

# 2TP – Stage : vade-mecum & évaluation

Version 2 - 11 mai 2019

Ce document explique :

- comment effectuer le stage de la formation « Technicien en Programmation »,
- comment ce stage sera évalué

---

## 1. CONTENU

---

1. Contenu.....	1
2. Evaluation .....	2
3. Objet du stage .....	3
Thème .....	3
Développement .....	3
Storyboarding.....	3
4. Organisation du stage.....	5
Méthode Agile : SCRUM .....	5
Les rôles dans l'équipe .....	5
Backlog.....	6
Planning des « sprints ».....	7

---

## 2. EVALUATION

---

A affiner.

*Extrait du dossier pédagogique :*

**Pour atteindre le seuil de réussite, l'étudiant sera capable :**

- ◆ de respecter les termes de la convention de stage ;
- ◆ de participer aux différents travaux du métier de technicien en programmation en vue de développer son autonomie et ses capacités d'auto-évaluation ;
- ◆ de rédiger un rapport de stage :
  - ◆ décrivant le contexte professionnel au sein de l'entreprise,
  - ◆ décrivant les différentes tâches exécutées,
  - ◆ décrivant les problèmes professionnels rencontrés pendant le stage,
  - ◆ mettant en évidence l'apport de ses activités de stage dans sa formation ;
- ◆ de tenir et de compléter un carnet de stage ;
- ◆ de défendre oralement son rapport de stage.

**Pour déterminer le degré de maîtrise,** il sera tenu compte des critères suivants :

- ◆ le degré de qualité des comportements professionnels et relationnels adoptés,
- ◆ le degré d'autonomie atteint,
- ◆ le niveau de cohérence, de précision et de logique du rapport de stage,
- ◆ le niveau de pertinence du vocabulaire technique.

---

## 3. OBJET DU STAGE

---

### THÈME

---

Le stage proposé par Burotix couvre le domaine des applications web, et en particulier des applications mobiles. Les stagiaires ont pour mission de découvrir, d'appréhender et de maîtriser du mieux qu'ils le peuvent les outils nécessaires au développement d'applications web/mobiles.

L'objectif concret et ultime du stage est le développement, sous méthodologie agile, d'une application web/mobile de **quizz** et de sa **gestion**, contenant des requêtes client/serveur et une interface graphique. Les fonctionnalités demandées sont les suivantes, en convenant que l'appli s'appelle « A » et l'utilisateur « U » :

- A possède une base de données de questions avec 4 réponses possibles dont une bonne.
- A pose une question à U, choisie aléatoirement, avec les 4 réponses possibles.
- U répond, en choisissant une de ces quatre réponses, et A collecte la réponse.
- Après 10 questions, A présente le score final à U.
- U a la possibilité d'enrichir la base de données avec des nouvelles questions et ses quatre réponses, par un interface approprié proposé par A.

Pour atteindre cet objectif, l'équipe s'inspirera de la procédure suivante, non contraignante :

1. Etude du milieu des applis mobiles
2. Choix de l'architecture : standalone, client/serveur.
3. Choix de la plate-forme : langage(s), framework, base de données
4. Développement du prototype 1, contenant 2 questions
5. Développement du prototype 2, contenant 20 questions et la possibilité d'ajouter des nouvelles questions.
6. Développement du prototype 3, contenant le scoring.

### DÉVELOPPEMENT

---

L'équipe détermine la ou les plate-formes qui conviennent à son travail.

Une préférence est donnée à l'Open Source, sans être contraignante.

### STORYBOARDING

---

Les interfaces utilisateurs peuvent se concevoir sous forme de maquette en utilisant le ruban « storyboarding » de PowerPoint . Celui-ci apparaît pour peu que Visual Studio soit installé sur la machine.

Pour esquisser le GUI d'une application, le « Storyboarding » permet de dessiner une maquette de l'interface-utilisateur, sans le développement (ni XAML ni VB), mais très rapidement et très parlant. Voici un exemple de storyboard, qui peut se dessiner en moins d'une minute (en vert, les détails à observer) – à télécharger depuis [www.burotix.be](http://www.burotix.be) :

## 2TP – Stage : vade-mecum

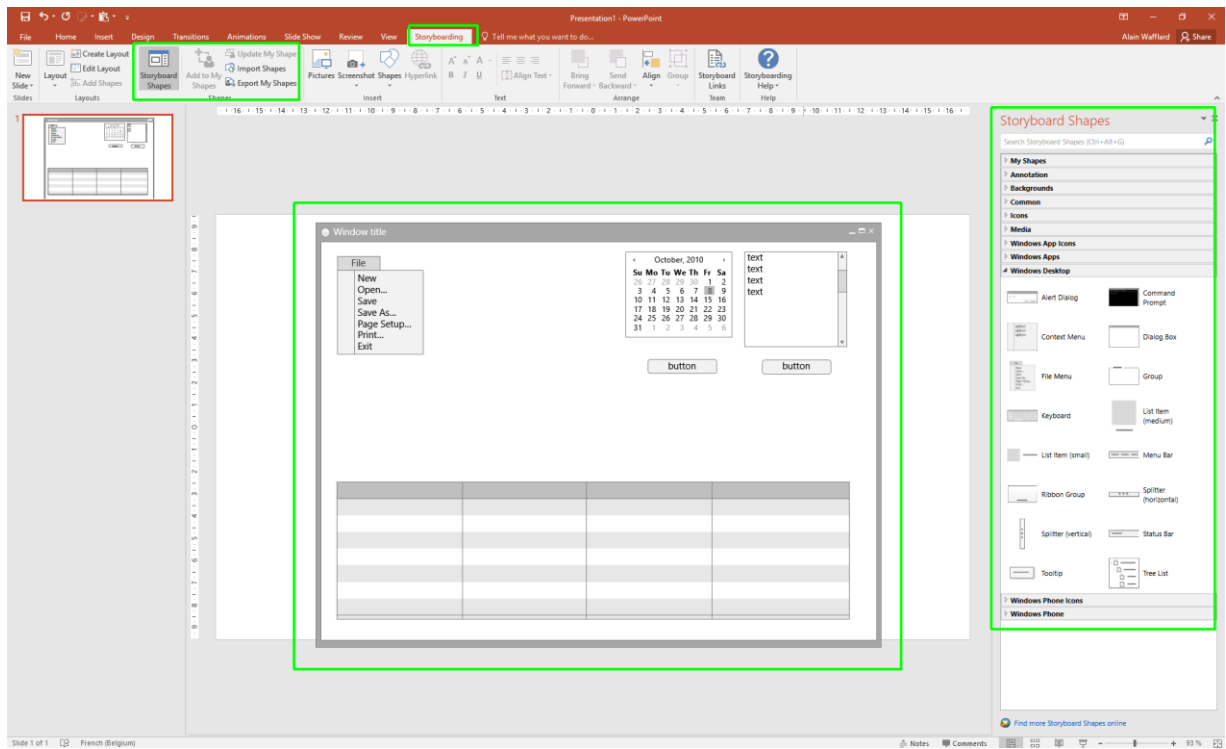


Figure 1 : Exemple de PowerPoint Storyboarding

---

## 4. ORGANISATION DU STAGE

---

### MÉTHODE AGILE : SCRUM

---

Les étudiants sont invités à relire les notes sur les méthodes de développement de produit IT, et en particulier sur les méthodes de type « **Agile** », en [www.burotix.be](http://www.burotix.be) / Support de cours / Programmation / Gestion de projet : « methodes\_de\_developpement.pdf »

Conformément à la méthode **Agile Scrum**, les démos sont planifiées chaque semaine, le mercredi après-midi, de 15h00 à 17h00.

Lisez aussi Wikipedia : Source : [https://fr.wikipedia.org/wiki/Scrum\\_\(d%C3%A9veloppement\)](https://fr.wikipedia.org/wiki/Scrum_(d%C3%A9veloppement))

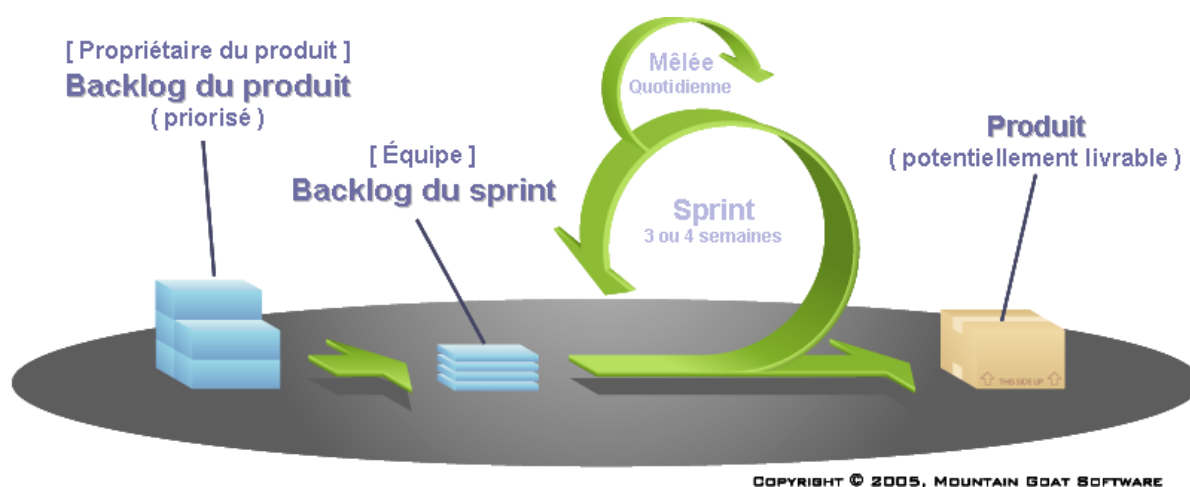


Figure 2 : SCRUM

---

### LES RÔLES DANS L'ÉQUIPE

---

Les trois étudiants travaillent en équipe, en collaboration avec le maître de stage. Les rôles sont répartis comme suit :

- **Product Owner** : Maître de stage
- **Scrum Master** : Étudiant 1 (à temps partiel) - à déterminer
- **Équipe de développement** : étudiant 1 (à temps partiel), étudiants 2 et 3 - à déterminer

Ci-dessous quelques explications sommaires sur les différents rôles.

#### Product Owner

Le **Product Owner** est le représentant des clients et des utilisateurs. Il est responsable de maximiser la valeur du produit et du travail de l'équipe de développement. Il est seul à orienter l'activité de l'équipe de développement. Ses tâches sont :

- expliciter les éléments du **backlog** (cf section suivante)
- définir l'ordre dans lequel les fonctionnalités seront développées

Il participe avec l'équipe à la définition de l'objectif du **sprint** au début de celui-ci. L'objectif peut être fonctionnel, technique ou organisationnel. Il est important qu'il reste disponible pour répondre aux questions de l'équipe et lui donner son avis sur divers aspects du produit.

### Scrum Master

Le **Scrum Master** est responsable de la compréhension, de l'adhésion et de la mise en œuvre du produit. Son autorité s'exerce sur le processus de développement (modalités des réunions, etc.), mais il ne dispose d'aucune autorité sur les autres membres de l'équipe. Ses attributions sont :

- communiquer la vision et les objectifs à l'équipe
- aider le **Product Owner** à organiser le **backlog**.
- coacher l'**équipe de développement**

### Équipe de développement

L'**équipe de développement** a pour responsabilité de livrer à chaque fin d'itération une nouvelle version de l'application enrichie de nouvelles fonctionnalités. Elle est collectivement responsable de la qualité du produit.

Elle ne comporte pas de rôles prédéfinis ; elle est habilitée à organiser et gérer son propre travail. Elle choisit la façon d'accomplir son travail, sans que ce soit imposé par une personne externe. Il n'y a pas non plus de notion de hiérarchie interne : toutes les décisions sont prises ensemble.

Elle s'adresse directement au **Product Owner** et ne prend ses instructions que de lui. Son activité est issue du carnet de produit uniquement.

## BACKLOG

L'équipe utilise le **backlog** comme outil central de communication et de suivi.

Le **backlog** est une liste ordonnée de tout ce qui pourrait être requis dans le produit et est l'unique source des besoins pour tous les changements à effectuer sur le produit. C'est un document qui évolue constamment au cours de la vie du produit. Chaque élément du **backlog** représente une fonctionnalité, un besoin, une amélioration ou un correctif.

Le **backlog**, son évolution et sa publication sont de la responsabilité du **Product Owner**, qui peut changer à discrétion l'ordre des éléments, en ajouter, modifier le découpage en éléments, modifier leur description, ou supprimer des éléments qui n'ont pas encore été réalisés par l'équipe de développement.

Le **backlog** du stage est installé sur le **cloud**. Par exemple, sous Google Keep : <https://keep.google.com/>.

Le **backlog** est en pratique un ensemble de post-it, dont la couleur signifie :

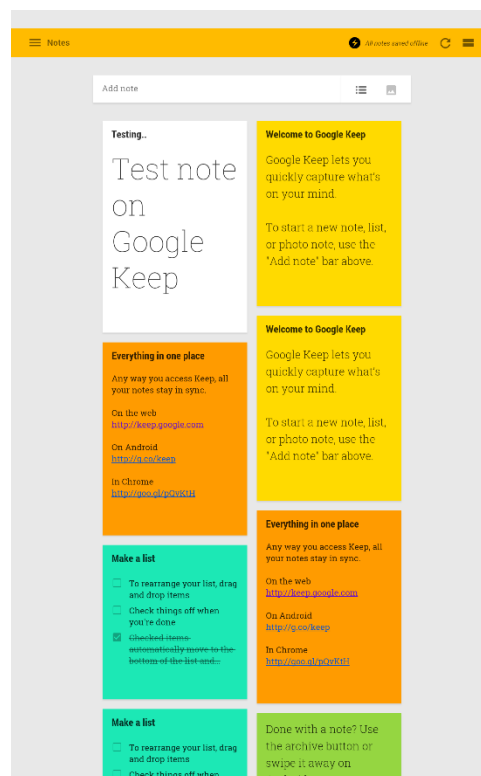


Figure 3 : Exemple de **backlog**, sous Google Keep

2TP – Stage : vade-mecum

- Rouge = à faire
- Orange = en cours de développement
- Jaune = développement terminé, à tester
- Vert = testé, tâche terminée

Le niveau de détails des fonctionnalités décrites dans le **backlog** est laissé à l'appréciation de chacun. Le mieux est le moins.

## **PLANNING DES « SPRINTS »**

---

### **Lundi 13 mai 2019 à 9h**

- Kick-off - Énoncé du projet de stage
- Intentions pour le premier sprint

### **Mercredi 15 mai 2019 à 15h**

- 1<sup>ère</sup> démo, évaluation du réalisé
- Intentions pour le sprint suivant

### **Mercredi 22 mai 2019 à 15h**

- 2<sup>e</sup> démo, évaluation du réalisé
- Intentions pour le sprint suivant

### **Mercredi 29 mai 2019 à 15h**

- 4<sup>e</sup> démo, évaluation du réalisé
- Intentions pour la finalisation

### **Vendredi 31 juin 2019 à 15h**

- 5<sup>e</sup> et dernière démo, évaluation du réalisé final

Des **scrums** (mêlées – réunions informelles, courtes et concrètes) peuvent être organisées à la demande.