

Culture et pratiques professionnelles du développeur moderne

Présentation générale

Ce cours explore les **valeurs, pratiques et responsabilités** qui structurent le métier de développeur aujourd'hui.

Au-delà du code, il s'agit de comprendre les **dynamiques humaines, éthiques et économiques** du numérique, et de situer sa propre pratique dans cet écosystème.

L'objectif est de permettre à chaque apprenant de construire une **culture professionnelle solide**, nourrie par la collaboration, la veille et la conscience des enjeux contemporains du développement logiciel.

Objectifs pédagogiques

À l'issue du cours, vous serez capable de :

- Concevoir et maintenir un **portfolio professionnel** cohérent et documenté.
- Contribuer à des projets **open source**, en comprenant leurs principes et leur gouvernance.
- Intégrer les **principes de responsabilité numérique** (accessibilité, sobriété, sécurité, respect des données).
- Mettre en place une **veille technologique régulière et exploitable**.
- Comprendre les **modèles économiques et culturels** du logiciel libre et de l'innovation ouverte.

Compétences visées

- Situer sa pratique dans le cadre professionnel, éthique et social du développement.
- Valoriser et documenter son travail de manière professionnelle.
- Collaborer au sein de communautés et d'écosystèmes ouverts.

- Analyser les impacts économiques, techniques et environnementaux du numérique.
- Entretenir une démarche de veille et d'amélioration continue.

Progression du cours

1. Construire son identité professionnelle

Objectif : formaliser sa trajectoire et ses compétences dans un espace public et structuré.

Contenus :

- Introduction à l'identité numérique du développeur.
- Construction du portfolio : organisation, présentation, cohérence visuelle et technique.
- Documentation claire : README, technologies utilisées, contexte des projets.
- Choix des outils et hébergement (GitHub, GitLab, site personnel, etc.).

Activités :

- Création ou amélioration du portfolio personnel.
- Rédaction de fiches-projets synthétiques et documentées.

2. Comprendre la logique du libre et de la collaboration

Objectif : découvrir les pratiques de collaboration ouvertes et les logiques économiques du logiciel libre.

Contenus :

- Principes du logiciel libre et de l'open source.
- Licences et droits d'usage : MIT, GPL, Apache, Creative Commons.
- Modèles économiques : sponsoring, open core, dual licensing, services associés.
- Gouvernance et gestion communautaire : rôles, PR, documentation, validation.

Activités :

- Étude de cas : analyse de la structure et du modèle d'un projet open source.
- Proposition d'une petite amélioration documentaire ou technique à un projet public.

3. Mettre en place une veille technologique efficace

Objectif : apprendre à suivre l'évolution du secteur et à capitaliser ses découvertes.

Contenus :

- Outils de veille : agrégateurs RSS, newsletters techniques, plateformes de partage.
- Organisation de la veille : tri, classement, annotation, archivage.
- Sources fiables et repérage des tendances durables.
- Partage des découvertes : bonnes pratiques de diffusion et de documentation.

Activités :

- Création d'une carte de veille personnelle (via Notion, Obsidian, Raindrop, ou équivalent).
- Mise en commun de ressources et synthèse mensuelle collective.

4. Intégrer la responsabilité numérique dans la pratique

Objectif : adopter une approche consciente et responsable du développement.

Contenus :

- Inclusion et accessibilité numérique (principes WCAG, ARIA, ergonomie universelle).
- Sobriété numérique : optimisation du front-end, compression, hébergement raisonné.
- Sécurité des données et respect des utilisateurs : RGPD, sécurité by design.
- Éthique des algorithmes : biais, transparence, redevabilité.

Activités :

- Audit d'un site ou d'une application existante sous l'angle éthique et environnemental.
- Rédaction de recommandations techniques et comportementales.

5. Comprendre les logiques d'innovation ouverte

Objectif : situer la pratique du développement dans une dynamique collective et créative.

Contenus :

- Définition et enjeux de l'open innovation.
- Interaction entre entreprises, communautés et institutions.
- API publiques, jeux de données ouverts, mutualisation des connaissances.

- Exemples : initiatives publiques et collaborations inter-entreprises.

Activités :

- Recherche et présentation d'un projet d'innovation ouverte.
- Proposition d'un usage ou d'une amélioration basée sur des données ouvertes.

Livrables attendus

- **Portfolio professionnel complet** et mis à jour.
- **Fiche d'analyse d'un projet open source**, incluant licence, modèle économique et gouvernance.
- **Carte de veille technologique** personnelle et commentée.
- **Rapport d'audit éthique et écologique** d'une application ou d'un service web.
- **Synthèse sur un projet d'innovation ouverte** existant.

Évaluation du cours

Domaine évalué	Critère principal	Pondération
Portfolio	Structure, clarté, cohérence, documentation	20 %
Collaboration & open source	Qualité de l'analyse et compréhension du modèle	20 %
Veille & innovation	Régularité, pertinence, esprit critique	20 %
Responsabilité numérique	Rigueur de l'audit et pertinence des recommandations	20 %
Réflexion professionnelle	Capacité d'analyse et recul sur la pratique	20 %

Modalités pédagogiques

- Cours théoriques illustrés d'études de cas.
- Travaux individuels guidés et revus par les formateurs.

- Ateliers de recherche et d'analyse documentaire.
- Production de livrables directement exploitables dans le portfolio.

Publics et finalité

- **Apprenants** : acquérir une compréhension globale du métier de développeur et de ses responsabilités.
- **Formateurs** : accompagner la construction de la posture professionnelle et de la culture du code.
- **Responsables pédagogiques** : intégrer ce cours dans une logique de professionnalisation durable et responsable.

Philosophie du cours

Ce cours forme des développeurs conscients de leur impact, capables de relier le code à la société, la technique à la culture, et la collaboration à la responsabilité.

Il ne s'agit pas seulement d'apprendre à coder, mais d'apprendre **pourquoi** et **comment** coder mieux.