

420-451-SF Projet Synthèse - Automne 2015

Sprint 3 – Critères d'évaluation

Pondération : 26 % de la session.

Date de remise : lundi 9 novembre avant minuit.

Consignes de remise : Backlog de l'équipe sur **Pivotal** et sprint 3 de l'équipe sur **Github** (master).

Évaluation individuelle – (6%)

Évalué par le professeur superviseur	100%
Qualité de la participation au sein de l'équipe	50 %
Quantité de travail réalisé	50 %

Évaluation d'équipe (8 %)

Évalué par le professeur superviseur	100%
Qualité du code (autant pour les tests que pour l'application). Toutes les bonnes pratiques de programmation et principes Clean Code. <ul style="list-style-type: none">• Pas de répétition• Noms significatifs (variables, classes, fonctions, etc.)• Respects d'un standard de nomenclature• Pas de « magical strings » (utilisation de T4MVC et externalisation)• Simplicité, découpage et séparation des responsabilités (en particulier pour les contrôleurs)• Etc.	40%
Application <ul style="list-style-type: none">• Architecture, dépendances, services, algorithmes• Sécurité (authentification / autorisations, Cross-Site Request Forgery, données sensibles cryptées)• BD (modèle de données)• Réusinage / « Refactoring » (revues de code/commentaires superviseur considérés)	20%
Tests unitaires Des tests unitaires doivent être écrits pour les contrôleurs, la validation des champs des « views models », les services (sauf ceux dont la dépendance ne peut être coupée, dans ce cas faire des tests d'intégration) et toutes les méthodes des classes que vous avez écrites. <ul style="list-style-type: none">• Qualité des tests et pertinence• Structure du projet de tests• Couverture• Utilisation judicieuse des techniques (mocks)	20%
Tests d'acceptation <ul style="list-style-type: none">• Qualité des tests et pertinence (happy path)• Structure du projet de tests d'acceptation et projet « automation »• Utilisation judicieuse des outils (bddfy et seleno)• Utilisation judicieuse des techniques (Page Object et UiComponents)	20%

Intégration continue et déploiement – Évaluation d'équipe (2 %)

Évalué par le professeur superviseur	100 %
<ul style="list-style-type: none">Utilisation judicieuse de GitHub (exemples : gestion des branches, pull request, commentaires pertinents)Interventions pertinentes et rapides lorsque Jenkins échoueEnvironnement <i>staging</i> et <i>production</i> toujours fonctionnel	100%

Organisation de l'équipe – Évaluation d'équipe (5 %)

Évalué par le professeur superviseur	100 %
<ul style="list-style-type: none">Cérémonies d'équipe (Scrum)Planifications sprint, rencontres quotidiennes, rétrospectives, etc.Rythme de développement soutenu et soutenable – VitesseDéveloppement Agile (collaboration, code collectif, revue de code...)	100%

Conception / rétroaction du client – Évaluation d'équipe (5 %)

Évalué par le professeur client	100 %
<ul style="list-style-type: none">User stories : description et critères d'acceptation (analyse)	25 %
<ul style="list-style-type: none">Fonctionnalités livrées : respect des priorités et des critères d'acceptation	25 %
<ul style="list-style-type: none">Expérience utilisateur sur le site	25 %
<ul style="list-style-type: none">Communication avec le client (professionnelle et suffisante)	25 %

FRANÇAIS

Les erreurs de français seront pénalisées conformément à la PEA.

Se référer au plan de cours pour plus de détails.