

# **Projet de Génie Logiciel et méthode Agile SCRUM**

**Licence 3<sup>ème</sup> année**

## **Plateforme de Gestion d'Événements**

### Objectif

L'objectif est de concevoir et développer une plateforme web de gestion d'événements où les organisateurs peuvent créer et gérer des événements, et les participants peuvent consulter le programme, s'inscrire, et interagir autour de l'événement.

L'équipe de développement appliquera la méthode agile Scrum pour structurer le projet en sprints et favoriser une livraison progressive et itérative du produit.

### Description du Projet

La plateforme doit offrir une interface intuitive pour répondre aux besoins des organisateurs et des participants :

- Organisateurs :
  - Créer et configurer des événements (titre, description, lieu, date, capacité, programme).
  - Gérer les inscriptions et les participants.
  - Envoyer des notifications et des rappels.
- Participants :
  - Parcourir les événements disponibles.
  - S'inscrire et se désinscrire d'un événement.
  - Consulter le programme des sessions et recevoir des notifications.
  - Communiquer dans un espace de discussion dédié à chaque événement.

### Processus de Développement

Les étudiants devront suivre un cycle de développement structuré par étapes clés, en intégrant les processus de génie logiciel et en appliquant la méthode Scrum.

### Étapes de Développement et Processus Scrum

## 1. Préparation Initiale : Backlog de Produit

- Analyse des Besoins : Identifier les fonctionnalités principales de l'application en recueillant les exigences auprès d'un "client" (enseignant ou autre utilisateur).
- Création du Backlog de Produit : Décomposer le projet en fonctionnalités individuelles (user stories), en listant toutes les tâches nécessaires pour chaque fonctionnalité (inscription, notifications, création d'événements, etc.).
- Priorisation : Classer les user stories par ordre de priorité avec le Product Owner (par exemple, un enseignant qui simule ce rôle) pour définir les éléments du backlog prioritaires.

## 2. Organisation des Sprints

- Définition des Sprints : Diviser le projet en cycles de deux semaines pour chaque sprint.
- Planification du Sprint : Avant chaque sprint, le Scrum Master organise une réunion de planification pour sélectionner les user stories à réaliser, avec un objectif de sprint clair (exemple : implémentation de la fonctionnalité de création d'événements).
- Estimations : Utiliser une méthode d'estimation (par exemple, les story points) pour évaluer la charge de travail de chaque tâche.

## 3. Conception et Modélisation

- Conception de l'Architecture : Choisir une architecture en couches (MVC) pour structurer le projet.
- Modélisation des Données : Concevoir le modèle de données (utilisateurs, événements, inscriptions) et leurs relations dans une base de données.
- Design de l'Interface : Créer des maquettes de l'interface utilisateur pour valider la conception visuelle et ergonomique avec le Product Owner.
- Diagrammes UML : Créer des diagrammes de classes et de séquence pour documenter la structure et le fonctionnement de chaque module.

## 4. Développement et Réunions Quotidiennes (Daily Scrums)

- Implémentation des Fonctionnalités : Les développeurs travaillent en équipe pour coder et tester les fonctionnalités définies dans le sprint.
- Réunions Quotidiennes : Tous les jours, l'équipe se réunit pour un point de synchronisation rapide (Daily Scrum) pour évaluer l'avancement et partager les blocages éventuels.
- Tests Unitaires : Rédiger des tests unitaires pour chaque fonctionnalité implémentée afin de garantir la qualité et la fiabilité du code.

## 5. Revue de Sprint et Rétrospective

- Revue de Sprint : À la fin de chaque sprint, l'équipe présente au Product Owner le travail réalisé (démonstration des fonctionnalités développées).

- Feedback : Recueillir les retours du Product Owner et des parties prenantes pour ajuster les prochaines fonctionnalités ou priorités.
- Rétrospective de Sprint : L'équipe analyse le sprint passé, identifie ce qui a bien fonctionné et ce qui pourrait être amélioré pour les prochains sprints.

#### 6. Validation et Livraison Finale

- Tests d'Acceptation : Tester le produit final avec des utilisateurs pour recueillir des retours sur l'ergonomie et l'expérience utilisateur.
- Documentation : Rédiger la documentation technique (instructions pour l'installation, architecture du code, API, etc.) et un manuel utilisateur.
- Présentation Finale : Organiser une démonstration de l'application complète, présentant ses fonctionnalités principales et les technologies utilisées.

#### Technologies Recommandées

- Front-end : HTML, CSS, JavaScript, React ou Vue.js
- Back-end : Node.js (Express) ou Django
- Base de Données : MySQL, PostgreSQL ou MongoDB
- Outils de Collaboration : GitHub ou GitLab pour le code, Trello ou Jira pour la gestion des sprints

#### Évaluation du Projet

Le projet sera évalué sur la base des critères suivants :

1. Application de la méthode Scrum : Respect des rôles, des événements (Daily Scrums, Revue de Sprint, Rétrospective) et utilisation du backlog de produit.
2. Qualité des livrables : Analyse des besoins, spécifications, conception, et documentation.
3. Fonctionnalités et Ergonomie de l'application : Efficacité, convivialité, et capacité à répondre aux exigences définies.
4. Qualité du Code : Lisibilité, organisation, conformité aux standards de codage, et couverture de tests unitaires.
5. Présentation Finale : Qualité de la démo et clarté de la documentation fournie.

Ce projet permet aux étudiants d'acquérir de l'expérience pratique avec la méthodologie Scrum tout en appliquant les processus essentiels du génie logiciel. Ils apprennent ainsi à structurer un projet, à collaborer efficacement en équipe et à livrer un produit fonctionnel et documenté.