



# Agile Software Engineering

Durée 3 jour(s) (AGILE-SOFT-ENG)

Soutenir l'excellence technique dans l'entreprise

Formation Certifiante



Formation officielle



## Description

La discipline du génie logiciel a évolué avec l'introduction des principes et pratiques Lean, Agile et DevOps. De nouvelles compétences et approches aident les organisations à fournir des solutions logicielles plus rapidement, de manière plus prévisible et de meilleure qualité. Durant cette formation, les participants apprennent les principes et pratiques fondamentaux qui constituent la discipline du génie logiciel agile comme les pratiques XP, le Behavior-Driven Development (BDD) et le Test-Driven Development (TDD). Ainsi les participants sauront détailler, modéliser, concevoir, mettre en œuvre, vérifier et valider des US dans un pipeline de livraison continue SAFe. On parlera qualité dans le code (abstraction, encapsulation, programmation intentionnelle) et conception (SOLID, design patterns). À l'issue de cette formation, le stagiaire doit passer la certification SAFe Agile Software Engineer en ligne sur le site de Scaled Agile. Les supports de cours sont en anglais, l'animation en français.

## Objectifs

- Définir le génie logiciel agile : ses valeurs, principes et pratiques
- Penser test first
- S'aligner au travers de BDD (Behavior-Driven Development)
- Utiliser la Modélisation Agile pour communiquer
- Concevoir par contexte
- Concevoir des applications de qualité de code et de design
- Promouvoir les tests automatisés
- Collaborer entre architecture intentionnelle et émergente
- Appliquer les principes Lean-Agile pour optimiser le flux de valeur
- Créer un plan d'actions

## Public

- Développeurs
- Testeurs
- Agents du changement (Scrum Master, coach, manager)

## Prérequis

- Avoir suivi la formation SAFe® for Team
- Connaissance pratique des processus de développement de logiciels et de matériel
- Une compréhension minimum de l'anglais pour appréhender le support, suivre les exercices proposés et passer l'examen

## Répartition

40% Théorie, 60% Pratique



## Programme

### Jour 1

- La qualité logicielle par les principes et pratiques
- Accélérer le flux de valeur
- Appliquer l'architecture intentionnelle

### Jour 2

- Penser test first
- Utiliser le potentiel de la User Story
- S'aligner au travers de BDD (Behavior-Driven Development)

### Jour 3

- Communiquer avec des Modèles
- Concevoir des applications de qualité de code et de design