




MA & TMA

Partie 5





Programme d'aujourd'hui

- ◇ Retour sur les cours précédent cours (From scratch VS TMA)
- ◇ Rappel / présentation de la méthodologie Agile
- ◇ Rappel / présentation du testing
- ◇ Rappel / présentation du monitoring
- ◇ Développement d'une nouvelle version du projet
- ◇ Mise en œuvre de l'agilité, du testing et du monitoring dans le cadre de ce nouveau projet




From scratch VS TMA

- ◆ From scratch est tentant car il y a une maîtrise du code
- ◆ Si le travail au départ avance vite, les fonctionnalités avancées peuvent être aussi longue (voir plus) que si vous aviez repris le travail existant
- ◆ Pour une tierce personne, il faut prendre connaissance d'une nouvelle base de code..




Développement d'une nouvelle version du projet

- ◇ From scratch, par groupe de 2 ou 3
- ◇ Faites le projet qui aura le plus de chance d'être choisie par les autres pour le développement de la suite
 - ◇ Utilisant le framework de votre choix (Symfony, Laravel, React, React Native, Vue, Next, autre..)
 - ◇ En le présentant / transmettant de manière la plus convaincante
- ◇ Développer le maximum de fonctionnalités
- ◇ À commencer par un système d'authentification
- ◇ Couplé avec un système d'upload associé à l'utilisateur et au module choisi, accessible uniquement aux authentifiés
- ◇ Et les autres fonctionnalités qui en font une plateforme de peer-learning (gestion des accès au module, système de notation etc. cf CDC / specs)



Rappel / présentation de la méthodologie Agile

- ◇ Agile : cf cours [Jouer ce module - B3 Agile Devops / SCRUM](#)
- ◇ Les parties prenantes : Product Owner, Scrum Master
- ◇ Les cérémonies : Poker planning, Daily meeting, Retrospective
- ◇ Les outils : Trello, GIT



Mise en œuvre de l'agilité

- ◇ Utilisation de Trello & GIT
- ◇ Création de tickets pour définir une méthodologie dans le cadre de ce travail en groupe
- ◇ S'appuyer sur les documents existants
 - ◇ CDC
 - ◇ Specs
 - ◇ MLD



Rappel / présentation du monitoring

- ◇ Analyse statique du code : injections SQL, variables non initialisées, code dupliqué, mauvaises pratiques – SonarQube / SonarCloud
- ◇ Monitoring applicatif (APM) : erreurs PHP en temps réel – Sentry
- ◇ Sécurité – OWASP Dependency-Check + Snyk
- ◇ Métriques serveur – Prometheus + Grafana (ou UptimeRobot)

Rappel / présentation du testing

- ◇ Testing : TU, Fonctionnels (intégration, sécurité), E2E etc..
- ◇ Exemple sur Epsilon :
 - ◇ Test Unitaires

```
assert(isAuthorizedToJoin(0, [0,0,0,0]) === true);
```
 - ◇ Intégration :
Inscription mail existant/inexistant
 - ◇ Sécurité : Injection SQL sur le login
 - ◇ E2E : Parcours inscription complet

Barème de notation

- ◇ Authentification
 - ◇ Inscription / 3
 - ◇ Connexion / 2
 - ◇ Espace privé / 1
 - ◇ Confirmation inscription & changement de mot de passe par mail / 1
- ◇ Outils
 - ◇ Trello & GIT / 2
 - ◇ Tests / 2
 - ◇ CI/CD : GitHub actions OU Gitlab CI / 2
 - ◇ Monitoring / 2
- ◇ Modules peer-learning (exemple pour Epsilon)
 - ◇ Upload / 3
 - ◇ Système d'accès restreints aux modules / 2