

Sigle : INF1763 Gr. 01**Titre : Techniques et outils professionnels de développement logiciel****Session : Automne 2025 Horaire et local****Professeure : Hounkanrin, Grâce****1. Description du cours paraissant à l'annuaire :****Objectifs**

Au terme de ce cours, l'étudiant.e sera en mesure d'appliquer des méthodes de développement agile, à l'aide d'outils modernes pour le contrôle de version, la gestion des dépendances, ainsi que les tests et le déploiement automatisé.

Contenu

Méthodes de développement et de gestion de projet agiles. Travail d'équipe, revues de code. Réingénierie (refactoring). Contrôle de version (git, github), gestion des dépendances (Maven, Gradle, etc.). Environnements de développement et de production, outils de virtualisation et conteneurs. Construction et déploiement automatisés (make, ant, etc.), gestion de la configuration. Automatisation des tests et intégration continue (Jenkins, Circle CI, etc.). Approche DevOps. Ce cours comporte des séances obligatoires de travaux dirigés (TD).

[Descriptif – Annuaire](#)

2. Objectifs spécifiques du cours :**Méthodologie et Développement Agile**

Au terme de cette section, l'étudiant.e sera capable de :

- Expliquer et appliquer les principes fondamentaux des méthodes agiles
- Mettre en œuvre un processus de développement agile en équipe
- Conduire des revues de code (Code Reviews)
- Appliquer des techniques de réingénierie (Refactoring)

Outils de Contrôle de Version avec Git et GitHub

Au terme de cette section, l'étudiant.e sera capable de :

- Maîtriser les commandes de base et avancées de Git
- Gérer des branches de développement de manière efficace
- Collaborer sur un projet en utilisant GitHub

Gestion des Dépendances

Au terme de cette section, l'étudiant.e sera capable de :

- Gérer les dépendances d'un projet avec Maven ou Gradle

Environnements de Développement et Déploiement

Au terme de cette section, l'étudiant.e sera capable de :

- Configurer et utiliser des environnements de développement virtuels
- Automatiser la construction (build) d'une application
- Mettre en place une chaîne d'intégration et de déploiement continu (CI/CD)

Tests Automatisés et Approche DevOps

Au terme de cette section, l'étudiant.e sera capable de :

- Écrire et exécuter différents types de tests automatisés
- Intégrer la culture et les pratiques DevOps

3. Stratégies pédagogiques :

Les formules pédagogiques suivantes seront utilisées :

Séances de cours en présentiel, de 3 h/semaine comprenant :

- Cours magistral avec démonstrations pratiques des outils
- Travaux dirigés (TD) obligatoires en laboratoire informatique
- Projet de développement en équipe utilisant les méthodologies agiles

Ce cours est à caractère pratique et appliqué.

Les méthodes d'apprentissage privilégiées pour la transmission des connaissances seront :

- **Apprentissage par la pratique** : manipulation directe des outils de développement
- **Pédagogie active** : travaux en équipe, revues de code collaboratives
- **Projet intégrateur** : développement d'une application complète du début à la fin
- **Séances de laboratoire** : configuration d'environnements, mise en place de pipelines CI/CD

Les attentes sont que les étudiant(e)s investissent au moins 90 heures de travail personnel en plus des 45 heures de cours et des heures de travail sur les projets d'équipe.

Équipement requis :

Les étudiant(e)s qui s'inscrivent à ce cours doivent s'assurer qu'ils ont :

- Un ordinateur portable avec au moins 8 GB de RAM
- Système d'exploitation : Windows 10/11, macOS, ou Linux
- Connexion Internet stable
- Comptes GitHub/GitLab (gratuits)
- Docker Desktop installé
- IDE au choix (IntelliJ IDEA, VS Code, Eclipse)

4. Heures de disponibilité ou modalités pour rendez-vous :

Disponible sur rendez-vous.

5. Plan détaillé du cours sur 15 semaines :

Semaine	Thèmes	Dates
1	Introduction aux méthodes agiles et formation des équipes (Séance virtuelle) <ul style="list-style-type: none">• Accueil des étudiant.es et présentation du syllabus	03-sep-2025

	<ul style="list-style-type: none"> • Introduction aux méthodologies agiles (Scrum, Kanban, XP) • Formation des équipes de développement (5 personnes) • Mise en place de l'environnement de développement • Présentation du projet de session 	
2	<p>Contrôle de version avec Git – Fondamentaux (Séance virtuelle)</p> <ul style="list-style-type: none"> • Concepts fondamentaux du contrôle de version • Installation et configuration de Git • Commandes de base : init, add, commit, status, log • Workflow Git local 	10-sep-2025
3	<p>Travail collaboratif avec Git et GitHub (Séance virtuelle)</p> <ul style="list-style-type: none"> • Repositories distants : clone, push, pull, fetch • Branches et stratégies de branching (Git Flow, GitHub Flow) • Merge et résolution de conflits • Pull requests et revues de code <p>TD 1 : Configuration Git et premiers commits (15 sept-2025)</p>	17-sep-2025
4	<p>Gestion des dépendances et outils de build</p> <ul style="list-style-type: none"> • Introduction à la gestion des dépendances • Maven pour Java : POM, dépendances, lifecycle • npm/yarn pour JavaScript/Node.js • Gradle : configuration et tâches <p>TD 2 : Collaboration en équipe sur GitHub (22 sept 2025)</p>	24-sep-2025
5	<p>Environnements de développement et configuration</p> <ul style="list-style-type: none"> • Configuration d'IDE (IntelliJ IDEA, VS Code) • Variables d'environnement et fichiers de configuration • Profils de développement (dev, test, prod) • Gestion des secrets et configurations sensibles 	01-oct-2025

	TD 3 : Configuration de projets avec Maven et npm (29 sept 2025)	
6	Virtualisation et conteneurs avec Docker <ul style="list-style-type: none"> • Introduction à la virtualisation et aux conteneurs • Docker : concepts, images, conteneurs • Dockerfile : création d'images personnalisées • Docker Compose : orchestration multi-conteneurs 	08-oct-2025
7	Semaine d'études	15-oct-2025
8	Examen mi-session	22-oct-2025
9	Tests automatisés et Test-Driven Development <ul style="list-style-type: none"> • Types de tests : unitaires, intégration, fonctionnels, E2E • Frameworks de test (JUnit, Jest, Selenium) • Test-Driven Development (TDD) et Behavior-Driven Development (BDD) • Couverture de code et métriques de qualité TD 4 : Configuration d'environnements multi-profils (27 oct 2025)	29-oct-2025
10	Intégration continue (CI) avec Jenkins/GitLab CI <ul style="list-style-type: none"> • Concepts et principes de l'intégration continue • Introduction à Jenkins : installation, configuration • Pipelines CI : build, test, analyse de code • GitLab CI/CD et GitHub Actions Déploiement continu et stratégies de déploiement <ul style="list-style-type: none"> • Déploiement continu vs livraison continue • Stratégies de déploiement : blue-green, canary, rolling • Outils de déploiement automatisé • Monitoring et logging des applications TD 5 : Implémentation de tests automatisés (03 nov 2025)	05-nov-2025
11	Approche DevOps et culture collaborative	12-nov-2025

	<ul style="list-style-type: none"> • Philosophie et culture DevOps • Collaboration entre équipes Dev et Ops • Métriques DevOps : DORA metrics, MTTR, MTBF • Amélioration continue et feedback loops <p>TD 6 : Configuration d'un pipeline CI / Mise en place d'un pipeline CD complet (10 nov 2025)</p>	
12	<p>Infrastructure as Code et gestion de configuration</p> <ul style="list-style-type: none"> • Concepts d'Infrastructure as Code (IaC) • Introduction à Ansible : playbooks, inventaires • Terraform : provisioning d'infrastructure • Gestion des secrets avec Vault/Kubernetes Secrets 	19-nov-2025
13	<p>Réingénierie (Refactoring) et qualité du code</p> <ul style="list-style-type: none"> • Techniques de refactoring et code smells • Dette technique : identification et gestion • Outils d'analyse statique (SonarQube, ESLint) • Métriques de qualité et maintenabilité <p>TD 7 : Automatisation de déploiement avec Ansible (24 nov 2025)</p>	26-nov-2025
14	<p>Présentations des projets de session</p> <ul style="list-style-type: none"> • Démonstration des applications développées • Présentation des pipelines CI/CD mis en place • Évaluation par les pairs et feedback • Retour d'expérience sur les méthodologies utilisées 	03-dec-2025

15	<p>Présentations des projets de session</p> <ul style="list-style-type: none"> • Démonstration des applications développées • Présentation des pipelines CI/CD mis en place • Évaluation par les pairs et feedback • Retour d'expérience sur les méthodologies utilisées 	10-dec-2025
----	---	-------------

6. Évaluation du cours :

Outils d'évaluation	
Projet de développement agile en équipe	40%
Examen mi-session	20%
Travaux pratiques	30%
Portfolio technique et documentation	10%

Détail des évaluations :

Projet de développement agile en équipe (40%)

- Développement d'une application web complète en équipe (3-4 personnes)
- Utilisation obligatoire des outils vus en cours (Git, CI/CD, Docker, tests automatisés)
- Présentation finale avec démonstration technique
- Évaluation de la qualité du code, de l'architecture et des pratiques DevOps

Examen mi-session (20%)

- Examen théorique et pratique à la semaine 8
- Questions sur les concepts agiles, Git, gestion des dépendances, Docker
- Exercices pratiques de configuration d'outils
- QCM et questions à développement court

Travaux pratiques individuels (30%)

- TP1 : Configuration Git et collaboration (7%)
- TP2 : Gestion des dépendances et build (8%)
- TP3 : Containerisation avec Docker (8%)
- TP4 : Pipeline CI/CD complet (7%)

Portfolio technique et documentation (10%)

- Documentation technique du projet (README, architecture, déploiement)
- Réflexion personnelle sur les apprentissages

7. Politiques départementales et institutionnelles :

- [Politiques relatives à la tenue des examens](#)
- [Note sur le plagiat et les fraudes](#)
- [Politique relative à la qualité de l'expression française écrite chez les étudiants et les étudiantes de premier cycle à l'UQO](#)
- Absence aux examens : [cadre de gestion](#), [demande de reprise d'examen \(formulaire\)](#)

Tolérance **ZÉRO** en matière de violence à caractère sexuel.

Le Bureau d'intervention et de prévention en matière de harcèlement (BIPH) a pour mission d'accueillir, soutenir et guider toute personne vivant une situation de harcèlement, de discrimination ou de violence à caractère sexuel. Le BIPH oriente ses actions afin de prévenir les violences à caractère sexuel pour que nous puissions étudier, travailler et s'épanouir dans un milieu sain et sécuritaire.

Vous vivez ou êtes une personne témoin d'une situation de violence à caractère sexuel ? Vous êtes une personne membre de la communauté étudiante ou une personne membre du personnel, autant à Gatineau qu'à Ripon et St-Jérôme, l'équipe du BIPH est là pour vous, sans jugement et en toute confidentialité.

Ensemble, participons à une culture de respect.

Pour de plus amples renseignements consultez [UQO.ca/biph](https://uqo.ca/biph) ou écrivez-nous au Biph@uqo.ca

8. Principales références :

Livres principaux :

- HUMBLE, Jez, FARLEY, David. Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation. Addison-Wesley Professional, 2010 (ISBN 978-0321601919).
- KIM, Gene, DEBOIS, Patrick, WILLIS, John, HUMBLE, Jez. The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations. IT Revolution Press, 2016 (ISBN 978-1942788003).
- CHACON, Scott, STRAUB, Ben. Pro Git. 2e édition, Apress, 2014 (ISBN 978-1484200773). Disponible gratuitement en ligne : <https://git-scm.com/book>
- MARTIN, Robert C. Clean Code: A Handbook of Agile Software Craftsmanship. Prentice Hall, 2008 (ISBN 978-0132350884).

Ressources techniques et documentation :

- FOWLER, Martin, BECK, Kent. Refactoring: Improving the Design of Existing Code. 2e édition, Addison-Wesley Professional, 2018 (ISBN 978-0134757599).
- NEWMAN, Sam. Building Microservices: Designing Fine-Grained Systems. 2e édition, O'Reilly Media, 2021 (ISBN 978-1492034025).
- BURNS, Brendan, BEDA, Joe. Kubernetes: Up and Running. 3e édition, O'Reilly Media, 2022 (ISBN 978-1098110208).

Documentation officielle et ressources en ligne :

- Git Documentation : <https://git-scm.com/doc>
- Atlassian Git Tutorials : <https://www.atlassian.com/git/tutorials>
- Docker Documentation : <https://docs.docker.com/>
- Jenkins Documentation : <https://www.jenkins.io/doc/>
- Maven Documentation : <https://maven.apache.org/guides/>
- Gradle User Manual : <https://docs.gradle.org/current/userguide/userguide.html>

Outils et logiciels requis :

- Git (système de contrôle de version)
- Docker Desktop (containerisation)
- IDE au choix : IntelliJ IDEA Community Edition, Visual Studio Code, Eclipse
- Comptes gratuits : GitHub, GitLab, Docker Hub
- Navigateur web moderne avec outils de développement

Articles et ressources complémentaires :

- Cahier de laboratoire pour la documentation des travaux pratiques
- Accès aux plateformes cloud (AWS/Azure/GCP - comptes étudiants gratuits)

9. Page Web du cours :

<https://moodle.uqo.ca>