

# PROJET D'INTEGRATION 2



Hiver 2015

420-616-GR

Ce cours permet l'intégration des connaissances acquises jusqu'à maintenant. Il se veut intégrateur et récapitulatif d'un ensemble important des cours précédents en ce qui concerne le développement et la programmation d'applications.

# PROJET D'INTEGRATION 2

420-616-GR

## Présentation

Ce cours permettra à l'étudiant d'effectuer les corrections de programmation sur un système de gestion existant. Il devra apporter des améliorations fonctionnelles à l'application et procéder au débogage. Il devra aussi effectuer toutes les étapes du cycle de vie du logiciel: de l'évaluation de la demande à l'implantation du système en passant par la conception logique et la réalisation technique. De plus, il devra réaliser toutes les modifications à la documentation technique ainsi qu'aux manuels d'aide à l'utilisateur.

## Cours vs Programme

Ce cours permet l'intégration des connaissances acquises jusqu'à maintenant. Il se veut intégrateur et récapitulatif d'un ensemble important des cours précédents en ce qui concerne le développement et la programmation d'applications.

## Cours vs Projet éducatif

La réalisation d'un projet de programmation permettra à l'étudiant non seulement de parfaire ses connaissances mais aussi de mettre en évidence ses goûts et ses capacités de création. Ainsi, l'étudiant sera amené à développer son autonomie, son sens de l'initiative et des responsabilités, à mettre en œuvre des actions pour aller plus haut et plus loin, à développer ses forces pour repousser ses limites et à assumer les efforts quotidiens nécessaires à sa réussite, rejoignant ainsi la mission du projet éducatif du cégep de Granby Haute-Yamaska.

## Énoncés de compétence et éléments de compétence

017A – Mettre en œuvre une application.

1. Planifier la mise en œuvre;
2. Mettre en place l'environnement;
3. Valider la qualité de la mise en œuvre;
4. Assurer le suivi de la mise en œuvre;
5. Produire le guide d'exploitation.

016Z – Assurer la production et la gestion de documents.

1. Choisir le support du document;
2. Produire un document technique et administratif;
3. Produire l'aide en ligne;
4. Produire un document de formation;
5. Diffuser un document;
6. Classer un document.

0177 – Assurer la qualité d'une application.

1. Planifier les tests;
2. Procéder à l'exécution des différents tests;

3. Valider la qualité de l'application.

0176 – Apporter des améliorations fonctionnelles à une application.

1. Analyser les caractéristiques de l'application;
2. Analyser la nature des améliorations à apporter à l'application;
3. Ajouter et modifier des fonctionnalités à une application;
4. Valider le fonctionnement de l'application.

0171 – Corriger des programmes.

1. Analyser le problème;
2. Déterminer la nature du problème;
3. Corriger le programme;
4. Valider la solution;
5. Apporter les corrections à la documentation.

## Modalités de fonctionnement

Au début de la session, un projet est présenté aux étudiants. Ce projet portera sur l'amélioration à apporter à une application existante par la création d'une nouvelle application.

La méthode utilisée pour la gestion du projet sera la méthode *Scrum*.

Le terme *Scrum* est emprunté au rugby et signifie mêlée. Ce processus s'articule en effet autour d'une équipe soudée, qui cherche à atteindre un but, comme c'est le cas en rugby pour avancer avec le ballon pendant une mêlée.

Le principe de base de *Scrum* est de focaliser l'équipe de façon itérative sur un ensemble de fonctionnalités à réaliser, dans des itérations de durée fixe, appelées sprints. Chaque sprint possède un but à atteindre, défini par le directeur de produit, à partir duquel sont choisies les fonctionnalités à implémenter dans ce sprint. Un sprint aboutit toujours sur la livraison d'un produit partiel fonctionnel.

Toute la classe collaborera ensemble pour toutes les étapes du développement.

Tout au long de la session, tous les participants devront avoir comme objectif: Réaliser un système efficace qui réponde aux besoins du client en tenant compte des critères suivants:

- la satisfaction et l'implication du client
- l'accent mis sur les capacités individuelles et l'expertise
- la collaboration, la confiance et la communication
- le développement itératif, simple et robuste
- l'intégration continue
- l'adaptation aux changements

De plus le développement devra mettre l'accent sur...

- Les individus et leurs interactions plus que les processus et les outils
- Des logiciels opérationnels plus qu'une documentation exhaustive
- La collaboration avec les clients plus que la négociation contractuelle
- L'adaptation au changement plus que le suivi d'un plan

## Modalité de participation:

Les heures de cours sont distribuées sur une base de 6 périodes par semaine pendant toute la session. Chaque étudiant doit **obligatoirement** être présent à toutes les heures de cours. De plus, vu la nature du cours, l'étudiant doit prévoir que les cours peuvent se prolonger un peu plus tard que l'heure prévue du vendredi après-midi.

Compte tenu de la nature du cours, toute absence ou retard sera pénalisé (points). Après deux absences, l'étudiant devra rencontrer le professeur afin d'évaluer la pertinence pour lui de continuer le projet. Après trois absences, c'est l'échec automatique. Veuillez noter aussi que trois retards équivalent à une absence.

Tout plagiat, tentative de plagiat ou participation à une tentative de plagiat entraîne automatiquement une note de zéro. Voici quelques exemples de fraude et de plagiat définis dans le Règlement disciplinaire sur le plagiat ou la fraude concernant les étudiants de l'Université de Montréal :

- l'utilisation totale ou partielle d'un texte d'autrui en le faisant passer pour sien ou sans indication de référence à l'occasion d'un examen, d'un travail ou d'une activité faisant l'objet d'une évaluation;
- l'exécution par une autre personne d'un travail ou d'une activité faisant l'objet d'une évaluation;
- le recours à toute aide non autorisée à l'occasion d'un examen ou pour la réalisation d'un travail;
- la présentation, sans autorisation, d'un même travail dans différents cours;
- l'obtention par moyen illicite de questions ou de réponses d'examen;
- la sollicitation, l'offre ou l'échange d'information pendant un examen.

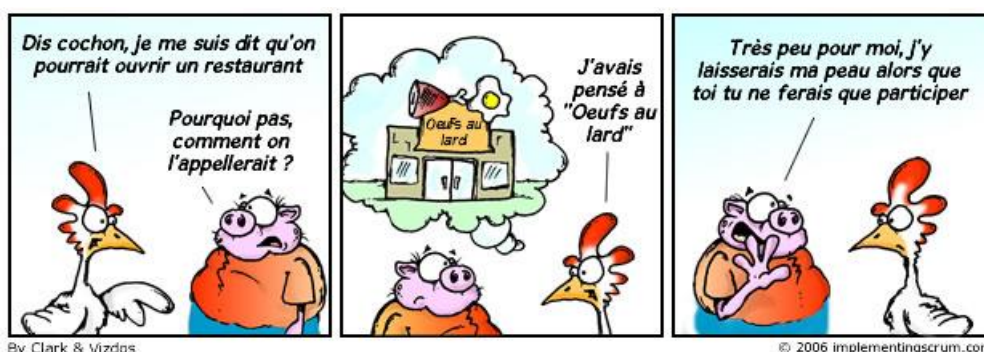
J'ajoute à cela:

- l'utilisation totale ou partielle du code (HTML, CSS, VB, etc.) d'autrui en le faisant passer pour sien ou sans indication de référence;
- utilisation d'images, de graphiques ou de logos prélevés sur un site Web;

Les manques à l'intégrité par la fraude ou le plagiat peuvent avoir des conséquences graves.

Pour tout travail, une pénalité jusqu'à concurrence de 10% peut être affectée à la note pour la qualité de la langue écrite

Aucun travail n'est fait individuellement



## Plan de cours

Étapes	Cours	Activités	Durée	Heure début
Présentation	23 jan	Fonctionnement du cours Discussion sur horaire La <b>méthode Scrum</b> et Présentation du projet Environnements technique et logique Le BackLog produit (présentation) <i>Pause</i>	30 min  60 min 30 min 30 min <i>15 min</i>	8h00  8h30 9h30 10h00 <i>10h30</i>
Sprint 0		Planification du sprint 1 (voir annexe 1) <ul style="list-style-type: none"> <li>Déterminer les stories à réaliser</li> <li>Estimer les stories (planning poker)</li> <li>Identifier les tâches</li> <li>Estimer les tâches (en heures)</li> </ul> <i>Dîner</i> Planification du sprint 1 (suite) <i>Fin de la journée</i>	60 min     <i>75 min</i> 90 min	10h45     <i>11h45</i> 13h00 14h30
Sprint 1	30 jan	Présentation de la journée Planification du sprint 1 (suite et fin) Engagement collectif à réaliser les tâches Tests et corrections <i>Dîner</i> Tests et corrections Mêlée quotidienne (voir annexe 2) <i>Fin de la journée</i>	5 min 60 min 15 min 2 h 25 m <i>75 min</i> 75 min 15 min	8h00 8h05 9h05 9h20 <i>11h45</i> 13h00 14h15 14h30
Sprint 1	6 fév	Présentation de la journée Mêlée quotidienne (voir annexe 2) Tests et corrections <i>Dîner</i> Tests et corrections Mêlée quotidienne (voir annexe 2) <i>Fin de la journée</i>	15 min 15 min 195 min <i>60 min</i> 75 min 15 min	8h00 8h15 8h30 <i>11h45</i> 12h45 14h00 14h15
Sprint 1	13 fév	Horaire à venir	300 min	8h00
Sprint 2	20 fév	Horaire à venir	300 min	8h00
Sprint 2	27 fév	Horaire à venir	300 min	8h00

## MODALITE D'EVALUATION:

À chaque rencontre hebdomadaire, lors de la mêlée quotidienne chaque membre de l'équipe fera l'objet d'une évaluation. Cette évaluation se basera sur l'atteinte des objectifs fixés à la rencontre précédente. De plus, périodiquement, des évaluations de la performance individuelle seront effectuées par l'ensemble des membres de l'équipe. Des rencontres régulières auront lieu avec chacun afin de discuter de l'évaluation et de permettre d'apporter les correctifs nécessaires s'il y a lieu.

A la fin de la session, l'ensemble de ces évaluations sera compilées afin de déterminer le résultat final de l'étudiant.

Toutefois, la base de l'évaluation sera la qualité du produit livré. En d'autres mots, le groupe obtiendra une note globale portant sur la qualité du logiciel et ses éléments. Ensuite les notes individuelles détermineront la note de chaque étudiant en appliquant un facteur de redressement positif ou négatif à la note globale.

## METHODOLOGIE

Toutes les heures de cours seront consacrées au développement du produit. Dès le début de la session, les étudiants seront introduits à la notion de l'équipe. En tant que membre de l'équipe de création, chacun devra donner le meilleur de soi-même afin d'aider l'équipe à atteindre ses objectifs.

## DISPONIBILITE:

Je serai présent toute la journée du vendredi avec l'équipe dans la classe. De plus pour le restant de la semaine, je serai toujours disponible par courriel.

## MEDIAGRAPHIE:

### Sites internet:

Comprendre Scrum en moins de 10 minutes: <http://www.thierry-pigot.fr/blog/gestion-de-projet/2010/09/27/scrum-en-moins-de-10-minutes.html>

Scrum, Agilité et Rock and roll, le blog de Claude Aubry: <http://www.aubryconseil.com/>

An Overview of Scrum for Agile Software Development:

<http://www.mountangoatsoftware.com/scrum/overview>

Le blog de Claude Aubry: <http://www.aubryconseil.com/>

### Livre:

Aubry Claude, Scrum, Le guide pratique de la méthode agile la plus populaire, Ed Dunod, 2010

Kohut Kevin, Professional Agile Development Process: Real World Development Using SCRUM, Ed Wrox, 2012

Cohn Mike, Succeeding with Agile: Software Development Using Scrum, Ed Addison Wesley Professionnal, 2009

Scwaber Ken, Beedle Mike, Agile Software Development with SCRUM, Prentice Hall, 2001

## Annexe 1: Planification du sprint

- Rappeler le contexte du sprint
  - Numéro du sprint
  - Date début et de fin
- Évaluer le périmètre potentiel
  - Déterminer les stories qui seront réalisées (de 5 à 10)
  - Estimer les stories (planning poker)
- Définir le but du sprint
  - Le but est énoncé en une phrase qui montre l'objectif principal du *sprint*.
  - Il représente la raison pour laquelle un nouveau sous-ensemble du produit est développé et sert de guide à l'équipe tout au long du sprint.
  - Exemple: authentification des utilisateurs, informatiser l'entrée des locations d'auto, etc.
- Identifier les tâches
  - Il s'agit de déterminer comment l'équipe s'organise pour réaliser les stories sélectionnées.
  - chaque story est présentée par le Product Owner et étudiée par l'équipe qui identifie les tâches nécessaires pour sa réalisation.
- Estimer les tâches
  - L'estimation du temps à passer sur une tâche est faite collectivement, par l'équipe. Il n'est pas nécessaire de passer beaucoup de temps à discuter d'une estimation : l'objectif principal est de finir une tâche (et finalement une story) pas de l'estimer.
- Prendre les tâches
  - Ce sont les membres de l'équipe qui prennent eux-mêmes les tâches. Il n'est pas utile d'aboutir à l'attribution de toutes les tâches : il suffit que chacun ait du travail pour les premiers jours du sprint ; l'affectation des autres tâches est différée, elles seront prises pendant le sprint en fonction des disponibilités des membres de l'équipe.
- S'engager collectivement
  - Pour finir la réunion, l'équipe s'engage à réaliser les stories sélectionnées. L'engagement collectif est important pour motiver l'équipe. Il peut permettre de déceler des réticences de certains, qu'il est préférable de prendre en compte avant de finir la réunion.

## Résultats

Le résultat principal est le plan de sprint, qui contient la liste des tâches avec leurs attributs, sous une forme facilement visible ou accessible.

La liste des tâches présente pour chacune les attributs suivants:

- Un nom et la description du travail à faire – Il suffit d'avoir un nom permettant d'identifier la tâche et, éventuellement, du texte collecté lors de la réunion permettant de comprendre le travail à faire.

- La story associée – À une story sont en général associées plusieurs tâches. À part les tâches dites storyless, toutes les tâches sont associées à une story.
- Le reste à faire estimé pour la tâche, en heures (ou en points) – L'estimation de l'effort nécessaire pour réaliser la tâche est faite pendant la réunion. Cela donne une première valeur, sachant que le reste à faire peut être actualisé pendant le sprint.
- La personne qui prend la tâche pour la réaliser – Une tâche peut être réalisée par une ou plusieurs personnes. Toutes les tâches ne sont pas prises à la fin de la réunion, seules le sont un petit sous-ensemble permettant à chacun d'avoir du travail pour le jour qui vient.

Note: Il ne faut pas commencer toutes les tâches dès le début du sprint. Pendant le sprint, les tâches seront prises de façon opportuniste : quand une personne finit une tâche, elle consulte la liste des tâches qui restent libres et en prend une, en tenant compte des priorités du sprint.



## Annexe 2: Mêlée Quotidienne (*Daily Scrum meeting*)

Cette réunion s'appelle standup meeting dans Extreme Programming.

Peu importe son nom, cette réunion est plutôt facile à décrire, il s'agit d'un point de rencontre où tous les membres de l'équipe répondent à trois questions simples et actualisent le plan de sprint.

Son but principal est d'optimiser la probabilité que l'équipe atteigne les objectifs du sprint. Les moyens pour atteindre ce but consistent en:

- Éliminer les obstacles nuisant à la progression de l'équipe.
- Garder l'équipe concentrée sur l'objectif du sprint.
- Évaluer l'avancement du travail pour le sprint en cours.
- Communiquer objectivement sur l'avancement.

Durée maximale: 15 minutes

Lieu: si l'équipe dispose d'un espace de travail ouvert avec un tableau mural où est affichée la liste des tâches, c'est l'endroit idéal.

Fréquence : le scrum est quotidien.

Quoi: Chaque membre de l'Équipe répond à 3 questions:

1. Présenter ce qui a été fait

Chaque participant répond à la première question :

Qu'as-tu fait depuis le dernier scrum ?

Il s'agit, pour chaque membre de l'équipe, de parler des tâches sur lesquelles il a travaillé. Pour chacune d'entre elles, il indique si la tâche est en cours ou si elle est finie.

Pour les équipes qui utilisent un tableau des tâches mural, réalisé avec des cartes (ou des Post-it), lorsque c'est le tour d'une personne de répondre à cette question, elle déplace physiquement la carte correspondant à la tâche dans la colonne « en cours » ou « fini ».

2. Prévoir ce qui va être fait

Chaque participant répond à la deuxième question :

Que prévois-tu de faire jusqu'au prochain scrum ?

Il s'agit de parler des tâches sur lesquelles il prévoit de travailler. Pour chacune d'entre elles, il indique en quoi elle consiste et s'il pense la finir dans la journée.

3. Chaque participant répond à la troisième question :

Quels sont les obstacles qui te freinent dans ton travail ?

Un obstacle empêche une tâche (ou plusieurs) de se dérouler normalement.

Avec les informations collectées sur les tâches et les obstacles, l'équipe acquiert une connaissance objective de ce qu'il lui reste à faire jusqu'à la fin du sprint. Elle peut prendre une décision sur l'ajustement du but du sprint. Il faut garder à l'esprit qu'un sprint est limité dans le temps et qu'il est toujours possible de supprimer du contenu prévu initialement mais pas de reculer la date de fin).