Je comprends que vous souhaitez développer une application web pour gérer le festival Doc à Tunis en utilisant une méthode Agile avec IceScrum. Je vais vous expliquer le projet en détail et suggérer des livrables possibles.

Le projet consiste à créer une application web pour gérer les activités du festival international de films documentaires Doc à Tunis. Voici les principales fonctionnalités à développer :

1. Gestion des films documentaires

2. Gestion des réalisateurs et producteurs

3. Planification des projections

4. Gestion du jury et des notations

5. Gestion des utilisateurs et des rôles

Pour développer ce projet de manière Agile avec IceScrum, voici comment nous pourrions procéder :

1. Définition du Product Backlog :

- Créer une liste de toutes les fonctionnalités (User Stories) nécessaires pour l'application.

- Prioriser ces fonctionnalités avec le Product Owner (probablement un responsable du festival).

2. Planification des Sprints :

- Diviser le projet en sprints de 2 à 4 semaines.

- Pour chaque sprint, sélectionner un ensemble de User Stories à réaliser.

3. Daily Scrums :

- Organiser des réunions quotidiennes courtes pour suivre l'avancement et identifier les obstacles.

4. Revues de Sprint :

- À la fin de chaque sprint, présenter les fonctionnalités développées aux parties prenantes.

5. Rétrospectives :

- Après chaque sprint, analyser ce qui a bien fonctionné et ce qui peut être amélioré.

Livrables possibles pour ce projet :

1. Product Backlog initial

2. Diagrammes UML (cas d'utilisation, classes, séquences)

3. Maquettes de l'interface utilisateur

4. Prototype fonctionnel de l'application (mise à jour à chaque sprint)

5. Documentation utilisateur

6. Documentation technique

7. Rapports de tests

8. Compte-rendu des revues de sprint et des rétrospectives

Pour modéliser ce projet en UML, voici les diagrammes que nous pourrions créer :

1. Diagramme de cas d'utilisation global :

```mermaid

graph TD

A[Responsable des inscriptions] -->|Enregistre| B(Films documentaires)

A -->|Enregistre| C(Réalisateurs et producteurs)

D[Responsable de la production] -->|Établit| E(Planning de projection)

F[Public] -->|Consulte| E

G[Membres du jury] -->|Évaluent| B

H[Président du jury] -->|Saisit les notes| I(Évaluations)

J[Administrateur] -->|Gère| K(Rôles et utilisateurs)

```

2. Diagramme de classes :

```mermaid

classDiagram

class Documentaire {

-code : String

-titre : String

-date : Date

-sujet : String

+getDetails()

}

class Personne {

-code : String

-nom : String

-prenom : String

-dateNaissance : Date

}

class Projection {

-jour : Date

-heure : Time

-lieu : String

}

class MembreJury {

-code : String

+attribuerNote()

}

class Evaluation {

-note : float

}

class Utilisateur {

-username : String

-password : String

-role : String

}

Documentaire "1" -- "1..\*" Personne : a pour réalisateur/producteur

Documentaire "1" -- "1..\*" Projection : est projeté

Documentaire "1" -- "1..\*" Evaluation : reçoit

MembreJury "1" -- "1..\*" Evaluation : donne

Personne <|-- MembreJury

Utilisateur "1" -- "1" Personne : est associé à

```

Pour le développement de l'application web, je recommande d'utiliser un framework moderne comme React ou Vue.js pour le frontend, et Node.js avec Express ou Django pour le backend. La base de données pourrait être PostgreSQL ou MongoDB, selon les besoins spécifiques du projet.

Voici un exemple de structure de projet pour une application React :

```plaintext

doc-a-tunis/

├── public/

│ └── index.html

├── src/

│ ├── components/

│ │ ├── DocumentaryList.js

│ │ ├── ProjectionSchedule.js

│ │ ├── JuryEvaluation.js

│ │ └── UserManagement.js

│ ├── pages/

│ │ ├── Home.js

│ │ ├── AdminDashboard.js

│ │ └── JuryDashboard.js

│ ├── services/

│ │ ├── api.js

│ │ └── auth.js

│ ├── App.js

│ └── index.js

├── package.json

└── README.md

```

Cette structure permet une organisation claire des composants, pages et services de l'application.

Pour utiliser efficacement IceScrum, voici quelques recommandations :

1. Créez un projet dans IceScrum et invitez tous les membres de l'équipe.

2. Définissez clairement les rôles (Product Owner, Scrum Master, développeurs).

3. Créez des User Stories détaillées pour chaque fonctionnalité.

4. Utilisez les fonctionnalités de planning poker pour estimer la complexité des tâches.

5. Suivez l'avancement du projet à l'aide des burndown charts et des tableaux Kanban.

N'hésitez pas à me poser des questions si vous avez besoin de plus de détails sur un aspect spécifique du projet ou sur l'utilisation d'IceScrum.

Bien sûr, je vais vous expliquer les diagrammes de cas d'utilisation et de classe, puis je vous donnerai les scénarios textuels correspondants.

1. Explication du diagramme de cas d'utilisation :

Le diagramme de cas d'utilisation que j'ai créé précédemment montre les principales interactions entre les acteurs (utilisateurs) et le système. Voici une explication détaillée :

- Le Responsable des inscriptions enregistre les films documentaires ainsi que les informations sur les réalisateurs et producteurs.

- Le Responsable de la production établit le planning de projection des films.

- Le Public peut consulter le planning de projection.

- Les Membres du jury évaluent les films documentaires.

- Le Président du jury saisit les notes attribuées par chaque membre.

- L'Administrateur gère les rôles et les utilisateurs du système.

2. Explication du diagramme de classe :

Le diagramme de classe représente la structure statique du système, montrant les classes, leurs attributs, et leurs relations. Voici une explication :

- La classe Documentaire contient les informations sur chaque film.

- La classe Personne est une classe générique pour les individus impliqués (réalisateurs, producteurs, membres du jury).

- La classe Projection représente une séance de projection d'un film.

- La classe MembreJury hérite de Personne et a une méthode pour attribuer une note.

- La classe Evaluation représente la note donnée à un documentaire par un membre du jury.

- La classe Utilisateur représente les comptes d'accès au système.

Les relations entre ces classes montrent comment elles interagissent (par exemple, un Documentaire est associé à une ou plusieurs Projections).

3. Scénarios textuels :

Maintenant, je vais vous fournir quelques scénarios textuels pour les principaux cas d'utilisation :

# Scénario 1 : Enregistrement d'un film documentaire

\*\*Acteur principal\*\* : Responsable des inscriptions

\*\*Préconditions\*\* : Le responsable est connecté au système

\*\*Scénario nominal\*\* :

1. Le responsable sélectionne l'option "Ajouter un nouveau documentaire"

2. Le système affiche un formulaire d'enregistrement

3. Le responsable saisit les informations du documentaire (code, titre, date, sujet)

4. Le responsable saisit les informations du réalisateur et du producteur

5. Le responsable valide l'enregistrement

6. Le système vérifie les informations

7. Le système enregistre le documentaire et affiche un message de confirmation

\*\*Extensions\*\* :

6a. Les informations sont incomplètes ou invalides

1. Le système affiche un message d'erreur

2. Retour à l'étape 3

# Scénario 2 : Établissement du planning de projection

\*\*Acteur principal\*\* : Responsable de la production

\*\*Préconditions\*\* : Le responsable est connecté au système, des documentaires sont enregistrés

\*\*Scénario nominal\*\* :

1. Le responsable sélectionne l'option "Établir le planning de projection"

2. Le système affiche la liste des documentaires sans projection planifiée

3. Pour chaque documentaire :

a. Le responsable sélectionne un documentaire

b. Le responsable saisit le jour, l'heure et le lieu de projection

c. Le système vérifie la disponibilité du créneau et du lieu

d. Le système enregistre la projection

4. Le responsable valide le planning

5. Le système publie le planning et le rend accessible au public

\*\*Extensions\*\* :

3c. Le créneau ou le lieu n'est pas disponible

1. Le système affiche un message d'erreur

2. Retour à l'étape 3b

# Scénario 3 : Évaluation d'un documentaire

\*\*Acteur principal\*\* : Président du jury

\*\*Préconditions\*\* : Le président est connecté au système, la projection du documentaire a eu lieu

\*\*Scénario nominal\*\* :

1. Le président sélectionne l'option "Évaluer un documentaire"

2. Le système affiche la liste des documentaires projetés

3. Le président sélectionne un documentaire

4. Pour chaque membre du jury :

a. Le président saisit la note attribuée

5. Le président valide les évaluations

6. Le système enregistre les notes et calcule la moyenne

\*\*Extensions\*\* :

4a. Une note est hors de la plage autorisée

1. Le système affiche un message d'erreur

2. Retour à l'étape 4a pour la note en question

Ces scénarios textuels décrivent en détail comment les principaux cas d'utilisation se déroulent, y compris les actions des utilisateurs, les réponses du système, et les potentielles exceptions ou erreurs.

Ils sont particulièrement utiles pour les développeurs qui implémentent ces fonctionnalités, ainsi que pour les testeurs qui doivent s'assurer que le système fonctionne comme prévu dans différentes situations.

N'hésitez pas à me demander si vous souhaitez des éclaircissements sur ces scénarios ou si vous voulez que je développe d'autres aspects du projet.