

## Identification du module

<b>Numéro de module</b>	<b>426</b>
<b>Titre</b>	Développer un logiciel avec des méthodes agiles.
<b>Compétence</b>	Appliquer des méthodes agiles pour le développement de logiciels dans les cycles de révision.
<b>Objectifs opérationnels</b>	<ol style="list-style-type: none"> <li>1 Mettre en œuvre une fonctionnalité donnée, dans le cadre d'un projet logiciel, avec une méthode agile.</li> <li>2 Réaliser et tester par étapes les fonctionnalités à l'aide de pratiques agiles dans les cycles de révision prescrits, et présenter en résumé la version logicielle.</li> <li>3 Mettre en œuvre, de manière ciblée, des échantillons de développement existants et/ou des composants logiciels testés pour résoudre le problème.</li> <li>4 Refléter les résultats et le déroulement du travail lors d'un cycle de révision, en déduire les conclusions pour la suite du déroulement.</li> <li>5 Mettre à disposition les documents du projet et les codes sources du programme dans un système de gestion des versions.</li> <li>6 Formuler de manière compréhensive le code source programme selon les conventions.</li> </ol>
<b>Domaine de compétence</b>	Ingénierie d'applications
<b>Objet</b>	Version logicielle fonctionnelle, testée et documentée.
<b>Niveau</b>	2
<b>Pré-requis</b>	Implémenter orienté objets.
<b>Nombre de leçons</b>	40
<b>Reconnaissance</b>	Certificat fédéral de capacité
<b>Version du module</b>	1.00

## Connaissances opérationnelles nécessaires

**Numéro de module**      **426**  
**Titre**                      Développer un logiciel avec des méthodes agiles.

---

**Compétence**                      Appliquer des méthodes agiles pour le développement de logiciels dans les cycles de révision.

---

### Connaissances opérationnelles nécessaires

- 1.1 Connaître une méthode agile pour le développement logiciel.
- 1.2 Connaître le contenu et la structure d'un User-Story.
- 1.3 Connaître des valeurs et des principes de méthodes agiles.
- 1.4 Connaître les limites des méthodes linéaires et agiles.
- 2.1 Connaître le déroulement et les éléments d'un cycle des versions.
- 2.2 Connaître le déroulement et le contenu d'une présentation des versions.
- 2.3 Connaître l'importance des versions fonctionnelles de logiciels.
- 2.4 Connaître les pratiques (Best Practices) du développement logiciel agile comme par ex. Design, Metapher, Refactoring, Testdrive Design, Continuous Integration, Pair Programming, Coding Standards.
- 2.5 Connaître les outils appropriés au développement agile pour les Units-Tests, System Integration, System Build, test d'acceptance, etc.
- 3.1 Connaître l'importance et la structure des échantillons significatifs de développement.
- 3.2 Connaître l'utilité de la réutilisation des composants logiciels existants.
- 3.3 Connaître un procédé pour l'évaluation de composants logiciels ainsi que pour résoudre des problèmes donnés.
- 4.1 Connaître un procédé pour refléter les résultats d'un cycle des versions.
- 4.2 Connaître des possibilités pour refléter son propre déroulement du travail dans un cycle des versions.
- 4.3 Connaître des possibilités pour optimiser la suite du procédé sur la base de la réflexion (SCRUM, ceremony, product backlog issues).
- 5.1 Connaître les avantages et utilités d'un système de gestion des versions utilisé communément.
- 5.2 Connaître les fonctionnalités et l'utilisation d'un système de gestion des versions, ainsi que les possibilités d'y intégrer divers objets.
- 5.3 Connaître les différences entre un système de gestion des versions avec accès exclusif et accès parallèle.



- 6.1 Connaître les concepts, les règles et procédures d'un Clean-Code.
- 6.2 Connaître l'importance du Refactoring pour l'amélioration intuitive de la lisibilité d'un code source programme.

---

Domaine de compétence	Ingénierie d'applications
Objet	Version logicielle fonctionnelle, testée et documentée.
Niveau	2
Pré-requis	Implémenter orienté objets.
Nombre de leçons	40
Reconnaissance	Certificat fédéral de capacité

---

Version du module	1.00
-------------------	------