

# *MySpectacle*

## *Audit final*

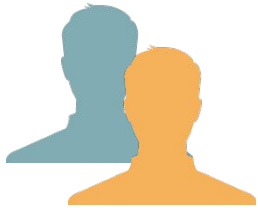
**FERREIRA Joffrey**  
**LAFRASSE Cédric**  
**TERRIER Bastien**  
**VEGREVILLE Thibaud**



# 1. Rappel du sujet

# MySpectacle

- Site d'e-commerce pour l'organisation et la réservation de spectacle
- Utilisateurs : gestionnaire, responsable, ouvreur, spectateur



Gestionnaire/responsable :

- Gestion de salle
- Visualisation des informations relatives à leurs salles



Ouvreur :

- Scan des billets (QR code)
- Faire payer une place à un spectateur



Spectateur :

- Visualisation des spectacles
- Réservation de place

# Technologies

## Applications



## Frontend



MATERIAL

## Backend



Java™

## Scripts/Déploiement



heroku



CI/CD

**Maven™**



## **2. Organisation**

# Organisation d'équipe



Sprint de 4 semaines



Daily meeting tous les mardi matin



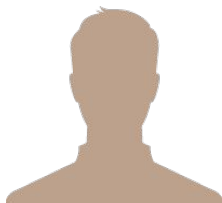
Réunions en présentiel ou via Discord

# Rôles



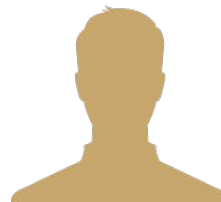
**Thibaud VEGREVILLE**  
Chef de Projet

DevOps



**Bastien TERRIER**  
Chargé communication

Full-Stack



**Joffrey FERREIRA**  
Quality Assurance

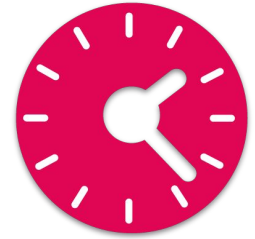
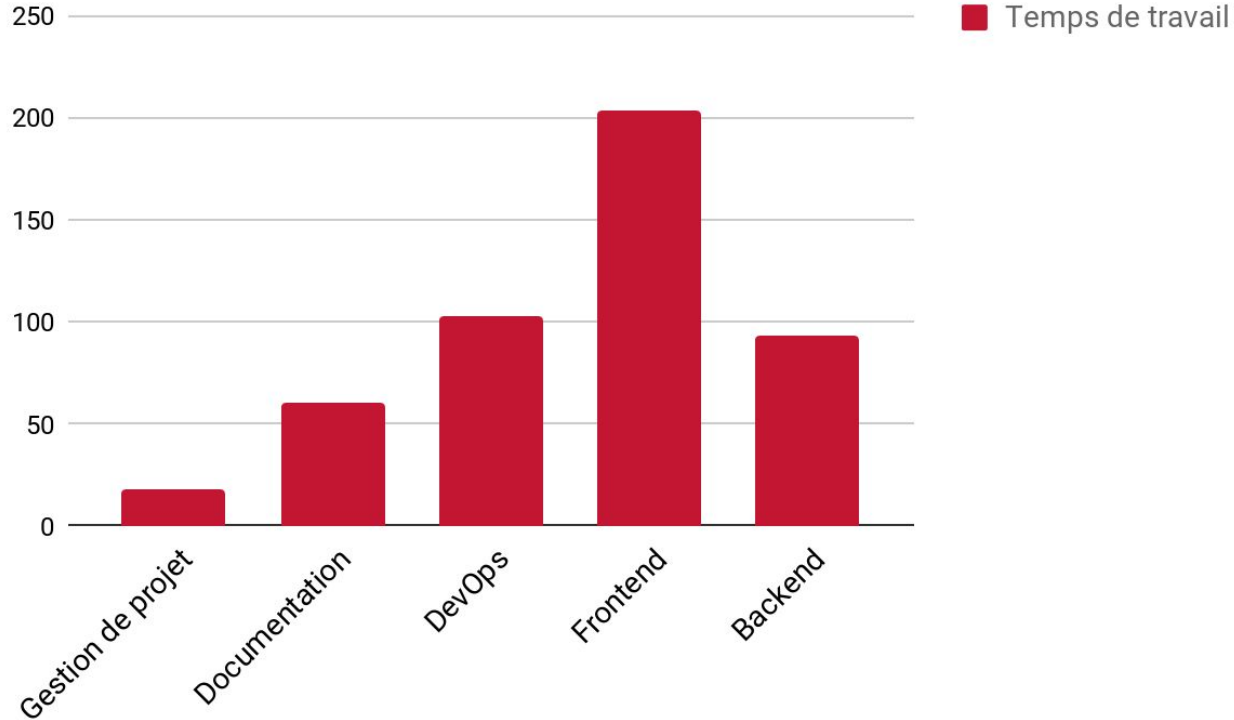
Orienté Backend - Ionic



**Cédric LAFRASSE**  
Scrum Master

Full-Stack

# Temps de travail



~ 477 heures

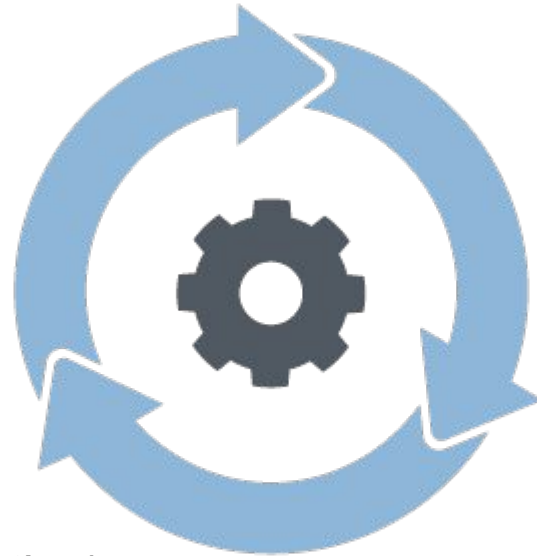




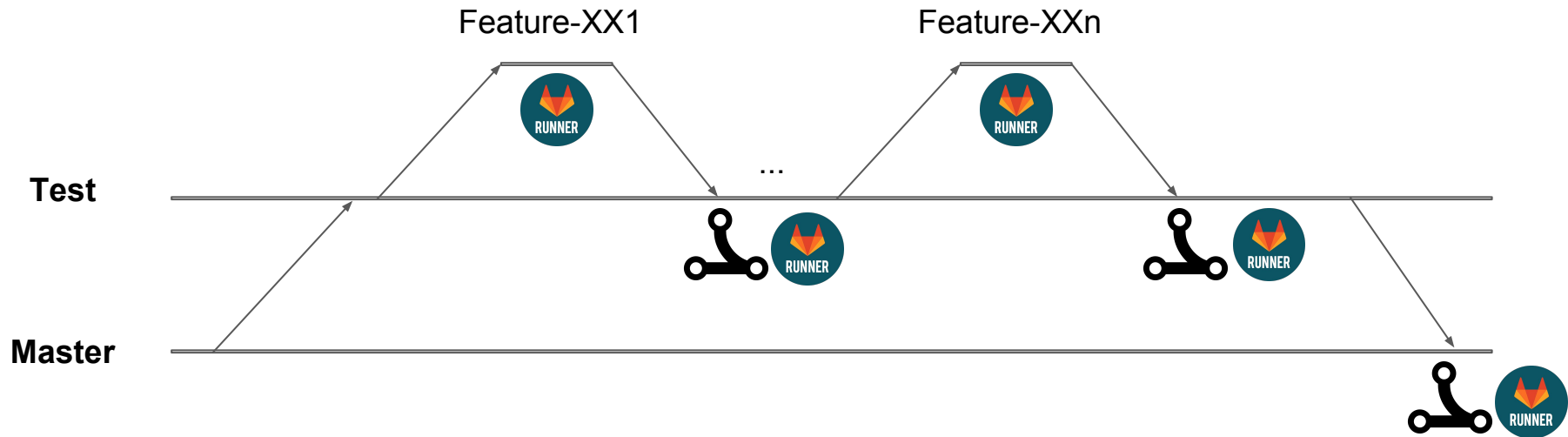
# **3. Démarche de réalisation des incréments**

# Cycle de vie logiciel

- Spécification (scénario)
- Conception
- Développement
- Tests
- Tests d'intégration
- Intégration continue
- Déploiement (déploiement en continu)



# Fonctionnement par merge request





# **4. Démarche de réalisation de Visualisation Détaillée**

# Back end

- Ajout de 2 méthodes dans l'API :
  - getSalleTauxRemplIntervalle(Long id, Date borneInfDate, Date borneSupDate, Integer intervalle) :  
*renvoie un tableau de taux de remplissage*
  - getResponsableSalle() : *renvoie la salle du responsable connecté*

## responsable-resource : Responsable Resource

Show/Hide | List Operations | Expand Operations

GET

/api/responsable/getSalle

getResponsableSalle

## salle-resource : Salle Resource

Show/Hide | List Operations | Expand Operations

GET

/api/salles

getAllSalles

POST

/api/salles

createSalle

PUT

/api/salles

updateSalle

GET

/api/salles/tauxRempl/{id}/borneInf/{borneInfDate}/borneSup/{borneSupDate}/intervalle/{intervalle}

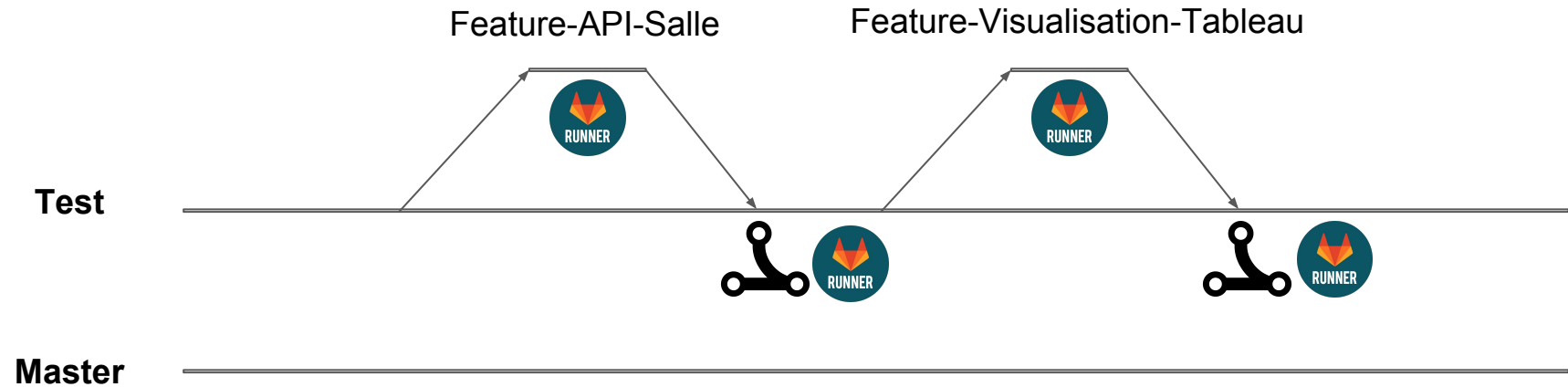
getSalleTauxRemplIntervalle

# Front end

- Ajout d'un tableau HTML dans la visualisation
- Synchronisation avec le Back End

Date	Espace 600	La Bobine	La belle électrique	L'ampérage
Décembre 2018	0	0.16666667	0.018181818	0.5
Janvier 2019	0	0	0.018181818	0
Février 2019	0	0	0.018181818	0
Mars 2019	0	0.16666667	0.018181818	0
Avril 2019	0	0	0	0

# Fonctionnement par feature





# 5. Résultats logiciel



# Bilan logiciel



Scénario 1 couvert



Scénarios 2 et 3 partiellement couverts



Retard sur le planning qui prévoyait une réalisation complète du scénario 3.



Déploiement Docker et Maven. Intégration continue avec l'outil de GitLab.



# 6. Démonstration



# 7. Résultats gestion de projet

# Gestion de projet

- Répartition des tâches basées sur les connaissances et préférences de chacun



# Code quality

- Manque de rigueur sur la régularité des rapports Sonar
- Manque de temps pour la mise en place de Gatling
- Jeu de tests
- Gitlab CI respectée





# 8. Évolutions personnelles

# Le bilan : les apports

## Frontend



- Angular 6
- Angular materials (responsive)
- chart.js

## Backend



- Spring
- API
- Java

## Déploiement



- Docker - Maven
- CI/CD
- Configuration

## Gestion de projet



- Git - Gitlab Gitlab - CI
- Qualité de code (Sonar)
- Scrum
- Travail en équipe

# Le bilan : les difficultés

- Service de déploiement sans carte bancaire
- Choix trop tardif sur le déploiement (régressions)
- Gestion du temps
- Gitlab Gricad (runners mal configurés pour la CI)
- Peu de connaissances sur Maven et JHipster en début de projet
- Limites physiques des machines

***Merci de votre attention***

***Des questions ?***