



CLEAN CRAFTER

FORMATION - V.2 du 26/05/2023

OBJECTIFS

- ✓ Comprendre ce que l'on entend par "Software Craftsmanship" et les pratiques qui y sont liées.
- ✓ Être capable d'appliquer les principes "Clean code" et "S.O.L.I.D" au quotidien.
- ✓ Utiliser l'Example Mapping afin de s'aligner sur les features à délivrer.
- ✓ Pratiquer le Test-Driven Development et comprendre qu'il ne s'agit pas d'une approche de test mais d'une approche de conception.
- ✓ Se sentir à l'aise pour refactorer du code, même du code legacy (sans tests).
- ✓ Comprendre l'importance des tests dans la production de logiciels et le développement.

PUBLIC CONCERNÉ

Développeurs de logiciels

PRÉ-REQUIS

- Connaître les bases de la Programmation Orientée Objet.
- Maîtriser un langage orienté objet comme Java ou C#.
- Être capable d'écrire et d'exécuter des tests unitaires.

DURÉE DE LA FORMATION ET MODALITÉS D'ORGANISATION

Durée totale : 2 jours

En présentiel ou en distanciel

Nombre de participants : 6 à 12

Horaires de la formation :

Généralement 9h – 17h30 avec une pause déjeuner.

Les horaires seront précisés aux participants en amont du cours.

MODALITÉS D'ASSISTANCE

Pédagogique et technique : Yoan Thirion, formateur et Coach Agile chez Pyxis Suisse.

Contact administratif

Carole GODEL, coordinatrice administrative de formation
Plus d'information sur : www.pyxis-suisse.ch
Contactez-nous : info@pyxis-suisse.ch

CONTENU DE LA FORMATION

Jour 1 : Clean Coder

- **Qu'est-ce que le "Software Craftsmanship" ?**
 - ✓ Le manifeste pour l'artisanat logiciel
 - ✓ Les pratiques associées
 - ✓ Le jeu "Craft challenges"
 - **Clean Code en pratiques**
 - ✓ Crappy-Driven Development kata (using mob programming)
 - ✓ Comment identifier des "code smells" et les remédier
 - ✓ Programmer's Brain: Clean Code du point de vu cognitif
 - **Les principes S.O.L.I.D**
 - ✓ Good code vs Bad Code
 - ✓ Refactoring à l'aide des principes S.O.L.I.D
 - **Test-Driven Development (T.D.D) dans la vraie vie**
 - ✓ Chicago school
 - ✓ Les 3 règles de T.D.D
 - ✓ Kata Roman Numeral
 - ✓ Pourquoi TDD n'est pas une approche de test
- ### Jour 2 : Testing at Heart
- **Les principes du "Clean Testing"**
 - ✓ Unit Test Anti-patterns et best practices
 - ✓ Nommage et propriétés des bons Unit Tests
 - ✓ Clean Tests kata
 - **Designer avec T.D.D – Xtrem T.D.D kata**
 - ✓ Mesurer la qualité de nos tests
 - Code coverage vs Code Quality
 - Mutation Testing
 - ✓ T.D.D et Design
 - Utiliser l'Exemple Mapping pour clarifier une fonctionnalité
 - Définir 1 Test List
 - ✓ Utiliser T.D.D comme "driver" de refactoring (strangler pattern)
 - **Comment refactorer du code Legacy**
 - ✓ Trip Service kata
 - ✓ Seams, Test Data Builders, Property-Based Testing, ...

MOYENS TECHNIQUES ET PEDAGOGIQUES

- **Installations et matériel requis :**
 - ✓ Tableau(s) blanc(s) / Tableau(s) papier(s).
 - ✓ Vidéoprojecteur.
 - ✓ Ordinateurs des participants (au moins 1 pour 2) avec votre IDE préféré.
 - ✓ Accès à Internet : accès libre à github / miro
- **Code katas en C# on .NET 6 et java (>=11)**
 - ✓ Les développeurs.es devraient au moins se sentir à l'aise avec un de ces langages.
 - ✓ Les participants.es pourront obtenir de l'aide pendant les exercices, mais nous n'aurons pas le temps de couvrir les bases de l'installation, de la compilation, de la syntaxe, etc.

DESCRIPTION DE LA FORMATION

Parce qu'il n'y a pas d'agilité sans excellence technique et apprentissage continu.

Dans cette formation, les participants découvriront ce qu'est le "Software Craftsmanship", l'état d'esprit associé, et les techniques / pratiques clés pour pouvoir travailler de manière itérative et incrémentale.

Cette formation vise à aider les développeur-ses à devenir meilleurs dans la conception, l'écriture et la maintenance de code.

"Porter continuellement attention à l'excellence technique et à la qualité de la conception renforce l'agilité." Manifeste Agile – Principe 9

Pyxis Suisse

rue du simplon, 37
B.P. 326, 1001 Lausanne
Courriel : info@pyxis-suisse.ch



Modus Operandi

Cette formation est un mélange de théorie et de pratique. Nous appliquerons chaque apprentissage sur des exemples concrets afin que chaque participant-e puisse avoir la capacité de comprendre par où commencer.

Puisque dans la réalité, nous sommes souvent amenés à travailler sur de l'existant et du code legacy, nous verrons également comment tester et refactorer du code dans ce contexte.

Les exercices pratiques sont disponibles en java et C# mais l'ensemble des concepts abordés sont agnostiques du langage utilisé au quotidien. Ainsi vous pourrez les utiliser quel que soit votre stack de développement (Typescript, .NET, Java, ...).

Les participants expérimenteront "l'apprentissage par la pratique" à travers différentes activités :

- Code katas en utilisant le "Pair Programming" et le "Mob Programming"
- Ateliers permettant de stimuler l'intelligence collective.

Concepts Couverts

- Software Craftsmanship / manifeste
- Extreme Programming
- Clean Code
- Principes S.O.L.I.D
- Specification by Example
- Example mapping
- Test-Driven Development
- Techniques de refactoring automatisés
- Anti-patterns de Unit Testing et bonnes pratiques associées
- Mutation Testing
- Pair / Mob Programming

Modules optionnels

En fonction de vos besoins spécifiques et des discussions que nous aurons avant la session, vous pourrez décider d'ajouter certains modules à cette formation :

- La programmation fonctionnelle démystifiée.
- Le principe "Tell Don't Ask" et la Clean Architecture expliquée.
- Garantir votre architecture logicielle dans le temps avec ArchUnit.
- Refactoring de code legacy en profondeur.
- Introduction au Domain Driven Design.
- Approval Testing où comment mettre rapidement sous tests du code legacy mais pas que...
- "Type-Driven Development" avec le "Property-Based Testing".
- Améliorer vos Code Reviews.
- Introduction au numérique responsable.

MODALITES ET DELAIS D'ACCES

L'inscription est réputée acquise lorsque le paiement est effectué, soit par virement soit par paiement en carte bancaire via le site www.pyxis-suisse.ch. Les inscriptions à la formation sont clôturées sept jours ouvrés avant le premier jour du cours.

Pyxis Suisse

rue du simplon, 37
B.P. 326, 1001 Lausanne
Courriel : info@pyxis-suisse.ch



ACCESSIBILITÉ

Notre équipe s'engage à accueillir sans discrimination les personnes en situation de handicap et à leur garantir l'égalité des droits et des chances pour accéder à l'emploi et à la formation.

Afin que nous prenions les dispositions nécessaires pour vous garantir un accueil adapté et une expérience optimale, merci de contacter Carole GODEL sur info@pyxis-suisse.ch ou par téléphone au +41 22 575 20 75.

SATISFACTION

Questionnaire d'évaluation de la satisfaction des apprenants en fin de formation.