

ECOLE PRATIQUE DES HAUTES ETUDES COMMERCIALES



EPHEC

INSTITUT D'ENSEIGNEMENT SUPERIEUR DE TYPE COURT

Projet d'intégration

Enoncé du projet

2019 – 2020

**Marie-Noël Vroman
Louis Van Dormael**

Baccalauréat en :

- *Technologie de l'Informatique*

1.	Introduction.....	3
2.	Acquis d'apprentissage du projet.....	3
3.	Choix du sujet.....	4
4.	Méthode SCRUM appliquée au projet.....	5
4.1.	Rôles et constitution de l'équipe.....	5
4.2.	Organisation temporelle	5
	Dates particulières :	6
	Sprints.....	6
	Séance encadrée de milieu de sprint	6
	Séance encadrée de début/fin de sprint : démo et rapport de sprint.....	7
	Phase de finalisation.....	7
4.3.	Outils à utiliser.....	8
	Trello	8
	Versionning.....	10
	Timesheets.....	10
5.	Deadlines et livrables.....	11
5.1.	Deadlines en fin de phase d'initialisation.....	11
5.2.	Deadlines de fin de sprint.....	11
5.3.	Deadline finale	11
	Sur la plateforme de développement (Github ou Bitbucket) :.....	11
	Sur Trello :	11
	Sur Moodle, au format PDF :	12
5.4.	Défense du projet	12
6.	Critères d'évaluation.....	13

1. Introduction

La troisième année du baccalauréat en Technologie de l'Informatique clôt le cursus en permettant à l'étudiant de faire le lien entre les différentes compétences acquises durant les deux premières années, et met l'accent sur l'intégration de ces compétences dans des projets professionnels. Cela se traduit au second semestre par la réalisation d'un stage en entreprise et d'un TFE, mais également, lors du premier semestre, par la réalisation d'un projet d'intégration en équipe.

L'objectif de ce projet est la réalisation d'un produit fini qui :

- illustre les compétences étudiées durant les deux premières années ainsi que la capacité des étudiants à les intégrer entre elles,
- témoigne de la mise en pratique d'une méthodologie professionnelle de gestion de projet et d'une démarche de qualité dans la réalisation

Concrètement, le thème du projet est laissé au choix de chaque équipe, modulo le respect de certaines contraintes listées plus bas. Au delà du sujet, les consignes de ce projet porteront essentiellement sur la méthodologie à adopter : il est demandé à chaque équipe de mettre en pratique la méthodologie *Scrum*, qui est une méthode de gestion de projets dite *Agile*.

Chaque équipe devra ainsi travailler par itérations de deux semaines, chaque itération visant à produire une version fonctionnelle du produit, qui sera validée par le « client ». Le but est d'arriver, à la fin du projet, à une réalisation de qualité dont les fonctionnalités correspondent au besoin du client, et dont l'ampleur aura été ajustée au fur et à mesure en priorisant les fonctionnalités pour arriver à un équilibre entre rythme de production de l'équipe et besoins du client.

Le projet d'intégration sera également l'occasion pour les étudiants d'entrer dans une démarche entrepreneuriale et de travailler les aspects liés à la Créativité et la Communication. Pour cela, vous aurez l'occasion de participer à des ateliers et d'être coachés à différents moments du projet par des entrepreneurs.

Enfin, l'approche du projet prendra en compte l'aspect RSE, dans le cadre du cours donné par Mme Hecquet.

2. Acquis d'apprentissage du projet

Ce projet permettra à l'étudiant de progresser dans les compétences suivantes :

- Gérer, en groupe, un projet multidisciplinaire
- Elaborer une méthodologie de travail (ici, Scrum) et de mettre en place, afin de les utiliser des outils de travail collaboratifs (Github, Trello, Timesheet...);
- Rechercher et utiliser les ressources adéquates
- Travailler tant en autonomie qu'en équipe dans le respect de la structure de l'environnement professionnel
- En choisissant une méthode d'analyse adaptée, exprimer une solution, par écrit et oralement, en tenant compte des contraintes, avec les formalismes appropriés, mais aussi pouvoir exprimer cette solution avec le vocabulaire technique adéquat aussi bien que dans un vocabulaire compréhensible par quelqu'un n'étant pas du métier;
- Assurer l'intégration d'éléments matériels informatiques et logiciels s'exécutant sous le contrôle de différents systèmes d'exploitation

3. Choix du sujet

Dans le cadre de ce projet, vous serez amenés à mettre au point un produit de votre choix, innovant et représentatif des compétences que vous avez acquises au cours de votre cursus à l'Ephec. Le choix de ce sujet va se faire en trois étapes :

1. Ateliers Créativité/Recherche d'idées, lors de la première semaine
2. Chaque groupe sélectionne une idée, la spécifie et la fait valider par l'équipe professorale.
3. Chaque groupe approfondit l'idée sélectionnée et la raffine en « fonctionnalités utilisateurs » (*user stories*) pour créer son *Product Backlog*.

Le sujet choisi doit permettre la réalisation d'un **produit fini fonctionnel** endéans les délais prescrits. Il doit également **intégrer** plusieurs compétences développées durant le cursus. A la base, il devrait au minimum comporter :

- Un site ou une application Web avec base de données
- Un aspect infrastructure réseaux
- Une analyse de sécurité et la mise en place de contre-mesures ad-hoc
- Une méthodologie de validation structurée et un souci qualitatif important (TDD, documentation, code propre, mode d'emploi clair et précis, ...)

En outre, il doit également comporter un aspect supplémentaire parmi la liste ci-dessous :

- Electronique (PIC, arduino, IoT...)
- Programmation mobile
- Traitement du signal
- VoIP
- ...

Différentes orientations sont possibles pour votre projet : Un objet connecté, une application personnelle, une plateforme en ligne, de la domotique, un projet plus « orienté hardware », ... N'oubliez pas non plus que vous avez la possibilité d'exploiter les ressources du FabLab pour développer un aspect « tangible » de votre projet. Le champ des possibles est vaste, et l'originalité d'une idée peut éventuellement primer sur le respect strict des consignes du projet ! N'hésitez donc pas à échanger avec l'équipe avant que cette dernière valide (ou non) votre sujet.

4. Méthode SCRUM appliquée au projet

Les éléments de la méthode Scrum qui seront mis en place dans le cadre du projet sont décrits dans le document « Introduction à Scrum », distribué conjointement à cet énoncé. Nous allons ici détailler la manière dont ces éléments sont « mis en scène » dans le cadre du projet.

4.1. Rôles et constitution de l'équipe

Chaque équipe sera constituée d'un nombre d'étudiants qui vous sera communiqué.

Parmi ces étudiants, il faudra désigner un Product Owner. Le Product Owner gardera ce rôle durant l'entièreté du projet. Ses responsabilités seront différentes de celles des autres étudiants, il doit donc en être conscient et être motivé et prêt à s'impliquer pour assurer ce rôle au mieux. Il aura à gérer l'évolution du Backlog, et devra garder une trace des raisons des principales modifications au backlog pour la réflexion demandée dans le rapport final.

L'équipe doit également se choisir un Scrum Master. Ce rôle peut être dynamique, et le Scrum Master peut changer d'un Sprint à l'autre. Chaque équipe peut décider de son mode de fonctionnement (Scrum Master unique sur tout le projet, ou bien rotation tous les sprints ou tous les deux sprints).

(Parmi les étudiants développeurs, en fonction du sujet, il pourra être utile de désigner des spécialistes. Ex : web, base de données, ... Le choix de ces spécialisations est laissé à l'appréciation de chaque équipe.)

Au niveau de l'équipe enseignante, deux personnes-ressources seront disponibles. D'une part, le professeur titulaire du groupe assurera le rôle de « coach », et passera dans les équipes durant les séances encadrées pour répondre aux questions des étudiants et observer la manière dont le processus se déroule. Le « coach » interviendra dans la rétrospective de sprint.

D'autre part, le professeur titulaire de l'autre groupe jouera le rôle de « client » ou d'« utilisateur-type », et assistera à la review de sprint. Le « client » sera le contact privilégié du Product Owner en cas d'interrogation de l'équipe sur certaines fonctionnalités

4.2. Organisation temporelle

Le projet correspondra à une unique release, elle-même composée de cinq sprints. La chronologie du projet est la suivante :

Le planning lié aux sprints est indiqué sur **la page Moodle du cours**. Consultez-le quotidiennement.

Le planning des ateliers s'y trouve également. (Lorsqu'un atelier est optionnel, cela est indiqué.)

Dates particulières :

- Mercredi 18 sept : de P1 à P5 ateliers Créativité et Disques
- Jeudi 19 sept 8h30 :
 - avoir une idée de vos groupes et des projets à développer
avoir lu le document d'introduction à scrum (Moodle)
- Lundi 02 oct 8h30 :
 - avoir remis sur Moodle un document comprenant :
 - Une **description textuelle du sujet du projet**, positionnant brièvement le produit (public-cible, avantage par rapport à l'existant, valeur ajoutée pour l'utilisateur, ...)
 - La **constitution détaillée de l'équipe**, avec pour chaque personne, les caractéristiques de son profil, ses points forts et ses points faibles
 - Les membres qui joueront le rôle de Product Owner et de Scrum Master
 - Les liens vers vos ressources partagées : trello, github, timesheet, autres...
 - Avoir complété votre trello avec
 - les différentes user stories et technical tasks du projet
 - les tâches sélectionnées, voire commencées ou terminées, pour le Sprint1, ainsi que les points associés
- Chaque semaine, être en ordre au niveau du github, du trello et du timesheet
- A chaque démo de sprint, un rapport de sprint (voir chapitre spécifique)
- Lundi 17 décembre à 18h : rapport final (voir chapitre spécifique)

Sprints

Les sprints sont au nombre de cinq, et comportent les jalons Scrum habituels : Le planning de sprint, les scrums quotidiens, la revue/démo et la rétrospective. Leur durée est de deux semaines. (Exceptionnellement, le sprint 3 qui inclut la semaine de Toussaint aura une durée de 3 semaines.)

Les scrums quotidiens sont à organiser de manière autonome par l'équipe en fonction de ses disponibilités et des contraintes liées aux autres cours. Lorsqu'il y a une séance d'avancement de milieu de sprint, un scrum est à prévoir pour profiter de la présence de l'encadrant « coach ».

La séance encadrée correspondant à la fin d'un sprint et au début du suivant sera très importante, car il faudra y organiser d'une part, la revue de sprint comprenant la démo à l'enseignant jouant le rôle du client ainsi que le rapport de sprint, et d'autre part la rétrospective et le planning du sprint suivant

Séance encadrée de milieu de sprint

En milieu de sprint, la séance encadrée sera consacrée à une entrevue avec le coach. Cette entrevue pourra, au choix de l'équipe, prendre la forme du scrum quotidien, ou bien pourra être l'occasion de discuter de points qui posent éventuellement problème.

L'objectif de cette réunion est d'aider les étudiants en cas de blocage, et de fournir du feedback sur la manière dont le travail d'équipe est mené jusqu'à là. L'interlocuteur privilégié du coach sera a priori le Scrum Master puisque la discussion sera centrée sur le processus, mais il est évident que chaque membre de l'équipe est invité à interpeller le coach sur quelque sujet que ce soit.

Séance encadrée de début/fin de sprint : démo et rapport de sprint

La première période de cette séance sera consacrée à la revue de Sprint. Etant données les contraintes temporelles, l'enseignant « client » ne pourra consacrer qu'un quart d'heure par groupe. Chaque équipe devra donc être prête pour la démo, en ayant soigneusement préparé les éléments suivants :

- La démo elle-même, prête à être démarrée dès l'arrivée du client.
- Un rapport de sprint avec :
 - Un lien vers la release produite durant le Sprint
 - Un petit rapport faisant le bilan du Sprint écoulé (point de vue avancement technique, méthodologie et gestion d'équipe), une analyse réflexive du fonctionnement du groupe (Scrum Master) ainsi que toute information ou remarque pertinentes vis-à-vis de ce sprint
 - Un récapitulatif des objectifs du sprint qui se termine, à savoir la liste des user stories initialement prévues et celles des user stories effectivement terminées (responsabilité du Product Owner)
 - Timesheet représentant le travail effectué par les membres de l'équipe
 - Schémas ou analyses pertinents réalisés durant le sprint (réseaux, électronique, database, architecture, fonctionnel, sécurité, etc...)
 - Les tests unitaires, fonctionnels ou automatisés réalisés
 - Une estimation de la vélocité de l'équipe, à savoir le nombre de points produits par l'équipe durant ce sprint (responsabilité du Scrum Master)
 - Les forces et faiblesses du sprint précédent, ainsi que les mesures prises pour le sprint actuel
 - Objectifs du prochain sprint, c'est-à-dire les use cases préparés et prioritaires qui vont servir pour la planification du prochain sprint (responsabilité du Product Owner)

L'objectif de cette démo est d'obtenir le feedback du client, d'une part sur ce qui a été montré, et d'autre part, sur ce qui est prévu. Le Client peut suggérer de nouvelles User Stories, ou bien proposer des modifications aux User Stories prévues pour le prochain Sprint. L'output de la démo est donc un Product Backlog et un planning de sprint mis à jour. L'interlocuteur privilégié du client sera le Product Owner, mais la discussion doit bien entendu être étendue à toute l'équipe, et les décisions doivent être prises collectivement.

Phase de finalisation

A la fin du dernier sprint, la review de sprint devra faire le point sur les tâches restant à effectuer pour que le produit soit livrable et livré et documenté: Mise en production et finalisation des livrables. Cette phase de finalisation se clôture par la remise des livrables et la présentation du projet.

4.3. Outils à utiliser

Afin de mener à bien le projet, il est demandé aux étudiants d'utiliser les outils suivants. Nous attendons de votre part un travail quotidien, de tous les membres de l'équipe. Ces outils, en plus de vous aider, nous permettront de suivre plus simplement l'activité de vos projets.

Trello

Trello est un outil simple permettant de mettre en place un Product Backlog facilement utilisable par tous. Il est demandé aux étudiants de donner accès à leur Product Backlog aux enseignants.

Chaque étudiant doit donc se créer un compte sur Trello. L'étudiant chargé du rôle de Product Owner créera le « Board » pour le projet, et le partagera donc avec les autres membres du projet ainsi qu'avec l'équipe enseignante.

Chaque équipe est libre d'organiser son board comme elle le souhaite, mais il est néanmoins fortement conseillé de suivre les guidelines suivantes :

- Créer cinq listes de cartes : To Do, Doing, To Do This Sprint, Done This Sprint, Done since Beginning.
- Utiliser une carte par Story (User Story, Technical Story ou autre)
- Identifier le type de story dans le titre de chaque carte, soit par un acronyme, soit par une couleur (ex : US pour User Story, TS pour Technical Story, B pour Bug,...)
- Décrire les User Story en utilisant la formulation « En tant que ..., je veux ... afin de... » (cfr document d'introduction à Scrum).
- Raffiner les Stories en tâches à l'intérieur d'une carte en utilisant la checklist
- Indiquer le n° du Sprint pour lequel la Story est prévue dans le titre de la carte
- Indiquer l'estimation en points de la story dans le titre de la carte.
- Assigner un développeur par carte/story

La figure 1 montre un exemple fictif de Backlog Produit construit avec Trello selon ces recommandations.

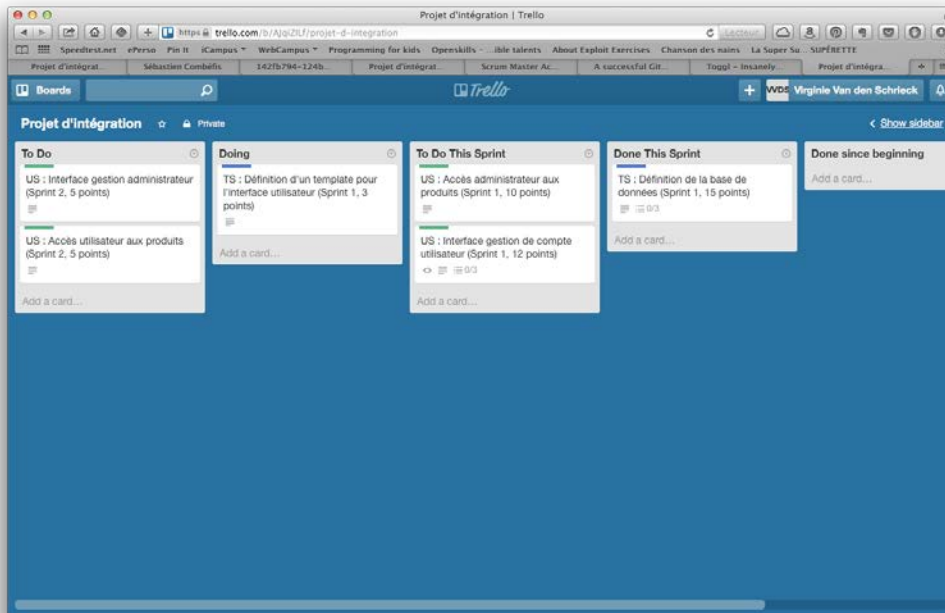


Figure 1 : Backlog produitsur Trello. Ici, le type des stories est indiqué à la fois dans le titre et par l'utilisation des tags de couleurs (bleu = story technique, vert = user story)

La figure 2, quant à elle, montre le contenu d'une des cartes du board. Cette carte est une story de type User Story. Notez l'utilisation de labels, l'estimation en points, l'identification du sprint, la description de la story, la liste des tâches et l'assignation à un développeur.

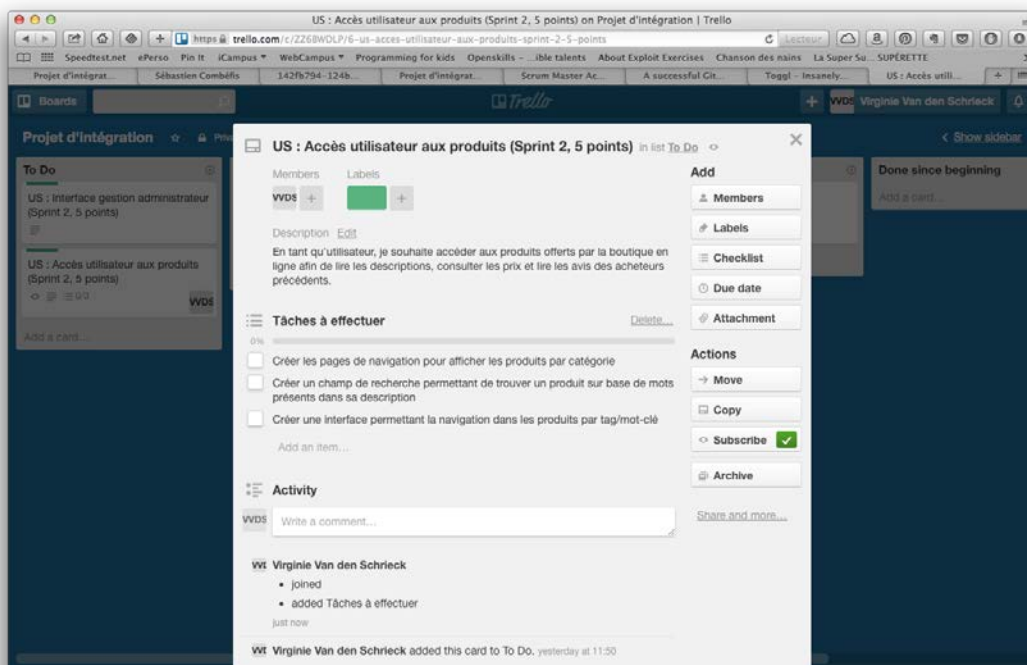


Figure 2 : Détails d'une User Story sur Trello

Remarque : il existe d'autres outils permettant d'effectuer cette gestion des tâches, certains plus poussés et plus spécifiques au développement informatique. Nous préférons par simplicité que vous utilisiez Trello. Si réellement vous désirez utiliser un

autre outil, venez nous en parler avant, soyez réellement convaincant sur les raisons qui vous amènent à l'utiliser, recevez notre accord, fournissez-nous toutes les ressources nécessaires pour facilement consulter votre outil ... et faites concrètement usage des fonctionnalités supplémentaires qui vous amèneraient à utiliser cet outil. Cette remarque est valable également pour l'outil de versionning, présenté ci-dessous.

Versionning

Le produit devra être développé en utilisant un outil de versionning et une plate-forme en ligne, à savoir Github. A partir de la fin du premier sprint, une version stable du produit partiel doit être **téléchargeable, installable documentée et utilisable** à tout moment (release). Cela implique donc que les étudiants travaillent avec différentes branches, et aient une stratégie cohérente de gestion de leurs versions. Il est recommandé de s'inspirer du « git flow », documenté entre autres dans les deux pages suivantes :

<http://nvie.com/posts/a-successful-git-branching-model/>
<https://www.atlassian.com/fr/git/workflows#!workflow-gitflow>

A la fin de chaque Sprint, la nouvelle release sera produite, avec les nouvelles fonctionnalités validées lors de la review de Sprint. La version finale du projet sera l'ultime release, et devra constituer le produit fini qui sera évalué par l'équipe enseignante.

Si les étudiants souhaitent malgré tout améliorer leur produit après la date de remise, ils doivent alors en parler avec l'équipe enseignante et effectuer ces changements « bonus » sur une autre branche, et produire éventuellement une release « corrigée » clairement distincte de celle précédemment remise.

Timesheets

Il est demandé aux étudiants de tenir soigneusement compte du temps passé sur le projet. Cette information leur servira pour améliorer leur processus de fonctionnement et leur capacité de planification, mais également dans le cadre des activités entrepreneuriales proposées en parallèle au projet, et notamment pour l'analyse des coûts de développement afin d'estimer la rentabilité du produit.

Les étudiants sont libres du choix d'un outil de timesheets :

- Feuille Excel
- Toggl : <https://www.toggl.com>
- Clocking IT : <http://www.clockingit.com>
- ...

L'important est d'être conscient de l'importance de la collecte de ces données de temps de travail, non pas à des fins d'évaluation de l'efficacité de développement, mais bien pour des raisons de rentabilité et dans la perspective d'une amélioration des capacités de planification.

Il faut donc veiller, dès le début du projet, à se discipliner à remplir ces timesheets **très régulièrement et le plus honnêtement possible**.

Les données collectées par chaque membre de l'équipe serviront pour le rapport final, afin de faire le bilan sur la rentabilité du produit développé.

5. Deadlines et livrables

5.1. Deadlines en fin de phase d'initialisation

Une première série de livrables est à produire durant le Sprint 1. Détaillé plus haut.

5.2. Deadlines de fin de sprint

A chaque fin de Sprint, chaque groupe publiera sur sa page de groupe et remettra en main propre à son « coach » le rapport de sprint. Détaillé plus haut.

5.3. Deadline finale

Le rapport est à remettre avant la défense technique (modalités précisées ultérieurement).

Les livrables à remettre sont les suivants :

Sur la plateforme de développement (Github) :

1. *Le produit fini*

Le produit développé durant le projet doit être soit directement utilisable (ex : URL publique), soit très facilement déployable sur les machines de l'Ephéc moyennant un packaging ad-hoc et un mode d'emploi d'installation détaillé. Il doit être téléchargeable depuis la plateforme de gestion de développement de logiciel choisie par l'équipe (Release sur Github). Le lien vers le site du produit doit avoir été indiqué sur la page du groupe dès le début du projet.

Le code source du projet doit être disponible conjointement au produit fini, toujours sur la plateforme de développement. Il doit être propre, respecter une convention de codage, dûment documenté et respecter les principes DRY (Don't Repeat Yourself) et KISS (Keep It Simple)

2. *La documentation*

La documentation pour les utilisateurs, rédigée dans un français correct et en fonction du public-cible visé, doit être jointe au produit au format TXT sur la plateforme de gestion de développement de l'équipe.

Sur Trello :

Le Product Backlog doit être à jour, avec chaque story classifiée selon son état (finie/en cours/à faire).

1. Le rapport

L'équipe doit fournir un document au format PDF reprenant les éléments suivants :

- La **description initiale** du produit telle que fournie dans le premier rapport
- Une **analyse du produit développé** par rapport à cette description initiale, avec une réflexion sur les différences obtenues. Le Product Owner peut se baser sur l'évolution du Backlog Produit pour mener cette discussion. L'accent sera mis sur les échanges avec le client, afin de démontrer que le développement s'est fait en phase avec les demandes de ce dernier. Les fonctionnalités présentées doivent être illustrées par des captures d'écran soigneusement choisies (ni trop, ni trop peu) et adéquatement commentées dans le texte.
- Une analyse des choix techniques réalisés dans le projet
- Une analyse des **aspects sécurité** liés au projet
- Une analyse de la **mise en pratique de Scrum et du travail d'équipe** : Qu'est ce qui vous a le plus aidé, qu'est ce qui a posé problème, comment a évolué la capacité du groupe, comment les différents rôles ont été tenus, etc. Cette section sera typiquement sous la responsabilité du ou des étudiants qui ont tenu le rôle de Scrum Master durant le projet.
- Un bilan des points forts et améliorations à apporter au projet, des difficultés rencontrées, sur les apprentissages et sur les perspectives apportées par ce projet
- Les différents schémas techniques et analyses important de votre produits, par exemple : Schémas entité-associations pour les bases de données, schémas réseaux, diagrammes de classe, schémas logiques et les plans des cartes (PCB) en électronique, datasheets et explications sur l'utilisation de composants particuliers, explication sur le fonctionnement des algorithmes existants utilisés traitement de signal (input, output et grandes étapes de transformations appliquées), API développée, approche sécurité, jeux de tests effectuées, intégration continue du projet, références de solutions choisies, etc...

Les étudiants veilleront à ce que le document soit de qualité professionnelle, rédigé dans un français correct utilisant le vocabulaire adéquat (public-cible = équipe enseignante), avec une mise en page soignée, et utilisant adéquatement schémas, figures et tableaux.

2. Une analyse personnelle

Chaque étudiant remet également un document d'une page ou deux détaillant la manière dont il a vécu ce projet en fonction du rôle qu'il a joué, ce qu'il a pensé des pratiques Scrum, les compétences qu'il estime avoir améliorées, les difficultés rencontrées et d'éventuelles suggestions à l'équipe enseignante concernant l'organisation du projet.

5.4. Défense du projet

Les modalités précises des défenses/jurys de projet seront précisées ultérieurement. Cette défense intègrera un volet RSE et entrepreneurial.

6. Critères d'évaluation

Plusieurs éléments sont pris en compte pour l'évaluation :

- **L'implication dans le projet.** Cette implication sera mesurée au travers de la participation aux séances (présence, intégration dans l'équipe, implication dans le travail, interaction avec les enseignants, ...) et au travers du respect des deadlines intermédiaires et des consignes pour les livrables. Cette note sera attribuée par le coach de chaque équipe, en concertation avec l'autre enseignant jouant le rôle du client.
- **La qualité de la production.** Des critères pris en compte sont les suivants :
 - Respect des consignes
 - Produit fini : Utilisabilité, ampleur, convivialité, conformité aux besoins du client, mode d'emploi ad-hoc, ...
 - Code source : Qualité, propreté, documentation, versionning, ...
 - Forme du rapport : Mise en page, style, orthographe ?
 - Contenu du rapport : Présence des éléments demandés, pertinence des analyses, qualité et justesse de l'analyse de sécurité
 - Contenu des annexes : Les schémas et annexes jointes au rapport démontrent la maîtrise des compétences travaillées au cours du cursus (schémas réseaux, base de données, UML, ...)
 - La défense du projet : Qualité des supports, pertinence du contenu par rapport au public visé, qualité de la prestation orale, réponses aux questions,
 - ...