

# Reducing technical debt

---

## Description

La dette technique est souvent ce qui plombe la productivité de l'entreprise d'une équipe ou de l'entreprise dans sa totalité. La capacité gérer et mitiger cette dette est un atout majeur pour les entreprises qui, en recherche de compétitivité, dépendent de plus en plus du développement de l'informatique.

Cette formation est animé par le numéro 1 mondial de la reprise de systèmes informatiques existantes, dites legacy ou patrimoniaux, afin de transmettre les techniques les plus abouties à ce sujet. Cette formation s'inscrit dans le prolongement de son livre « Working effectively with legacy code » en adressant la problématique à un autre niveau de granularité.

## Public visé

Développeurs et architectes travaillant dans des projets où le code est difficile à comprendre et à modifier.

## Prérequis

Avoir son portable avec un éditeur de texte avec coloration syntaxique, supportant au moins du code java et c#

## Objectifs

Savoir

- Identifier divers variétés de dette technique
- Evaluer la quantité de dette dans une base de code
- Construire des stratégies de design pour du refactoring de grande envergure et pour la réduction progressive des la dette
- Changer les processus de l'équipe de de l'entreprise afin de rendre le refactoring un élément de première importance

## Programme

### Jour 1

- Présentation et discussion - The History and Definition of Technical Debt
- Exercice et discussion - Exercise/Discussion - Technical Debt Identification/Assessment
- Présentation - Forces Leading to Technical Debt
- Demo/Exercice - Scratch Refactoring/Developing a Refactoring Plan
- Présentation - Dependency Breaking Practices for Test Coverage

### Jour 2

- Présentation - Finding High Cohesion Areas
- Exercice - Class Extraction without Testing
- Présentation - Team Practices for Debt Reduction
- Exercice - Feature Carding
- Présentation/exercice - Transparent Design Quality Practices
- Présentation - The Organizational Component of Technical Debt.