

# CATALOGUE DE FORMATIONS

Sensibilisez vos équipes aux **méthodes agiles**, formez vos managers, faites évoluer vos collaborateurs avec nos formations certifiantes. Nos experts vous dispensent un riche catalogue de formations **pour vos évolutions de carrières**.



# Sommaire

## Culture Agile ..... 6

Comprendre la démarche Agile – 2 jours.....	11
Travailler en équipe agile – 2 jours.....	12
Devenir Scrum Master – 2 jours.....	13
Devenir Product Owner – 2 jours.....	14
Devenir un Développeur agile – 2 jours.....	15
Devenir coach agile – 2 jours + 1 jour.....	16
Management Visuel – 1 jour.....	17
Kanban – 2 jours.....	18
Devenir facilitateur – 2 jours.....	19
Manager Agile – 2 jours.....	20
Chef de projets informatique – 4 jours.....	21

## Digital ..... 23

Angular – 3 jours.....	24
PHP Initiation – 3 jours.....	25
PHP Langage Objet – 4 jours.....	26
PHP Avancé – 4 jours.....	27
Spring Boot– 3 jours.....	28
React JS – 3 jours.....	29
HTML/CSS/JS - 3 jours.....	30

## Devops ..... 31

Les fondations de DevOps – 2 jours.....	35
Docker– 3 jours.....	36
Ansible – 3 jours.....	37
Git – 1 jour.....	38
Démarrer avec Git / Docker / PHPUnit – 2 jours.....	39

## Communication ..... 40

Devenir Speaker – 2 jours.....	41
Devenir Formateur – 2 jours.....	42



# Présentation de **Conserto**



**CONCERTO** est une société innovante et dynamique qui propose des prestations de **conseil, de formation et d'expertise informatique**. Depuis la création de l'entreprise, nous proposons un projet différent : une entreprise technologique où l'humain occupe la place principale.

Nous mettons en œuvre les technologies et les méthodologies adaptées à chaque contexte pour répondre au mieux à la demande de nos clients et les accompagner dans leurs stratégies d'entreprise.

## Nos offres



## Notre modèle

Chez **CONCERTO**, nous sommes persuadés que l'on peut développer une entreprise et donner du sens en étant cohérente avec les valeurs qu'elle défend :



# Pourquoi l'Academy ?



En 2017, CONSERTO a attribué près de **4%** de sa **masse salariale** à la formation (quand l'obligation légale est de 1%).

Notamment à travers les suivis de missions, les entretiens annuels et les entretiens professionnels.

Pour nous, **former** ce n'est pas uniquement une stratégie visant à améliorer les performances globales de l'entreprise, c'est aussi un excellent moyen de répondre à leurs besoins et de développer ainsi un véritable sentiment **d'appartenance au groupe**.

Nous avons fait le choix de proposer et financer aux salariés des événements comme le Devoxx, le Devfest, l'Agile Tour, l'Agile en Seine, le Breiz'Camp, le Jug Summer Camp, VivaTech (entre autres...) mais aussi de mettre à disposition des magazines techniques en complément de la formation.

Nous impliquons également nos collaborateurs dans l'animation ou la participation à des instants techniques en interne.



Nous avons donc décidé en 2018 de structurer notre encadrement avec un centre de **formation agréé**.

Numéro datadock : 52440808144

L'objectif de cette structure est double :

- permettre à nos consultants **d'évoluer** dans les meilleures conditions.
- **partager** notre **expertise** et notre savoir faire chez nos clients.

Nous avons un espace sur le site pour être informé des formations et s'inscrire  
[www.conserto.pro/academy](http://www.conserto.pro/academy)

Pour joindre l'équipe pédagogique dédiée : [academy@conserto.pro](mailto:academy@conserto.pro)



# Les domaines couverts

L'offre formation s'articule autour de trois domaines :

## Culture Agile

- Scrum
- Kanban, Développeur Agile...)
- Innovation managérial
- Facilitation

## Digital

- Angular, React
- Java, DotNet , PHP,Symfony
- HTML, CSS, JavaScript



## DevOps

- DevOps Fondation
- Docker
- Ansible
- Git
- Kubernetes

# Nos Implantations

Avec **8 agences réparties sur le territoire**, nous garantissons une proximité géographique mais aussi culturelle avec nos clients.

Historiquement implanté à Nantes (son siège), **CONCERTO** a multiplié ses agences pour se diversifier et maîtriser ses dépendances client et sectorielle.

Notre couverture actuelle nous permet d'accéder à des référencements de grands groupes nationaux en garantissant une intervention **locale**.

 Agence <b>Nantes</b>	> Février 2013
 Agence <b>Niort</b>	> Juin 2013
 Agence <b>Montpellier</b>	> Février 2014
 Agence <b>Rennes</b>	> Juin 2014
 Agence <b>Toulouse</b>	> Janvier 2015
 Agence <b>Paris</b>	> Janvier 2016
 Agence <b>Bordeaux</b>	> Mars 2018
 Agence <b>Lyon</b>	> Mai 2018



# Culture Agile

Actuellement les entreprises font face à 2 challenges : la progression du numérique et l'accélération de l'évolution des marchés.

De ces besoins d'adaptation permanente et d'occupation rapide de niche commerciale, est née **l'agilité**.

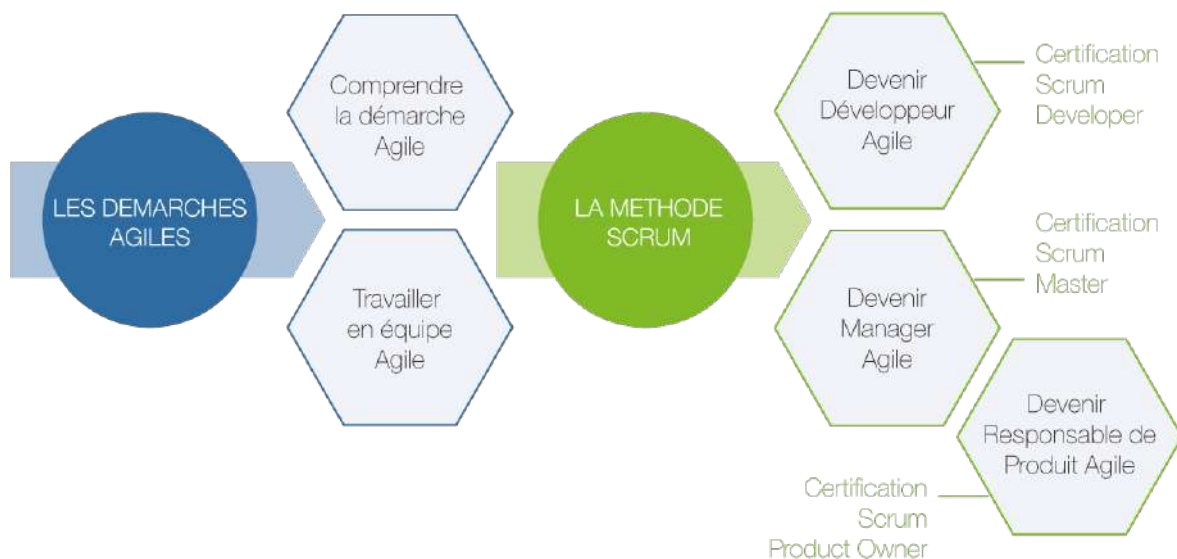
L'agilité est un état d'esprit qui, appliqué à notre manière de concevoir et réaliser nos projets et nos produits, intègre les interactions entre les personnes, l'adaptation aux changements, la collaboration avec le client et la livraison rapide d'un produit fonctionnel.

La méthode agile la plus utilisée en Europe et dans le monde est actuellement **SCRUM**. Son implémentation et son suivi permet de responsabiliser les personnes, d'auto-organiser les équipes polyvalentes et de soutenir l'amélioration continue. Autrefois cantonnée à la gestion de projets dans le domaine de l'informatique, désormais cette méthode est adoptée par de nombreuses équipes y compris dans le domaine de la comptabilité/gestion, commerciale, marketing, ressources humaines et fonctions transverses.

L'avènement de ces méthodes collaboratives, participatives et prônant en partie l'auto-organisation a bouleversé les méthodes de management. Dans ce contexte, le **Management 3.0** élaboré par Jurgen Appelo est une nouvelle approche qui s'appuie justement sur la collaboration, l'auto-organisation et la responsabilisation afin de faire émerger les forces individuelles et collectives de nos équipes.

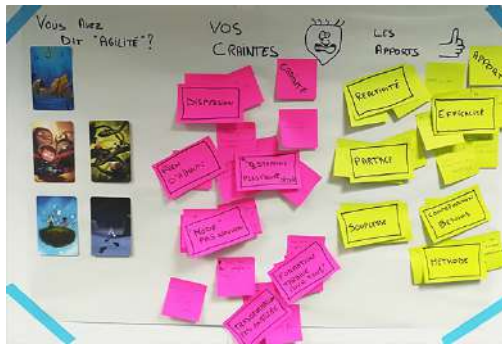
Hors contexte « gestion projet », l'adoption de certaines pratiques et état d'esprit de ces méthodes agiles et du management 3.0 permettent d'instaurer un environnement de travail combinant **efficience des équipes** et **bien-être au travail**.

## Cursus proposé



# Moyens pédagogiques

## En début de formation



Afin de d'illustrer certains concepts agiles, le formateur met en place un backlog de la formation sous la forme d'un kanban en début de formation.

Le backlog sera ensuite utilisé durant le déroulement de la session afin de suivre l'état d'avancement de la formation conformément au plan de cours et la vérification des attentes et objectifs par rapport aux attentes définis en début de formation.

## Pendant la formation

Les formations sont illustrées par des **jeux sérieux** permettant de déduire ou comprendre les fondements de l'agilité et des techniques utilisés.

Les formations sont ponctuées à la fin de chaque chapitre de questions de type QCM permettant d'évaluer les acquis des stagiaires tout au long de la formation.

## En fin de formation

À la fin de chaque stage, une mini rétrospective de la formation sera effectuée.

## Après la formation

Une version électronique du support en plus de la version papier ainsi que des exercices de type QCM sont envoyées à la fin de la formation par le formateur. La conception des supports est effectuée par les formateurs de CONCERTO ACADEMY.

## Passage de la certification scrum.org

Pour toutes les formations, un tableau de bord est mis à jour systématiquement synthétisant les évaluations de la formation par les stagiaires. Pour les formations certifiantes, un indicateur de passage à la certification est également tenu à jour.

Le taux de réussite à l'examen *scrum.org* avec les formateurs de **CONCERTO ACADEMY** est proche de 99%.

## Coaching après formation

Le formateur reste disponible pendant 30 jours après la formation pour aider le stagiaire à appliquer ses connaissances nouvellement acquises, et offre des conseils sur des applications spécifiques.

## Exemple de serious game

Chaque concept de l'agilité est appuyé par un ou plusieurs serious games.

Ces « serious games » sont de véritables jeux de rôle mettant les stagiaires en équipe en face de situations concrètes d'un projet et permettant de s'exercer sur le savoir-faire mais également sur le savoir être « Agile ».





### Dessine-moi une vache !

Les stagiaires se répartissent en deux groupes.

Chaque groupe reçoit un texte décrivant un dessin à réaliser en 3 minutes. Il s'agit de dessiner une prairie durant une belle journée d'été

Le texte pour le premier groupe est très détaillé (spécification détaillée) alors que l'autre texte donne une vision de ce qu'il faut dessiner (=user story).

Au résultat deux types de dessin

Ce jeu aborde la communication écrite vs. la communication orale, la documentation exhaustive vs un produit opérationnel, l'atomisation de l'expression des besoins...

### Lego4Scrum

Le but du jeu est de construire une ville en lego en utilisant le process scrum.

Cet atelier bien connu des agilistes est un exercice très complet qui permet d'aborder les concepts d'itération, d'incrément, de définition de terminé, d'estimation par point avec le planning poker, le burndown



### Le bâton d'Helium

Les stagiaires sont alignés pour former deux rangées. Le formateur dépose un bâton fait de papier sur l'index de chaque stagiaire. Le but étant de baisser ce bâton à terre.

Lors du debrief, les notions de communication efficace, de coopération, de patience et de gestion de petites contributions des autres dans la réalisation de grands résultats prometteurs sont abordées

### Rétrospective

À la fin du stage, le formateur peut mettre en pratique une technique d'animation de rétrospective comme « l'étoile de mer ».

Le sujet peut porter sur la formation ou bien lors d'une intra sur le fonctionnement du service. C'est un exercice qui permet aux stagiaires de se projeter à court terme sur la mise en œuvre des pratiques apprises







### Le guide touristique

Jeu de simulation qui consiste à créer un guide touristique papier en 2 ou 3 itérations.

Chaque stagiaire se voit attribuer un rôle scrum à chaque itération.

Ce jeu, fil rouge de la formation, permet d'aborder tous les concepts de scrum.

Il permet également de mettre en pratique les concepts scrum dans la réalisation d'un vrai produit.

### Engagement cube

Les stagiaires ont à leur disposition des petits cubes en bois. Ils doivent s'engager sur le nombre de cubes qu'ils vont être en mesure d'empiler. Le jeu se déroule en plusieurs itérations (engagement individuel sans apprentissage, avec apprentissage et engagement collectif)

Ce jeu permet de faire constater les différentes stratégies personnelles par rapport à l'engagement et l'impact du contexte de groupe sur l'engagement de chacun.



L'agilité est un domaine de **spécialisation fort** chez CONSERTO.

Tous nos formateurs sont des coaches qui interviennent non seulement chez nos clients, mais **accompagnent** également au niveau du management, le pilotage de l'innovation et l'ensemble des transformations internes.

Dans un contexte de forte croissance, agiles par nature, nous avons fait de la culture du changement une priorité.





## Culture Agile ..... 6

Comprendre la démarche Agile – 2 jours.....	11
Travailler en équipe agile – 2 jours.....	12
Devenir Scrum Master – 2 jours.....	13
Devenir Product Owner – 2 jours.....	14
Devenir un Développeur agile – 2 jours.....	15
Devenir coach agile – 2 jours + 1 jour.....	16
Management Visuel – 1 jour.....	17
Kanban – 2 jours.....	18
Devenir facilitateur – 2 jours.....	19
Manager Agile – 2 jours.....	20
Chef de projets informatique – 4 jours.....	21



# Comprendre la démarche Agile – 2 jours

## Objectifs

- Appréhender les fondements théoriques et éthiques de l'agilité
- Comprendre les apports de l'agilité principalement l'adaptation aux changements et la réactivité

## Public

Étudiants, Managers, Fonctionnels, Chef de projet/produit, Développeurs

## Prérequis

Aucun hormis avoir une connaissance sur l'état de l'art de la gestion de projet

## Méthodes pédagogiques

Pour ancrer les apprentissages :  
Théorie 40% - Ateliers 60%

## Introduction

- Tour de table (l'arbre à personnages)
- Evolution des méthodes traditionnelles
- Le manifeste agile
- Les 12 principes de l'agilité

## Les différentes « méthodes » agiles

- Histoire des « méthodes » agiles
- Le Modern Agile
- eXtreme Programming

## Aperçu du cadre SCRUM

- Le Product Owner
- Le Scrum Master
- Les différentes maturités du Scrum Master

## Organisation et lancement

- Comment démarrer un projet en mode agile ?
- Construire le bon produit avec le lean startup et le design thinking
- Le forfait agile
- 

## Le suivi et le contrôle

- Le management visuel
- Le daily standup
- La rétrospective

## Aperçu de la méthode Kanban

- Concept de flux titré
- Goulot d'étranglement
- Kanban Vs Scrum

## Réussir un projet Agile

- Quand utiliser une démarche agile ?
- Les contraintes humaines
- Les outils.
- Les causes d'échecs.
- Les zones de risques.
- Retours d'expérience



# Travailler en équipe agile – 2 jours

## Objectifs

Appréhender les fondements théoriques et éthiques de l'agilité

Comprendre les apports de l'agilité principalement l'adaptation aux changements et la réactivité

Comprendre SCRUM et adopter les bonnes pratiques

Faire émerger la responsabilisation des personnes et l'auto-organisation

Découvrir les méthodes et pratiques du management 3.0

Se projeter sur les premiers jalons de la mise en pratique

## Public

Étudiants, Managers, Fonctionnels, Chef de projet/produit, Développeurs

## Prérequis

Aucun hormis être ouvert au travail collaboratif en équipe

## Méthodes pédagogiques

Pour ancrer les apprentissages :  
Théorie 40% - Ateliers 60%

### Module 1 : SCRUM et Les Méthodes Agiles

Partie 1 : Sensibilisation méthodes agiles

- Histoire de l'agilité
- Pourquoi les méthodes agiles
- Valeurs et principes des méthodes agiles
- Objectifs des méthodes agiles

Partie 2 : Focus sur SCRUM

- Le cycle et les cérémoniaux associés
- La notion de gestion du temps
- Les rôles dans une équipe polyvalente
- Les artefacts (Backlog)
- Suivi d'un projet SCRUM

### Module 2 : Cadrage d'un projet agile

Partie 1 : Découverte de Serious Games

- Pour accompagner l'intelligence collective
- Pour prioriser
- Pour planifier

Partie 2 : Utilisation du Management Visuel

- Principe du management visuel
- Établir son ScrumBoard

### Module 3 : Sensibilisation Management 3.0 – Management Collaboratif

Partie 1 : Découvrir le Management 3.0 et ses implications

- Introduction aux différentes formes de leadership
- Comprendre le positionnement manager/équipe

Partie 2 : Connaître et expérimenter les outils et pratiques du management 3.0

- Pour l'affectation des tâches : Delegation Poker et Delegation Board
- Pour identifier et gérer les compétences de l'équipe : Matrice des compétences
- Pour innover ensemble : Pop-Corn Flow



# Devenir Scrum Master – 2 jours

## Objectifs

- Comprendre les fondamentaux de SCRUM
- Savoir initier et conduire un projet SCRUM
- Préparer l'examen Scrum Master (PSM I)

## Public

Étudiants, Managers, Fonctionnels, Chef de projet/produit, futurs scrum master, Développeurs

## Prérequis

Idéalement avoir suivi la formation « Comprendre la démarche agile »

Avoir une connaissance sur l'état de l'art de la gestion de projet

## Méthodes pédagogiques

Pour ancrer les apprentissages :  
Théorie 40% - Ateliers 60%  
Des serious games sont joués tout au long de la formation.

La certification est passée après la formation



## Introduction à l'agilité

- Tour de table (l'arbre à personnages)
- Evolution des méthodes traditionnelles
- Le manifeste agile
- Les 12 principes de l'agilité
- Les différentes « méthodes » agiles
- 

## Aperçu du cadre SCRUM

- Le Product Owner
- Le Scrum Master
- Les différentes maturités du Scrum Master
- 

## La planification

- Le story mapping
- Le sprint review
- Définition de terminé.
- L'objectif du sprint
- La vélocité
- Les estimations agiles (planning poker...)

## Les artefacts

- Le backlog
- User stories
- Prioriser et gérer un backlog
- INVEST
- La dette technique
- Le sprint backlog
- L'incrément

## Le suivi et le contrôle

- Le management visuel
- Le burndown, burnup...
- Le daily scrum
- Le sprint review
- La rétrospective

## Scrum à grande échelle

- Ce que dit scrum
- Aperçu du nouveau cadre Nexus
- Le cas Spotify

## Préparation à l'examen (1/2j)

- Passage d'un examen blanc de 80 questions



# Devenir Product Owner – 2 jours

## Objectifs

- **Maîtriser le rôle du Product Owner au sein de Scrum**
- **Comment formaliser et entretenir un Product Backlog**
- **Prendre connaissance des outils de suivi du progrès**
- **Préparer l'examen Product Owner PSPO I**

## Public

Manager, MOA, fonctionnel, chef de projet, chef de produit, futur product owner

## Prérequis

Idéalement avoir suivi la formation « Comprendre la démarche agile »

Avoir une connaissance sur l'état de l'art de la gestion de projet

## Méthodes pédagogiques

Pour ancrer les apprentissages :  
Théorie 40% - Ateliers 60%

Des serious games sont joués tout au long de la formation.

La certification est passée après la formation



## Introduction à l'agilité

- Tour de table (l'arbre à personnages)
- Evolution des méthodes traditionnelles
- Le manifeste agile
- Les 12 principes de l'agilité
- Aperçu du cadre SCRUM
- Les rôles dans SCRUM
- Le bon et le mauvais product owner

## Pilotage par la valeur

- Qu'est-ce que la valeur ?
- Calculer le ROI et le TCO
- Les KPI
- Le modèle de Kano
- 

## La planification

- L'impact mapping
- Le story mapping
- Le sprint review
- L'objectif du sprint
- Les estimations agiles

## La gestion du backlog

- User stories
- INVEST
- Méthode OOPSI
- Organiser et gérer le backlog
- La dette technique

## Suivre un projet agile

- Le management visuel
- Le burndown, burnup...
- Le daily scrum, sprint review, rétrospective

## Le Product Owner dans l'innovation et la stratégie d'entreprise

- Empathy map
- Persona
- La value proposition canvas
- Construire le bon produit avec le lean startup et le design thinking

## Le forfait « agile » ? comment faire ?

### Préparation à l'examen (1/2j)

- Passage d'un examen blanc de 80 questions



# Devenir un Développeur agile – 2 jours

## Objectifs

- Comprendre les fondamentaux de SCRUM
- Connaître les bonnes pratiques de développement Agile
- Connaître les techniques de tests Agile
- Se préparer à l'examen PSD I

## Public

Développeur, Scrum Master

## Prérequis

Avoir suivi la formation «Comprendre la démarche agile »

Ou avoir déjà travaillé dans un projet en mode agile

## Méthodes pédagogiques

Pour ancrer les apprentissages :  
Théorie 40% - Ateliers 60%

Des serious games sont joués tout au long de la formation.

La certification est passée après la formation



## Introduction à l'agilité

- Tour de table (l'arbre à personnages)
- Evolution des méthodes traditionnelles
- Le manifeste agile
- Les 12 principes de l'agilité
- Les différentes « méthodes » agiles
- Les pratiques XP (pair programming, intégration continue...)

## Aperçu du cadre SCRUM

- Les rôles (Product Owner, Scrum Master , Equipe de développement)
- Les cérémonies (sprint planning, daily scrum, sprint review, sprint retrospective)
- Les artefacts (backlog, sprint backlog, increment)

## La planification

- Définition de terminé.
- L'objectif du sprint
- La vélocité
- Les estimations agiles (planning poker...)

## Les artefacts

- Le backlog
- Le sprint backlog
- L'incrément

## Le suivi et le contrôle

- Le management visuel
- Le burndown, burnup...
- La rétrospective

## Qualité et agilité

- Le quadrant agile
- Les Tests Unitaires
- TDD : Principes et Concept
- Les Styles de TDD
- Refactoring et Clean Code
- TDD et Code Legacy
- Les Pratiques Connexes
- Vers TDD et au-delà

## Préparation à l'examen (1/2j)

- Passage d'un examen blanc de 80 questions



# Devenir coach agile – 2 jours + 1 jour

## Objectifs

- Comprendre et adopter les postures du coach
- S'exercer sur le métier de coach dans un environnement bienveillant
- Améliorer sa capacité d'écoute

## Public

Scrum Master, Product Owner, Manager

## Prérequis

Avoir suivi une formation Scrum.

Et avoir déjà avoir une expérience significative dans un projet en mode agile en tant que Scrum Master ou Product Owner.

## Méthodes pédagogiques

Pour ancrer les apprentissages :  
Théorie 40% - Exercices 60%

Des exercices de coaching sont programmés tout au long de la formation.

Le module de base a une durée de 2 jours.

Le jour 3 est dédié à la supervision et à la pratique.

## Jour 1 et 2

### Tour de table et attentes Qu'est-ce que le coaching Agile

- Les 8 compétences du coach
- Les différentes postures du Coach
- Le cadre et la démarche du coach (RPBDC)
- Le contrat de coaching

### Les outils du coach

- Les outils projectifs (photolangage, Lego..)
- Adopter un état d'esprit +/-
- Améliorer son sens de l'écoute
- Le Host Leadership
- Solution Focus

### Le coaching d'équipe

- Le modèle de Tuckman
- Evaluer la maturité agile de l'équipe
- Conflits d'équipe
- Conflit individuel
- Favoriser la métacommunication

### Conduite du changement et difficultés

- Les changements perçus à différents niveaux
- Le modèle ADKAR

## Jour 3 (optionnel) : Supervision

La supervision permet au coach de prendre du recul, mieux vivre les situations d'accompagnement, de bénéficier d'un regard neuf d'un tiers et d'améliorer sa pratique.

### Construire son identité de coach

#### Les types de Supervision

- Le cadre et définition
- La supervision individuelle
- Le Codev

#### Cas d'études

- Retour d'expérience
- Exercice de coaching et supervision sur de cas réels.





# Management Visuel – 1 jour

## Objectifs

- Comprendre l'intérêt et le fonctionnement du management visuel
- Appréhender les différentes applications possibles de la démarche
- Comment améliorer la performance d'une équipe grâce au management visuel
- 
- Comment mettre en œuvre une telle démarche

## Public

Manager, chef de projet, scrum master, product owner, développeur...

## Prérequis

Aucun

## Méthodes pédagogiques

Pour ancrer les apprentissages :  
Théorie 30% - Ateliers 70%

Des serious games sont joués tout au long de la formation.

## Les bases du management visuel

- L'Obeya : Kesako ?
- Les avantages du management visuel
- Zoom sur KANBAN
- Les approches visuelles dans SCRUM et KANBAN
- Les indicateurs et leurs représentations graphiques (WIP, Flux cumulatif, burndown...)
- Comment co-construire un management visuel ?
- 

## Les principaux outils

- Le mindmapping,
- La matrice des attendues,
- l'arbre à post-it
- Le diagramme des affinités,
- Le speed boat...
- 

## Mettre en place un management visuel

- Définir le type de management visuel
- Le matériel
- Les différentes cérémonies autour du management visuel

Serious game : Simulation de la mise en place et de la gestion au quotidien d'un management visuel

- 



# Kanban – 2 jours

## Objectifs

- Comprendre les principes de l'agilité
- 
- Connaître les différentes démarches agiles
- Comprendre le fonctionnement de KANBAN
- Expérimenter par la pratique
- Comprendre l'optimisation du temps de traitement

## Public

Manager, Développeur, Scrum master, Product Owner...

## Prérequis

Avoir eu une expérience dans la gestion de projets dans l'IT

## Méthodes pédagogiques

Pour ancrer les apprentissages :  
Théorie 40% - Ateliers 60%

Des serious games sont joués tout au long de la formation.

Dans un contexte agile, SCRUM est le cadre adopté par la plupart des entreprises mais ce n'est pas la seule approche. Il est parfois plus judicieux d'adopter un mode de fonctionnement de type KANBAN dans un contexte multi projets, multi équipes et très changeant.

## Introduction à l'Agilité

- Le Manifeste Agile
- Les principes de l'Agilité
- Limites des méthodes classiques / prédictives
- Tour d'horizon des différentes méthodes agiles
- 

## Les pratiques agiles

- Livraisons itératives, incrémentales et adaptatives
- L'auto-organisation de l'équipe
- Les estimations
- La priorisation et la planification
- La rétrospective
- 

## Principes du Kanban

- Kanban, késako ?
- Visualiser le flux
- Les goulets d'étranglement (théorie des contraintes)
- WIP Limit
- Organiser un Kanban
- 

## Mesurer le flux

- Temps total (Lead Time)
- Temps de résolution (Cycle Time)
- Travail en cours et goulet d'étranglement (Cumulative Flow Diagram)
- 

## Etudier et améliorer le système Kanban

- Théorie des contraintes
- Calcul des WIP Limit avec la loi de Little
- Réunion quotidienne et feedback
- Optimiser et réduire le temps de traitement
- 



# Devenir facilitateur – 2 jours

## Objectifs

- Améliorer l'efficacité des réunions
- Améliorer les échanges entre les collaborateurs
- Approfondir une problématique et la résoudre par la créativité
- Travailler sa posture de facilitateur

## Public

Manager, Chef de projets, Consultants, Formateur, Coordinateur ou toute personne amenée à conduire des réunions.

## Prérequis

Aucun

## Méthodes pédagogiques

Pour ancrer les apprentissages :  
Théorie 40% - Ateliers 60%

Des serious games sont joués tout au long de la formation.

Comment permettre à chaque membre d'un groupe de s'impliquer dans la construction d'une solution acceptable par tous ?

Comment engager le groupe face à un enjeu commun et dans la mise en œuvre des actions qui en découlent ?

La facilitation peut répondre à ces challenges. Cette formation permettra au stagiaire de comprendre et animer un groupe en tirant parti de l'intelligence collective.

### Introduction à l'intelligence collective

*Serious Game* : Briser la glace pour apprendre à se connaître, jauger son public, donner de la cohésion au groupe.

- La force de l'intelligence collective

*Serious Game* : NASA GAME

- Les dangers du groupe (paradoxe d'Abilène, expérience de Ash...)
- 

### Introduction aux innovations games

- La résolution de problèmes
- Les techniques de génération d'idées en groupe
- La rétrospective et la priorisation des actions

*Serious Game* :: brainwriting, speedboat, blason, matrice DIXIT...

- 

### La posture du facilitateur

- Mobiliser un collectif et coopérer : quels enjeux ?
- Comprendre les processus de résistance au changement
- Construire la carte des partenaires et comprendre la stratégie des alliés

*Serious Game* : Les anti-patterns du facilitateur

- 

### Mobiliser l'intelligence collective dans le groupe

- Concevoir des ateliers collaboratifs (règles des 7P)
- Le matériel
- Cohésion d'équipe et team building

*Serious Game* :: Chocomiam game

- 



# Manager Agile – 2 jours

## Objectifs

- Comprendre le Management 3.0 et l'agilité
- Comment responsabiliser les équipes et favoriser l'auto-organisation
- Comprendre et motiver les personnes
- Développer les compétences des équipes
- Savoir gérer les changements et accompagner la croissance des structures agiles
- 

## Public

Direction, Manager, Responsable produit, Scrum master, Coach agile, RH.

## Prérequis

Connaissance des méthodes agiles est un plus mais pas indispensable.

## Méthodes pédagogiques

Pour ancrer les apprentissages :  
Théorie 50% - Ateliers 50%

Des serious games sont joués tout au long de la formation.

## Agilité et complexité

- Comprendre l'agilité et ses pratiques
- La différence entre complexe et compliqué
- La fin des systèmes prédictifs
- Les enjeux et défis de l'adoption de l'agilité
- Leader et Manager Agile
- Propriétés d'une équipe agile et d'une organisation agile

## Motiver les personnes

- Différence entre motivation extrinsèque et intrinsèque
- Les 10 désirs intrinsèques
- Comprendre ce qui est important pour les personnes de votre équipe
- La grille de célébration

## Responsabiliser les équipes

- Les fondamentaux de l'auto-organisation
- Exemples d'entreprises libérées
- Protocoles de décision, Gestion de conflit
- Les rôles délégués
- Cultiver la relation de confiance

## Développer les compétences

- Niveau de compétence et de discipline
- Quand et comment appliquer les 7 méthodes du développement de compétence ?
- Comment mesurer les progrès dans un système complexe ?
- Les effets de la sous-optimisation

## Aligner les contraintes

- Les 3 responsabilités du manager
- Différence entre gouvernance et leadership
- Définir un objectif commun à l'équipe
- Protéger l'équipe auto-organisée

## Développer la structure

- Agrandir une structure organisationnelle comme une fractale
- Différentes formes d'organisations agiles
- Comment concilier spécialisation et généralisation
- Comment choisir entre des équipes fonctionnelles et cross-fonctionnelles
- Le leadership informel et la croissance des titres
- Traiter les équipes comme des unités de valeurs dans un réseau de valeurs

## Gérer le changement

- Les 4 faces de la gestion des changements
- Le modèle ADKAR
- La diffusion d'une innovation culturelle



# Chef de projets informatique – 4 jours

## Objectifs

- **Développer l'état de l'art et son expertise en matière de gestion de projet**
- **Harmoniser les méthodes de gestion de projet (outils, CR, reporting...)**
- **Adopter des outils, des méthodes et des langages communs.**

## Public

Professionnels de l'informatique

## Prérequis

Pas de prérequis nécessaires

## Méthodes pédagogiques

Pour ancrer les apprentissages :  
Théorie 70% - Ateliers 30%

Chaque concept abordé est illustré par des retours d'expériences du formateur. Nous lettons à disposition des modèles, plan-types, check-lists, tableaux de bord... directement utilisables.

Le chef de projets informatique est le garant des délais, des coûts et de la qualité. Il est également garant du respect des besoins des utilisateurs.

Mais au-delà de ces missions centrales, il doit également réaliser les études nécessaires en s'assurant que la solution proposée s'intègre dans le Système d'Information existant, élaborer le cahier des charges adéquat, gérer les développements, planifier le travail des développeurs et/ou des équipes externes en tenant compte des délais impartis et enfin assurer la recette.

Autant de missions qui nécessitent une organisation sans faille et le retour à des outils ou des méthodes qui lui seront confiés. Les participants à cette formation de 4 jours disposeront des fondamentaux de la conduite de projets.

### Qu'est-ce qu'un projet ?

- Définir un projet
- Le triangle QCD
- Le cycle de vie d'un projet
- Les acteurs du projet
- Les types de contrats
- Portefeuille de projets / Multi projets
- Les grandes phases d'un projet Informatique

### Que demande-t-on à un CP ?

Ses activités : Planification, Analyse du besoin, Manager, coordonner, gérer les risques, gérer les coûts, assurer la qualité, ...

- Qualités requises d'un chef de projet
- Analyser les besoins
- Planifier le projet
- Analyser les risques sur votre projet
- Assurer la gouvernance de projet (types de réunion pour un projet avec leur principe de gestion)

### Méthodes et techniques de gestion de projet

- Méthodes : PRINCE2, Waterfall, Cycle en V, Itil, Agilité, etc...
- Découpage – Notion d'Organigramme Technique
- GANTT et PERT
- Méthodes d'estimation des charges
- Calcul du chemin critique
- Calcul des marges totales et libres



# Chef de projets informatique – 4 jours (suite du programme)

## Objectifs

- Développer l'état de l'art et son expertise en matière de gestion de projet
- Harmoniser les méthodes de gestion de projet (outils, CR, reporting...)
- Adopter des outils, des méthodes et des langages communs.

## Public

Professionnels de l'informatique

## Prérequis

Pas de prérequis nécessaires

## Méthodes pédagogiques

Pour ancrer les apprentissages :  
Théorie 70% - Ateliers 30%

Chaque concept abordé est illustré par des retours d'expériences du formateur. Nous lettons à disposition des modèles, plan-types, check-lists, tableaux de bord... directement utilisables.

## Les outils

Outils utilisés par le chef de projet pour ses activités (Tableau de gestion des risques, Gantt, Tableaux de bord, etc...)

- Plan d'Assurance Qualité de projet
- Tableaux de bord – Journal de bord
- Avancement par la durée ou par la charge, le Reste à faire
- Courbe en S (cbtp, cbte, crte) et principe d'analyse
- Tableau de jalons
- Indice de performance pour estimer le RAF (Exigences)
- Indicateurs clés dans un projet

## Management d'équipe

- Profil socio-dynamique d'une équipe
- Style de management et utilisation
- Fédérer et Motiver son équipe
- Atelier Moving motivator
- Les enjeux et les attentes d'une équipe
- Atelier Wish Matrix
- Savoir déléguer, responsabiliser
- Atelier Delegation Board

## Ateliers simulation de projet sur la forme de jeux de rôles

- Recueil des besoins
- Gouvernance de projet
- Communication de crise

## Atelier Accompagnement au changement

- Etude de cas
- Bonnes pratiques

## Ateliers thématiques

- Méthode de résolution de problème (Pareto, diagramme de cause à effet, les 5 P,...)
- L'amélioration continue
- Génération d'idées
- Hiérarchiser les idées
- REX Projet



## SOMMAIRE

<b>Digital .....</b>	<b>23</b>
Angular – 3 jours .....	24
PHP Initiation – 3 jours .....	25
PHP Langage Objet – 4 jours.....	26
PHP Avancé – 4 jours.....	27
Spring Boot– 3 jours.....	28
React JS – 3 jours.....	29
HTML/CSS/JS - 3 jours.....	30



# Angular – 3 jours

## Objectifs

- Être capable de développer une application avec Angular point de départ pour une utilisation avancée du Framework

## Public

Développeurs

## Prérequis

Bases en HTML5 / CSS3 / JS  
Bases en Programmation Orientée Objet

## Méthodes pédagogiques

Alternance entre cours et travaux pratiques  
4 travaux pratiques, avec une difficulté croissante sont réalisés  
Les corrigés sont fournis

## Introduction

- D'où vient Angular
- A quoi sert Angular
- Pourquoi l'utiliser

## Pré-requis

- ECMAScript (ES2015+)
- TypeScript
- Reactive X (RxJS)
- NodeJS et NPM
- Environnement de développement
- 

## Angular – Partie 1

- Utiliser Angular CLI
- Structure d'une application Angular
- Qu'est-ce qu'un composant ?
- Syntaxe des Templates

## TP 1 – Créer son premier composant

- Création d'un composant basique

## Angular – Partie 2

- Directives
- Interactions entre les Composants
- Cycle de vie des Composants

## TP 2 – Créer des composants réutilisables

- Itération et recherche sur des composants

## Angular – Partie 3

- Pipes
- Services

## TP 3 – Récupération de données via une API externe

- Utilisation de l'API The Movie Database

## Angular – Partie 4

- Utilisation avancée d'Angular CLI
- Router

## TP 4 – Utilisation du Router

- Création d'une nouvelle page, avec système de navigation

## Angular – Partie 5

- Bonnes pratiques sur un projet conséquent





# PHP Initiation – 3 jours

## Objectifs

- **Comprendre comment fonctionne PHP et réaliser un mini site web avec des formulaires et une partie sécurisée en autonomie.**

## Public

Stagiaire, Développeur junior

## Prérequis

Avoir des bases en algorithmique.  
Connaissance basique de HTML et CSS.  
Connaissance basique de Linux.

## Méthodes pédagogiques

50% de théorie, 50% de pratique via des exercices et des travaux pratiques pour mettre en œuvre les compétences apprises

### Introduction à PHP

- Historique
- Versions
- Évolution

### Ecosystème PHP

- Documentation
- Logiciels
- IDE
- AFUP

### Serveur HTTP et ligne de commande

- Serveur HTTP
- Ligne de commande

### Configuration et résolution d'erreurs

- Les extensions
- Xdebug

### Base Web

- MVC

### Bases du langage

- PHPDoc
- Syntaxe
- Types de données
- Opérateurs
- Boucles
- Formulaires
- Expressions régulières
- Manipulation de chaînes et de tableaux

### Création, utilisation et test de fonctions

- Fonctions
- Fonctions anonymes

### Stockage de données

- Filesystem (binaire, XML, JSON)
- Bases de données (PDO)

### Cookies et sessions



# PHP Langage Objet – 4 jours

## Objectifs

- Comprendre l'architecture d'une application PHP orientée objet.
- Réaliser un projet en respectant les bonnes pratiques

## Public

Stagiaire, Développeur junior

## Prérequis

Avoir des bases en PHP.  
Connaissance de Linux..

Méthodes pédagogiques

50% de théorie, 50% de pratique via des exercices et des travaux pratiques pour mettre en œuvre les compétences apprises

## Un environnement de développement efficace

- Homestead

## Utilisation de bibliothèques

- Composer

## PSR (PHP Standards Recommendations) Langage objet

- Objets, classes et héritages
- Les namespaces
- Les Interfaces
- Les traits
- Classes et fonctions anonymes
- Exceptions
- Méthodes magiques
- Sérialisation
- Encapsulation
- Polymorphisme

## SPL

- Exceptions
- Itérateurs
- ...

## Nouveautés de PHP7

### Tests unitaires

- Pourquoi ? Comment ?
- PHPUnit
- Mocks
- Test Driven Development
- Refactoring

### Les CMS et les frameworks

### Déploiement d'une application

- Deployer



# PHP Avancé – 4 jours

## Objectifs

- Développer une application de A à Z, de l'architecture au développement

## Public

Développeur senior ou junior avec déjà une expérience PHP significative.

## Prérequis

Maîtrise de PHP et des concepts de base de l'objet.  
Connaissance de Linux

## Méthodes pédagogiques

50% de théorie, 50% de pratique via des exercices et des travaux pratiques pour mettre en œuvre les compétences apprises

## Design patterns

Les principes du génie logiciel, GRASP, SOLID...

## Langages objet

- Les traits
- Les interfaces
- Les générateurs
- Réflexion
- Late state binding

## Gestion du cache

## Outils de mesure de la qualité

- PHP\_CodeSniffer
- PHPDepend
- Sonar...

## Intégration continue

## Sécurité

- Injection SQL
- CSRF et le XSS
- Chiffrement et hachage
- Echappement



# Spring Boot– 3 jours

## Objectifs

- **Savoir développer une première application fonctionnelle en Spring Boot de A à Z**
- **Savoir packager et distribuer une application Spring Boot**

## Public

Développeur JAVA n'ayant jamais pratiqué ou peu Spring Boot.

## Prérequis

Pratique de la programmation JAVA/J2EE

## Méthodes pédagogiques

40% de théorie, 60% de pratique  
Développement d'une application fil rouge permettant de mettre en pratique les compétences apprises

## Rappels sur spring core

- Inversion de dépendance
- Les Annotations
- Notion de Scope

## Panorama de l'offre spring boot

- Accès SGBD: jdbc, JPA, mongodb, ...
- Accès broker: JMS, RabbitMQ, Kafka

## Présentation du framework Spring

- Conteneur léger & inversion de dépendance
- Configuration

## Programmation WEB MVC

- Modèle MVC
- Contrôleurs
- Bonnes pratiques

## Accès aux données

- JDBC
- Spring DATA JPA

## Présentation des autres Modules Spring

- Spring security
  - Présentation
  - Cas d'exemples
- Spring boot Actuator
  - Endpoints: Activation , sécurisation
  - Health check

## Cas Pratique:

- Initialisation d'un docker compose
- Utilisation de Spring Initializr
- La configuration
  - Starters
  - Fichiers properties / yaml
  - Logging, Monitoring
  - Mise en œuvre de REST
  - Mise en œuvre de JPA
  - Mise en œuvre de la sécurité
  - Tester
  - Builder: Fat jar/ war

## Microservices

- Les concepts
  - Définitions & outils de base

## Aspects avancés

- API Management
- Service Registry
- Spring cloud: presentation



# React JS – 3 jours

## Objectifs

- Être capable de développer une application complexe avec React

## Public

Développeurs web

## Prérequis

À l'aise avec javascript et ESNext.  
Connait le paradigme du FRP.  
Base en programmation orientée objet.

## Méthodes pédagogiques

Alternance entre cours, démonstrations et travaux pratiques autour d'un fil rouge. Les corrigés sont fournis.

## React

- Qu'est-ce que React
- Virtual DOM
- React dans l'écosystème JS aujourd'hui

## Mise en place d'un projet

- Intégrer dans un projet existant
- Utiliser un boilerplate
- Outils de développement
- Les bonnes pratiques
- Les tests avec React

## Les composants

- L'approche composant
- Les JSX
- Les méthodes du cycle de vie
- Les hooks
- Context API
- Le style et les composants
- Tester correctement les composants

## React Router

- Les différents routers et leurs utilisations

## Store

- Comprendre le pattern Flux
- Implémenter un store correctement avec Redux
- Utiliser redux à fond

## Écosystème

- L'asynchronisme avec react et redux
- Modifier les headers de vos pages
- Les transitions CSS avec react
- 

## Le rendu côté serveur

- Connaître les pour et les contres et comprendre les difficultés
- Mettre en place un SSR



# HTML/CSS/JS - 3 jours

## Objectifs

- **Savoir créer une page HTML, la mettre en page avec du CSS puis enfin ajouter du dynamisme avec du JS.**
- **Découverte aussi des méthodes « modernes » de mise en page CSS (flexbox, grid) ainsi qu'un peu d'ES6+**

## Public

Webdesigner, Développeur venant d'un autre langage.

## Prérequis

Avoir une connaissance minimum de son ordinateur ainsi qu'une initiation à l'algorithmie

## Méthodes pédagogiques

60 % théorie, 40% pratique  
Un mini site sera créé au cours de 3 TPs : un pour le HTML, un pour le CSS et un pour le JS

## Introduction

- Historique d'Internet et son évolution
- Les outils conseillés

## HTML

- La structure
- Les balises
- Les attributs
- Comment déboguer
- TP

## CSS

- Structure
- Spécificités
- Les sélecteurs
- Les propriétés
- Les unités
- Modèle de boîte
- Positionnement
- Flexbox
- Grid
- Les fonctions
- Media Query
- Animation
- Comment déboguer
- TP

## JS

- Les objets
- Les opérateurs
- Les instructions
- Souplesse et faiblesse du langage
- Comment déboguer
- Jargon JS
- Les événements
- ES6+
- TP



# Devops

Comme le mouvement Agile a rapproché donneurs d'ordre et équipes de réalisation autour d'une vision commune orientée « produits », le mouvement DevOps rapproche les équipes de développement et les équipes d'exploitation autour d'une vision commune orientée "service", afin de concilier réactivité et qualité de service au profit des métiers.

L'approche DevOps permet d'accélérer le « Time to market » en fluidifiant la chaîne de fabrication des projets informatiques, en unifiant les processus et les outils, de la conception à la livraison à l'utilisateur final.

## Pourquoi cette approche ?

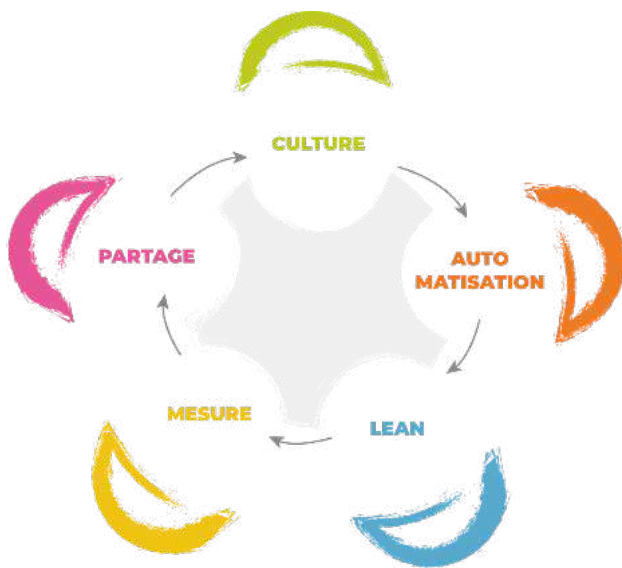
Afin d'accompagner les clients dans leurs transformations digitales et répondre à leurs enjeux :

Soutenir le business et être compétitif	Optimiser les coûts	Développer des outils fiables et performants pour les utilisateurs	Améliorer la qualité de service
Respecter les engagements	Être force d'innovation	Fournir des interfaces techniques simples (API, IHM...)	Collaborer et regagner la confiance



## Le mouvement DevOps autour du Framework « CALMS »

Le mouvement DevOps permet d'unifier les forces, des infrastructures aux applications, dans un cercle vertueux symbolisé par Le Framework « CALMS »



**Culture :** Principale mission d'une initiative DevOps, la culture du partage des responsabilités, de la transparence et du retour d'information doit se faire à tous les niveaux de l'organisation en supprimant les silos.

**Automatisation :** Permet d'éliminer les tâches manuelles et répétitives, de suivre des processus et de créer des systèmes fiables sur l'ensemble du pipeline de distribution.

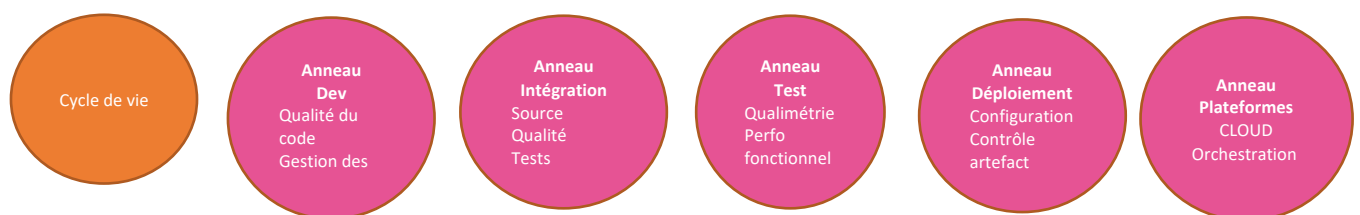
**Lean :** Afin de créer un flux continu, en maximisant la valeur et en évitant le gaspillage tout en apportant la notion d'amélioration continue et de droit à l'échec.

**Mesure :** Avec Devops tout est mesuré et accessible à tous : des métriques projet à la remontée d'indicateurs utiles pour la surveillance et à l'amélioration de la performance de l'ensemble de la chaîne.

**Partage :** les équipes Dev et Ops doivent collaborer et partager des objectifs communs ainsi que les responsabilités et les réussites. Les méthodes Agile, IITIL, Lean doivent être utilisées et harmonisées entre tous les acteurs.

## Le Pipeline d'intégration et de Déploiement Continu :

Il est réalisé en créant une chaîne d'outils intégrés et automatisés qui contient généralement les catégories d'outils suivants :



**Gestion du cycle de vie des applications (ALM)** - Les outils de gestion du cycle de vie des applications créent un environnement normalisé pour la communication et la collaboration entre les équipes de développement et les groupes associés.


**Build :** Ces outils fournissent un référentiel logiciel pour le stockage et la gestion des versions des fichiers binaires et des métadonnées associées.

**Qualité du Code :** L'automatisation est utilisée pour écrire ou automatiser le processus de compilation du code source.

**Référentiel de gestion des artefacts :** Ces outils fournissent un référentiel logiciel pour le stockage et la gestion des versions des fichiers binaires et des métadonnées associées.







**Test** : Les outils de test automatisés incluent les tests unitaires et de réception, les tests de performance et les tests de charge.

**Intégration continue** : Les outils de CI automatisent le processus de compilation du code dans une construction après que les développeurs aient archivé leur code dans un référentiel central. Une fois que le serveur CI a construit le système, il exécute des tests unitaires et d'intégration, enregistre les résultats et publie généralement une version étiquetée des artefacts déployables.

**Déploiement continu** : Les outils de CD automatisent les déploiements d'applications dans les différents environnements dont celui de production. Ils facilitent les retours d'informations et la livraison continue tout en fournissant les pistes d'audit requises, les versions et le suivi des approbations.

Cette étape est la plus complexe à mettre en œuvre et repose pour beaucoup sur le choix qui sera fait de l'outil de déploiement en adéquation avec les Dev et Ops, mais est également lié aux contraintes et type d'infrastructure de l'entreprise (Docker, Cloud etc...)





# SOMMAIRE

<b>Devops .....</b>	<b>31</b>
Les fondations de DevOps – 2 jours.....	35
Docker– 3 jours.....	36
Ansible – 3 jours.....	37
Git – 1 jour .....	38
Démarrer avec Git / Docker / PHPUnit – 2 jours .....	39



# Les fondations de DevOps – 2 jours

## Objectifs

- Comprendre ce qu'est le mouvement DevOps
- Connaître les enjeux métier de DevOps
- Maîtriser les valeurs, principes et pratiques de DevOps au travers du modèle CALMS

## Public

Étudiants, Managers, Fonctionnels, Chef de projet/produit, Infrastructures Développeurs, et ITSM/, Coach Agile.

## Prérequis

Aucun

## Méthodes pédagogiques

Pour ancrer les apprentissages :  
Théorie 70% - Ateliers 30%

## Module 1 : Définition de DevOps

- Pourquoi DevOps ?
- Définition
- Exercice
- Conclusion ce qu'est DevOps
- Quizz
- Le modèle CALMS

## Module 2 : Le modèle CALMS

### Partie A : LA CULTURE

- Pourquoi transformer l'organisation ?
- Comprendre la culture de l'organisation
- La dette culturelle
- La culture Devops
- Conduire un changement de Culture
- La transformation de l'Organisation

### Partie B : L'AUTOMATISATION

- Pourquoi automatiser ?
- Le pipeline de déploiement
- Périmètre de l'automatisation
- La Tool chain
- Le Cycle de vie et les anneaux du devOps
- Quizz

### Partie C : LE LEAN

- Définition du LEAN
- Le Gaspillage
- Les 5 Principes du Lean
- DevOps et les 3 voies
- Pratiques associées au Lean

### Partie D : LA MESURE

- Introduction : Devops et performance IT
- Mesurer la performance
- Mesurer, Monitorer l'ensemble de la chaine

### Partie E : LE PARTAGE (SHARE)

- Notion de partage
- Compétences et équipe Devops
- Collaboration
- Améliorer la communication
- Les outils collaboratifs de DevOps



# Docker- 3 jours

## Objectifs

- Appréhender la containerisation
- Créer et modifier des containers en local et en remote
- Être autonome sur l'exploitation d'une machine Docker
- Connaitre les fondamentaux de Docker (volumes / networks / reprises / etc)
- Connaitre docker compose et la théorie simple sur les orchestrateurs

## Public

Développeurs, Exploitants

## Prérequis

Connaissances lignes de commandes Unix (notions seulement)

## Méthodes pédagogiques

Approche orientée Dev & Ops

50% Théorie / 50% Pratique

Après chaque point vu en théorie, un TP de mise en pratique

## Qu'est-ce que Docker? Pourquoi utiliser docker ?

### Docker VS Machine Virtuelle

### L'exploitation de docker

### Containers, Images et Dockerfile

### Registry Docker

### Cycle de vie d'un container

### Création d'un container

- Choisir son image
- Le Dockerfile
- Compilation de l'image

### Lancement container

- Les commandes de base de Docker
- Docker instance
- Images
- Registry
- Containers
- La commande "run"
- Docker Clean

### Récupération de Docker

- Docker et ses montées de version
- Installation

### Les Volumes

- Le partage de répertoire depuis la machine hôte
- Le Volume Docker

### Network

- Les commandes
- Network "docker" : le bridge particulier

### Demos fonctionnalités

- Healthcheck
- Spring Boot
- Entrypoint / cmd surcharge
- Valeur par défaut
- Restart
- Load-Balancer

### Docker compose

- Versionning
- Notion de service
- Commandes de docker-compose
- Structure du docker-compose.yml
- Quelques balises courantes
- Les substitutions de variable
- Surcharges de docker compose

### Docker remote

### Docker en production

### Les orchestrateurs

- Docker Swarm
- Kubernetes
- Amazon ECS, Google Container Service (GCS), Microsoft Azure

### Y a t'il d'autres "Container Runtime"?



# Ansible – 3 jours

## Objectifs

- Apprendre le mode de fonctionnement d'Ansible
- Configurer l'agent ANSIBLE
- Utiliser Ansible VAULT pour stocker les informations sensibles
- Sécuriser les échanges entre l'agent et les nodes
- Savoir utiliser les modules proposés par la plate-forme

## Public

Ingénieurs et administrateurs Système

## Prérequis

Avoir des notions sur les commandes UNIX

## Méthodes pédagogiques

Pour ancrer les apprentissages :  
Théorie 20% - Ateliers 80%

Ansible est une plate-forme de configuration et la gestion des ordinateurs. Elle combine le déploiement de logiciels multi-nœuds, l'exécution des tâches ad-hoc, et la gestion de configuration.

### C'est quoi Ansible ?

#### Mode de fonctionnement

- La documentation ansible
- Premiers pas
- Installation
- Première connexion
- Outils de ligne de commande
- TP Premiers pas

#### Configuration

- Les options les plus importantes
- Configuration de base .ansible.cfg
- Organisation du repertoire de travail
- Fichiers d'inventaire
- Fichiers de Variables
- Priorité des variables
- Les bonnes pratiques de Configuration
- TP Fichiers d'inventaire et des variables

#### Introduction aux Modules

- Les modules Ad Hoc
- Le module blockinfile
- Le module file
- Le module service
- Le module package
- D'autres modules
- TP Modules

#### Les playbooks

- Bonnes pratiques exécution des playbooks
- Écriture des playbooks (Paramètres les plus utilisés)
- Les variables dans les playbooks
- Templating (filtres , module template)
- Les boucles
- Les conditions
- Les blocks
- Les tags
- Gestion d'erreurs
- Bonnes pratiques écriture de Playbooks
- TP Playbooks

#### Roles

- Fonctionnement des Rôles
- Bonnes pratiques écriture de Rôles
- TP Rôles

#### Ansible VAULT

- Encoder des fichiers avec ANSIBLE
- Encoder des variables en fichier de texte en clair
- TP Ansible Vault

#### TP final



## Objectifs

---

- Découvrir des commandes de base afin d'utiliser Git au quotidien

## Public

---

Développeurs, Exploitants

## Prérequis

---

Connaissances lignes de commandes Unix (notions seulement)

## Méthodes pédagogiques

---

50% Théorie / 50% Pratique

Après chaque point vu en théorie, un TP de mise en pratique

Qu'est-ce que Git

Fonctionnement Git

Le projet git

Les commandes de base

Les branches

Merge & rebase

Bonnes pratiques

Modifier un commit / le cherry-pick / Stash

Gitlab (pull request + review)



# Démarrer avec Git / Docker / PHPUnit – 2 jours

## Objectifs

- **Comprendre le fonctionnement global de chaque outil et être capable de l'utiliser en connaissant les bases**

## Public

Développeurs, chefs de projets

## Prérequis

Connaissances lignes de commandes Unix (notions seulement)

## Méthodes pédagogiques

25% Théorie / 750% Pratique

Après chaque point vu en théorie, un TP de mise en pratique

## Git

- Qu'est-ce que Git
- Fonctionnement Git
- Le projet git
- Les commandes de base
- Les branches
- Merge & rebase
- Bonnes pratiques
- Modifier un commit
- Cherry-pick, stash et rebase interactif
- Utilisation avec Gitlab

## Docker

- Qu'est-ce que Docker
- Pourquoi utiliser docker ?
- Différence entre Docker et VM
- Analyse d'un Dockerfile
- Analyse d'un Docker Compose
- Commandes utiles au quotidien

## PHPUnit

- Qu'est-ce qu'un test unitaire ?
- Que dois-je tester ?
- Assertions
- Annotations
- Fournisseur de données et exceptions
- Données de test
- Couverture de code
- Configuration
- Bonnes pratiques
- Mocks



## SOMMAIRE

<b>Communication .....</b>	<b>40</b>
Devenir Speaker – 2 jours.....	41
Devenir Formateur – 2 jours .....	42





# Devenir Speaker – 2 jours

## Objectifs

- Appréhender l'andragogie
- Comprendre comment partager un savoir
- Acquérir des bonnes postures et pratiques

## Public

Toutes personnes ayant un savoir à partager

## Prérequis

Aucun prérequis nécessaire

## Méthodes pédagogiques

40% Théorie / 60% Pratique

## Savoir-être

- Bonnes pratiques
- Andragogie
- Concentration

## Les types d'interventions

- Pitch elevator
- Quickies
- Conférence
- Atelier
- Meetup

## La préparation

- Construction d'un sujet
- Définition du public visé
- Rédaction d'un call for paper
- Tester votre talk

## Le moment M

- Atelier

## L'after

- Rédaction d'un Retour d'EXpérience
- Se protéger du spleen
- Amélioration continue



# Devenir Formateur – 2 jours

## Objectifs

- Appréhender l'andragogie
- Comprendre comment partager un savoir
- Acquérir des bonnes postures et pratiques

## Public

Toutes personnes ayant un savoir à partager

## Prérequis

Aucun prérequis nécessaire

## Méthodes pédagogiques

40% Théorie / 60% Pratique

## Savoir-être

- Bonnes pratiques
- Andragogie
- Concentration

## Savoir à transmettre.

- Identifier mes savoirs
- Être expert VS être enseignant

## Construire un cours

- Structure d'un enseignement
- Alternance Théorie/Pratique
- Mise en place de contrôle régulier

## Animer une formation

- Préparation
- Outils
- Organisation de la salle
- Trucs et astuce.

## L'after

- L'administratif
- Se protéger du spleen
- Amélioration continue

