirées au sort**, chacune

pour présenter l'un des aspects. Les autres membres de l'équipe répondront aux questions.

- Une partie de la note sera définie par l'équipe elle-même.

Chaque membre de l'équipe répartira un nombre de points égal à 1 de moins que la taille de son équipe entre ses coéquipiers. Chaque coéquipier aura un nombre de points compris entre -1 et 3. Par exemple, pour une équipe de 6 personnes, chaque équipier pourra répartir 5 points sur ses coéquipiers en donnant à chacun un nombre de points entre -1 et 3.

La note d'un étudiant sera donc : (note de l'enseignant) + (moyenne des points attribuées par ses coéquipiers - 1).

0.0.3 Projets Git

À venir...

0.0.4 2023/01/25, décalée à 2023/02/06 : Jalon « Équipes constituées » (étudiants... Ou enseignants).

0.0.5 2023/02/03, décalée au 2023/02/08 : Jalon « Création projets Git » (Xavier Crégut).

Création des projets Git qui seront utilisés pour le projet. Les projets ne peuvent être créés qu'une fois les équipes constituées.

0.0.6 2023/02/11 décalée à 2023/02/13 : Jalon « Liste des sujets envisagés » (équipes).

Chaque équipe déposera sur le projet Git dans le dossier `livrables` un document `sujets.pdf` qui contiendra au moins 4 sujets de projet envisagés, classés par ordre de préférence décroissante de l'équipe (le premier sujet est le sujet préféré de l'équipe).

Les sujets proposés doivent être décrits dans le document `sujets.pdf` (tout en minuscules) qui doit être déposé dans le dossier « `livrables` »

Chaque sujet doit être décrit sur une page environ dans le but de convaincre de la pertinence et de l'intérêt de votre projet. Vous devez considérer que ce document s'adresse à des clients potentiels, à des business angels, à votre supérieur hiérarchique...

Il doit y avoir suffisamment d'information sur le projet pour que votre enseignant puisse le comprendre et évaluer le travail envisagé. Ce document sera le point de départ pour discuter de votre projet avec votre enseignant. C'est lui qui validera le sujet de votre projet au final.

En aucun cas il ne faut parler des solutions informatiques qui sera apportées pour réaliser ce projet. Il ne s'agit pas de décrire comment ce sujet sera traité d'un point de vue programmation mais ce qu'il apportera à ses utilisateurs.

Quelques recommandations :

1. Le projet doit être suffisamment ambitieux puisque les équipes sont de grande taille ! Vous ne devez pas pouvoir terminer le projet dans le temps imparti. C'est le but.
2. L'application développée devra être pourvue d'une interface graphique.

3. Vous ne ferez ni une application concurrente, ni une application distribuée, ni une application client/serveur (ces notions seront vues en 2A).

Attention, vous ne pouvez pas prendre un sujet qui serait du type « boîte à jeux » où chaque membre de l'équipe s'occuperait d'un des jeux !

L'enseignant évaluera les sujets et donnera le feu vert à l'équipe pour traiter l'un des sujets (ou vous aidera à construire un sujet adapté aux attentes).

Exemples de sujets proposés et choisis les années passées

- Simulation de la propagation d'un virus
- Application de proposition de repas
- Gestion et maintenance d'une maison intelligente
- Escape game
- Réalisation d'un RayTracer
- Jeu guidé pour apprendre la programmation
- Application de gestion des crèches
- Manuscript Manager (aide à l'écriture d'un roman)

0.0.7 2023/02/17 : Jalon « Sélection du sujet de projet » (enseignant responsable de l'équipe de projet).

L'enseignant responsable de l'équipe de projet communiquera le projet retenu.

0.0.8 2023/02/25 : Jalon « Fonctionnalités de l'application » (équipe).

Il s'agit de décrire les fonctionnalités de l'application que vous allez développer dans un document appelé fonctionnalites.pdf (nom à respecter impérativement, nom tout en minuscules, sans accents), à déposer dans le dossier « livrables ». Il doit contenir :

1. une page de garde qui rappelle le numéro de l'équipe, ses membres, le titre du projet, l'objet du document...
2. l'objectif général du projet
3. une description des fonctionnalités (attendues) de l'application qui sera développée. Il s'agit de décrire le service rendu (besoin) et non la manière dont ce service sera rendu (comment).
4. les interfaces utilisateur envisagées (esquisses réalisées à main levée ou avec un outil de dessin).
5. plusieurs cas d'usage (scénarios) pour comprendre les fonctionnalités (et, plus tard, valider l'application développée).
6. (facultatif) En annexe, lister les points durs, les points qui vous paraissent difficiles dans le projet proposé.

Il ne faut pas hésiter à mettre beaucoup de fonctionnalités. Les étapes suivantes consisteront à les organiser, définir des priorités, etc. Tout ne sera pas à développer lors du projet long !

Attention : Ici, on ne doit pas parler de programmation objet, d'UML, de Java, etc. On ne veut pas savoir comment sera réalisée l'application mais bien qu'elles sont les possibilités qu'elle offrira à ses utilisateurs.

Ce document pourrait donc servir de trame pour le futur manuel utilisateur. Il restera à ajouter quelques captures d'écrans pour remplacer les esquisses et les modes opératoires tels qu'ils ont été finalement réalisés.

0.0.9 Semaines 13 à 21 : Réalisation du projet en 3 itérations.

Le projet sera organisé en trois itérations de 2 semaines (plus une itération 0).

Attention : à chaque fin d'itération, chaque membre de l'équipe doit déposer un très court rapport INDIVIDUEL qui décrit ce qu'il a fait lors de l'itération. Ce rapport sera nommé rapport#-\$USER.pdf (tout en minuscules) où \$USER est votre identifiant LDAP (votre login) et # est le numéro de l'itération (1, 2 ou 3). Ce rapport doit être court et précis et lister les activités réalisées (en indiquant le degré d'avancement) sur une page maximum. Les détails de la réalisation seront donnés dans le rapport général.

À la fin de chaque itération, il faut aussi rendre le code complet de l'application dans un fichier source#.jar, le manuel utilisateur manuel-utilisateur#.pdf et le rapport associé rapport#.pdf, # est le numéro de l'itération (1, 2 ou 3).

Le rapport documente votre projet. Il doit inclure les éléments élaborés suite aux séances gestion de projet mais aussi les éléments de conception (diagrammes UML). Les versions intermédiaires de ce rapport ne seront pas évaluées. Le rapport à la dernière itération constituera le rapport final.

Le manuel utilisateur explique comment utiliser le programme livré. Comme pour le rapport, les version intermédiaires ne seront pas évaluées.

Attention à bien vérifier que les .java sont présents dans l'archive).

On peut construire un jar depuis Eclipse (il faut penser à sélectionner l'inclusion des sources) ou depuis la ligne de commande. Les options de la commande jar sont globalement les mêmes que tar.

0.0.10 Semaine 13 : 2023/03/27 au 2023/04/01, Itération 0

.

0.0.11 Semaines 14 et 15 : 2023/04/03 au 2023/04/14, Itération 1

.

0.0.12 2023/04/14, Remise des livrables Itération 1 (rapports individuels + code + manuel utilisateur + rapport général).

0.0.13 Semaines 16 & 19 : 2023/04/16 au 2023/05/13, Itération 2

.

0.0.14 2023/04/13, Fin Itération 2 : rapports individuels + code + manuel utilisateur + rapport général.

0.0.15 Semaines 20 & 21 : 2023/05/15 au 2023/05/26, Itération 3

0.0.16 2023/05/26, Fin Itération 3 : rapports individuels + code + manuel utilisateur + rapport général).

Vous devez rendre dans le dossier livrables de votre dépôt SVN :

1. Le rapport (rapport3.pdf) doit inclure :

- (a) Un rappel très succinct du sujet. Les détails seront dans le manuel utilisateur.
- (b) Les principales fonctionnalités (user stories ou epics) identifiées en précisant leur état de réalisation (et dans quelle itération elles ont été faites pour celles qui sont terminées ou commencées).
- (c) Le découpage de l'application en sous-systèmes (en paquetages).
- (d) Les diagrammes de classe qui donnent l'architecture de votre application. On pourra soit faire un diagramme de classe global, soit un diagramme de classe par fonctionnalité.
On ne fera apparaître sur ces diagrammes que les informations utiles pour comprendre l'architecture. En particulier, il est inutile de faire apparaître toutes les méthodes de toutes les classes.
- (e) Les principaux choix de conception et réalisation, les problèmes rencontrés et les solutions apportées, etc.
- (f) L'organisation de l'équipe et la mise en œuvre des méthodes agiles.

Il n'y a pas de nombre de pages à respecter. Il faut être concis et précis. Sachez que ce rapport est le point d'entrée sur votre code. Il doit permettre au lecteur de comprendre comment vous avez traité votre sujet et donc de savoir ce que vous avez fait sans avoir à consulter le code. Si le lecteur est intéressé par un aspect particulier, la lecture du rapport devrait l'éclairer sur ce qui a été fait, comment ça a été fait et les sources à lire pour avoir tous les détails d'implantation.

2. le manuel utilisateur : manuel-utilisateur3.pdf

3. les sources de votre projet : sources3.jar.

0.0.17 2023/05/30, 8h : Remise du support pour la présentation orale

Le support de présentation utilisé pour l'oral du projet : presentation.pdf (voir ci-dessous pour les consignes) doit être déposé dans le dossier livrables.

0.0.18 2023/05/30, 8h : Points d'équipe

Modalité à venir...

0.0.19 2023/05/30 : Présentation orale (à partir des livrables du Gitlab).

Pour la **présentation orale**, le déroulement sera le suivant :

1. 15 minutes (maximum à ne pas dépasser !) à utiliser pour :
 - une démonstration de l'application réalisée
 - une présentation technique du travail réalisé (architecture, choix de conception et réalisation, etc.)
 - une présentation de l'organisation de l'équipe.

Vous devriez passer de 3 à 6 minutes sur chacune de ces trois parties.

Attention : Les trois orateurs seront tirés au sort. Chacun doit donc être prêt à présenter chacune des trois parties.

Vous êtes libres d'organiser ces trois présentations dans l'ordre que vous souhaitez (même s'il paraît logique de commencer par la démonstration car tous les membres du jury ne connaîtront pas votre sujet).

2. 10-15 minutes de questions/discussions.

Il peut être intéressant de commencer par la démonstration car c'est l'occasion de rappeler le périmètre du projet et comment il a été traité du point de vue des utilisateurs. La présentation orale se concentrera sur les aspects techniques et organisationnels du projet.

La **présentation technique** doit être technique ! Il n'est pas utile de nous détailler le sujet (la démonstration l'a fait). Ce qu'il faut faire, c'est nous expliquer comment vous l'avez traité : quels sont les choix faits ? Quels sont les problèmes rencontrés et les solutions apportées. Nous vous rappelons qu'UML est un bon outil pour communiquer !

La **gestion de projet** doit expliquer comment l'équipe s'est organisée, les outils utilisés en accord avec ce qui a été présenté en gestion de projet.

Horaires de passage pour l'oral

Toutes les équipes assistent à toutes les présentations de leur groupe de TD.

L'ordre de passage des équipes sera choisi au fur et à mesure.