DEVSECOPS POUR L'EMBARQUÉ ET L'IOT

AVANCÉ | ATELIER | RÉALISABLE EN ANGLAIS

FFCNCERCERXIO28 | PRIX : 2 550 € | DURÉE : 3 JOURS

Pauses et déjeuners offerts

PRÉSENTATION

La formation DevSecOps intègre la sécurité à chaque étape du cycle de vie informatique. Elle reprend les bases de la qualité logicielle et des processus en continu, avec une application concrète à l'embarqué et à l'IoT, tout en fournissant les outils pour faire face aux enjeux du terrain.

OBJECTIFS

- Mesurer l'intérêt d'investir dans la qualité de code et dans la maîtrise de la dette technique
- Exploiter les outils permettant de contrôler et assurer la qualité et la sécurité du code
- Mettre en œuvre les différentes étapes et les outils associés dans le processus de développement en continu

PROGRAMME

Introduction

DevSecOps, Code Quality Management
- Good Code vs. Bad Code; Coding Style
- Règles de codage ; Revue de code
- Analyse statique et dynamique de code
- Complexité
- Test unitaire
- Couverture de code

Travaux pratiques
- Coding Style (formater du code source C avec clang-format, vérifier du code source Python avec pylint)
- Exemple de revue de code
- Compiler et analyser du code source C avec clang, analyser du code source C++ avec clang-tidy et cppcheck, checking Python source code avec pylint
- Compiler et analyser avec Valgrind et ASAN
- C (mesurer la complexité cyclomatique avec cccc), Python (mesurer la complexité cyclomatique avec radon)
- Tests unitaires de code (C avec Unity, Python avec pytest)
- Couverture de code (C, Pyhton/gcovr)

DevSecOps, Inspection continue
- Dette technique : définition of done
- Inspection : installation du serveur SonarQube, paramétrage de SonarQube,
- Analyser le code source C et Python
- Comprendre les résultats d'analyse de SonarQube, Quality Gate
- Utilisation de SonarLint

Travaux pratiques
- Configurer sonar-project properties generic properties
- Configurer sonar-project.properties (paramètres spécifiques au langage C et Python), analyser du code source en C et Python (Configuring Quality Profile)
- Corriger les anomalies

DevSecOps, Intégration Continue
- Introduction à l'outil GitLab-CI : qu'est ce que la CI/CD ? CI/CD avec GitLab et external tools, GitLab-CI (GitLab's integrated CI/CD)
- GitLab-CI pour le développeur de code source : affichage des résultats de pipeline, forcer ou laisser de côté les exécutions de pipeline
- GitLab Runners
- GitLab-CI pour le développer de pipeline : introduction au langage YAML, définir des jobs, enregistrer les job artefacts, organisation des jobs dans un pipeline, définir des conditions sur l'exécution de pipeline, organiser les fichier YML, définir un système de pipelines
- GitLab-CI pour les développeurs de job : choisir un runner, hetling errors, caching job data
- DevSecOps, premier Pipeline avec GitLab-CI (cas d'un pojet en C et Python)
- GitLab en tant qu'utilisateur : explorer un projet sur GitLab, introduction à markdown, créer une « issue » et un merge request

Travaux pratiques
- Configuration de Git/Gitlab, premier pipeline, enregistrer les artefacts, multiples stages, conditional pipelines
- Configuration des caches
- Initialiser git project (côté Gitlab), initialiser git project (côté client git et Gitlab), initialiser git project (côté client git)
- Créer une « issue » et un merge request

DevSecOps, Développement Continu
- Qu'est ce qui se cache derrière le déploiement SW ?
- Process de déploiement continu
- Déploiement sur de multiples environnements
- Bonnes pratiques du déploiement

Travaux pratiques
- Vérification d'application, créer une image docker avec application, déploiement Docker, test de déploiement, déploiement Docker - Rollback

Synthèse et conclusion

PUBLIC/PRÉREQUIS

Personnes ayant une première expérience de développement ou de pilotage projet et désirant développer des architectures robustes.
Des connaissances de base du fonctionnement Unix ou Linux sont souhaitables afin de tirer pleinement profit de cette formation.

RESPONSABLE(S)

Vincent JOURDON
Directeur des Opérations chez Smile Embedded et Connected Systems. Formateur / Expert / Auditeur qualité de code - intégration continue dans des contextes embarqués et IoT.

Théo ZIMMERMANN
Enseignant-chercheur en génie logiciel pour la sûreté et la sécurité des systèmes à Télécom Paris et acteur du logiciel libre, notamment en tant que développeur cœur du logiciel Coq. Sa recherche a pour objectif de comprendre et d'améliorer les processus et outils de maintenance collaborative de logiciel, en particulier dans le contexte des écosystèmes de logiciels libres.

MODALITÉS PÉDAGOGIQUES

Formation opérationnelle et pratique avec retour d'expérience de pairs, audit clients et expertises métier.

Les pièges à éviter sont abordés afin de permettre aux apprenants de rapidement trouver leurs marques lorsqu'ils devront résoudre ces problèmes par eux-mêmes.