MIAGE 1
Projet Développement Logiciel
–
Responsable : Adrien Le Roch
Contact : adrien.leroch@univ-rennes1.fr
2022/2023

Gestion de projet – Projet Développement Logiciel (PDL)

Votre équipe a la responsabilité de mener à bien un projet logiciel avec des technologies et données ouvertes. De nombreux défis sont à relever, nécessitant des compétences en gestion de projet, en modélisation, et en programmation.

Cette mise en situation doit permettre de mieux comprendre et appréhender la difficulté de développer du logiciel dans un contexte concret. Des techniques et outils (git, github/gitlab, Maven, JUnit, etc.) de développement logiciel, bien connus de l'industrie, seront utilisés. Des choix technologiques devront également être effectuées. Il est attendu de la part de chaque étudiant de PDL de démontrer qu'elle ou il est capable :

- de comprendre le besoin client et proposer une solution technique (exigences, spécificité des technologies et du domaine d'application, architecture, format des données, etc.)
- de contribuer très concrètement à un projet à la fois au niveau du code (créations/ajouts de fonctionnalités, cas de tests, etc.), de la documentation, ou éventuellement (pour les motivés) de la mise en œuvre de l'intégration continue
- de maîtriser un ensemble de technologies (Java et son écosystème, CSV, JSON, JUnit, API Web, etc.) et de techniques (e.g., test) importantes
- de s'adapter à l'évolution d'un projet et de ses exigences
- de travailler collectivement
- de tenir les dates de rendu

- de valider de manière continue les exigences et l'implémentation[1]

1 Contexte général des travaux à réaliser

Cette année, un seul sujet est proposé. Ce sujet donne lieu à la réalisation d'une application *from scratch.* Le résultat final attendu est une application fonctionnelle et stable offrant les fonctionnalités attendues.

En considérant le résultat final précédemment présenté, il faut faire des choix technologiques afin de répondre au besoin. Le choix des technologies est à l'initiative du groupe de travail et sera à valider avec le client.

La note pourra être individualisée si on observe un manque évident de travail à l'intérieur d'un groupe. À ce titre, l'activité sur Github/Gitlab sera observée. Par défaut, la note du groupe s'applique à chaque membre.

2 Travail demandé

Au cours du projet vous devrez rendre différents « livrables ».

Eliciter des exigences (EX)

Vous allez écrire trois documents sur votre dépôt Github/Gitlab:

- Un document README.md qui permet à tout visiteur du projet de comprendre rapidement le but du projet, les fonctionnalités supportées, les fonctionnalités à développer dans le futur, la couverture des tests, les technologies utilisées, les participants et le contexte du projet, etc.
- Un document d'architecture DESIGN.md à la racine du Github/Gitlab qui reprend les grands éléments d'architecture incluant des modèles statiques

[1] La « validation » continue a différents impacts sur le déroulement du projet : il s'agit de valider avec le client les exigences, de démontrer avec différents prototypes la valeur de son travail et sa conformité par rapport aux attentes, de tester l'implémentation pour démontrer sa robustesse, etc.

(organisation des packages, descriptions des classes principales et de leurs responsabilités, etc.) ainsi que des modèles dynamiques (flux des événements, scénarios nominaux et exceptionnels, etc.). Il est fortement conseillé d'utiliser UML (diagramme de cas d'utilisation, diagramme de classes/d'objets, machines à états, etc.). L'objectif de ce document est qu'un potentiel contributeur externe puisse comprendre l'architecture et les fonctionnalités du projet pour pouvoir éventuellement reprendre le code et l'étendre/l'améliorer.

- Un document INSTALL.md décrivant comment on construit le projet à partir du code source, comment on exécute les suites de tests, comment on exécute le logiciel
- Un cahier de recette RECETTE.md décrivant les tests à effectuer depuis le front end afin de valider les fonctionnalités de l'application.

Tous les documents peuvent être écrits en anglais ou en français.

Il y a un « client » : Adrien Le Roch (adrien.leroch@univ-rennes1.fr). Vous devez discuter avec le client tout au long du projet pour valider vos choix.

La date de rendu pour les trois documents est fixée au **15 octobre 2023**.

Attention : le livrable EX implique de *dores et déjà maîtriser les technologies choisies et de développer une solution logicielle* ; ce travail est indispensable pour pouvoir discuter avec le client des exigences, des choix technologiques, et de valider avec lui certains prototypes.
Il est possible qu'au cours de l'avancée du projet certains éléments (fonctionnalités, choix techniques, etc.) soient remis en cause : c'est parfaitement normal ! Les groupes décriront et justifieront ces changements lors de soutenance (cf PR ci-dessous).

Sprint (SP1 + SP2)

Au cours du projet, il est demandé d'effectuer deux tâches précises, non triviales dans un laps de temps prédéfini.

Deux sprints sont à réaliser, SP1 et SP2,

SP1 : création de la V1 (Version 1)

L'objectif de SP1 est de réaliser une première version du logiciel (qu'on appellera « version 1 », abrégée en V1). Cette version sera livrée au client afin qu'il réalise des tests et fasse un retour global et concret. Il est donc essentiel que cette V1 intègre la majorité des fonctionnalités attendues par le client. Il est cependant tout à fait acceptable que cette version comporte des bugs, que l'IHM soit peu ergonome, voir même que certaines fonctionnalités (idéalement les fonctionnalités les moins importantes) ne soit pas encore testables.

Plusieurs éléments sont attendus à la livraison de la V0 :

- Code propre et rédigé de manière cohérente (exemple ici)
- Présence de commentaires type Javadoc pour les classes, attributs de classe et méthodes
- Bugs (si bugs il y a) tracés sur GitHub/GitLab
- Couverture de test d'au moins 50 %
- Mise à jour des documents README.md, DESIGN.md et INSTALL.md

Concernant les bugs et les tests unitaires, la méthode suivante sera à suivre :

- Écrire une issue Github/GitLab (en anglais) à chaque fois que vous identifiez un bug
- Écrire des cas de tests automatiques (avec JUnit) qui démontrent la présence de bugs
- Ensuite, et seulement ensuite, corriger/modifier le logiciel

En sortie du SP1, le client réalisera une mini-audition de la V0 et des retours vous seront fournis par mail et oralement. Ces retours seront à prendre en compte lors du SP2.

La date de rendu de la V0 est fixée au **26 novembre 2023.**

SP2 : prise en compte des retours du SP1 finition du logiciel pour atteindre la V2 (version 2)

Les objectifs sur SP2 s'inscrivent directement dans la continuité du SP1, à la différence que l'objectif est cette fois-ci

de produire une version du logiciel satisfaisant tout les fonctionnalités attendues par le client, testé au maximum, relativement facile à utiliser (l'ergonomie n'est pas votre spécialité donc on ne va pas trop s'attarder dessus) et stable (pas de crash ou de blocage).

La présence de bugs mineurs, en nombre raisonnable, sera toléré si ceux-ci sont tracés sur GitHub/GitLab.

Les documents sont aussi à rendre dans leurs versions finalisées. Le README.md sera à mettre à jour pour expliciter d'éventuels fonctionnalités non développés (d'un commun accord avec le client).

La date de rendu de la V2 est fixée au **22 décembre 2023.**

Le code sera nécessairement hébergé sur github ou gitlab, dans un repository publique.

Présentation (PR)

- 20' de présentation
  - Rappel du contexte
  - Capture des exigences et validation avec le client (EX)
  - Expliquer l'implémentation, l'architecture du projet, les technologies utilisées, et son déploiement (SP)
  - Démonstration (SP)
  - Retour d'expérience (difficultés rencontrées, adéquation par rapport au cahier des charges, technologies et méthodes maîtrisées, etc.)
- 10' de questions par le jury

Rendu

Les dates de rendus sont strictes et imposées.
EX : 15 octobre 2023 (23h59) ;
SP1 : 26 novembre 2023 (23h59)
SP2 : 22 décembre 2023 (23h59)
PR : mi-janvier 2020
La note est sur 20:

- 5 points pour EX,
- 10 points pour SP1+SP2 (dont 5 points sur les tests)
- 5 points pour PR

Les notifications des rendus s'effectueront par un email du responsable du groupe (et évidemment via Github/GitLab pour le code et les instructions).

Comment commencer ?

Avant de commencer la rédaction du moindre document, l'implémentation, ou de simplement discuter avec le « client » , il est **nécessaire d'écrire, dans le document README.md, l'objectif général du projet *avec vos propres mots*.** Cela doit valider votre compréhension du sujet.

Un autre objectif à atteindre le plus rapidement possible est que chaque membre du groupe soit à même de commiter le code qu'il aura développé, de prendre en compte les mises à jour de ses collègues, et bien sûr d'exécuter l'application et les cas de test. **Il faut donc mettre en place un « repository » Github/Gitlab très rapidement.**

Bon projet