

# 420-KBH-LG, PROJET D'INTÉGRATION

Saliha Yacoub

## Objectifs de cette séance:

- › Rappel :objectifs du cours
- › Dates importantes
- › Objectifs du sprint zéro ou phase Alpha
- › Contenu du document « Phase Alpha »
- › Prochaine rencontre

## Objectifs du cours:

- › Élaborer les spécifications d'un logiciel
- › Planifier et gérer un projet de développement logiciel
- › Effectuer la conception détaillée d'une application
- › Développer une stratégie de tests
- › Présenter un projet
- › Habiletés d'esprit d'équipe et d'esprit d'analyse

# Dates importantes

Dates	Évènements
Le 23 septembre	Remise du sprint zéro (Alpha)
Le 14 octobre	Remise du sprint 1
Le 04 novembre	Remise du sprint 2
Le 18 novembre	Remise du site Web du projet
Le 02 décembre	Remise du sprint 3
Le 09 décembre	Expo-info.
Chaque semaine	Rencontre avec le superviseur

## Objectifs du sprint zéro: la phase Alpha

- › Partager une vision claire du projet ·
- › Préparer l'environnement de développement·
- › Produire un backlog du produit estimé et priorisé·
- › Roder l'équipe sur le backlog initial·
- › Définir la posture ergonomique de l'interface·
- › Déterminer un plan de Release.
- › Selon le contexte, travailler sur l'architecture·
- › S'offrir une belle rétrospective

# Contenu de la phase Alpha

La remise de la phase Alpha ou du sprint zéro sera un document au format Word (Pas Excel) contenant les informations suivante:

- › Rappel de la description de votre projet.
- › Le backlog du produit détaillé, incluant les tests d'acceptation, estimé et priorisé. Les acteurs doivent-être parfaitement identifiés
- › Le modèle de conception (bases de données, interfaces, communications, architectures etc...). Chaque projet a ses spécificités. Certains projets n'ont pas de BD par exemple. Le modèle de conception peut être global (pour tout le projet) ou détaillé (pour un sprint)
- › Matériels et logiciels requis
- › Le plan de release.
- › Le backlog du premier sprint (sprint 1)
- › La planification du premier sprint (sprint 1)

# Le backlog du produit

## Le backlog ...

Le cœur de SCRUM est le *Product Backlog* ou *backlog du produit* qui représente la liste des requis priorisés. C'est un tableau de user-stories

- › SCRUM débute avec un produit backlog priorisé et estimé.
- › Chaque entrée du backlog représente une user-story décrite dans le langage et la terminologie du client. Chaque entrée sert à finaliser ce que le client désire obtenir.
- › Le Backlog doit être un document partagé, détenu par le PO. (dans votre cas, celui qui a eu l'idée)
- › Le backlog du produit est vivant.
- › Garder le **produit backlog** niveau métier. Il doit focaliser sur les buts métier et non les technologies

# Le backlog du produit

## › Ce qu'il faut faire :

- Cultiver le backlog : le backlog évolue dans le temps, il faudra le mettre à jour.
- Partager le backlog avec toute l'équipe
- Surveiller la taille du backlog : ne pas avoir plus de 150 éléments à faire dans le backlog

## › À Éviter :

- D'avoir plusieurs backlog pour le même de produit
- De ne pas avoir de backlog
- De confondre le backlog de produit avec le backlog de sprint



# Entrées du Backlog: Les user-stories

## › Définition:

- Une user-story ou un scénario est une exigence du système à développer formulée en une ou deux phrases dans le langage des utilisateurs pour servir un but.
- Sa granularité doit permettre à l'équipe de réalisation d'estimer son coût et de la réaliser entièrement à l'intérieur d'une itération (sprint)

## › Écriture narrative:

- En tant que <rôle> je veux <liste de tâches> afin de <valeur ajoutée ou résultat>

## › Exemple

- En tant que joueur, je veux connaître mon solde en écu afin de pouvoir acheter un item.

# Entrées du Backlog: Les user-stories

Une story doit respecter la propriété INVEST pour être considérée comme une bonne story.

- › **Indépendante:** lorsque le client peut en toute liberté décider de l'ordre dans lequel les scénarios (story) est implémenté sans qu'interviennent des contraintes techniques
- › **Négociable:** L'équipe de développement n'est pas contrainte par la manière dont sera implémenté le scénario : Elle a la latitude d'imaginer une solution efficace
- › **Vertical :** (V pour Valuable , a une valeur ajoutée) un bon scénario ou story décrit une fonctionnalité complète de l'application dont le client apprécie l'intérêt à l'intérieur d'une itération
- › **Estimable:** Connaitre le coût d'implémentation (en points)
- › **Suffisamment petit:** pour pouvoir l'estimer, la story doit être petite. (terminée à l'intérieur d'une itération ou d'un sprint)
- › **Testable:** la tester pour dire qu'elle est terminer → prévoir des tests d'acceptation.

# Le Backlog: Prioriser les stories

La technique utilisée pour prioriser les besoins dans un contexte itératif est celle de MoSCoW. L'avantage de la méthode *MoSCoW* réside dans la signification de l'acronyme, qui est plus compréhensible que d'autres techniques de priorisation comme élevé/moyen/faible

- › **M** pour **Must Have** : *DOIT* être fait. L'exigence est essentielle. Si elle n'est pas faite le projet échoue. On peut dire également priorité haute.
- › **S** pour **Should Have** : Il s'agit d'une exigence essentielle, qu'il faut faire dans la mesure du possible (*DEVRAIT*). Mais si elle n'est pas faite, on peut la contourner et la livrer plus tard.
- › **C** pour **Could Have**: Il s'agit d'une exigence souhaitable. Elle *POURRAIT être* faite dans la mesure où elle n'a pas d'impact sur les autres tâches
- › **W** pour **Won't Have** Il s'agit d'une exigence «Luxe». *NE SERA PAS* faite cette fois mais plus tard, mais intéressante et à garder pour la prochaine version

# Le Backlog: Estimer les stories

## Estimer les stories: le planning Poker

Pour commencer la séance de planification, il suffit de choisir une story connue de TOUS , (baseline) pour laquelle l'équipe décide en commun de lui fixer une valeur. Et, il est préférable e choisir une story de taille moyenne (3 ou 5) pour laisser une marge vers le bas et vers le haut

1. Le PO présente la story.
2. Les membre de l'équipe posent des questions pour clarifier la story.
3. Tous les participants présentent en même temps la carte choisie pour l'estimation
4. L'équipe discute des différences éventuelles entre les estimations.
5. On recommence jusqu'à une convergence des estimations.
6. On passe à la prochaine story.

Attention: **complexité ne veut pas dire plus de points. (plus long)**

# Le Backlog: Estimer les stories

Mike Cohen (un des contributeur à SCRUM) à propos de l'estimation des user-stories :

- › Prenez une équipe composée d'un chirurgien du cerveau et d'un enfant. Le backlog de produit contient les stories suivantes :
  - › En tant qu'enfant, je dois coller 1000 timbres sur 1000 enveloppes
  - › En tant que chirurgien, je dois faire une opération simple au cerveau.
- › Quelle est la story qui nécessite le plus de points ?

Il est probable que les deux stories se terminent en même temps. Que les deux stories aient besoin d'un même nombre de points pour se terminer.

Mais, si la question est : Quelle est la story la plus complexe? La réponse est claire : celle du chirurgien



CONCLUSION



QUESTIONS ??

## Prochaine séance:

- › Plan de release
- › Backlog de sprint
- › Planifier son sprint 1