Etat de l’art des technologies des systèmes d’information

Durée : 2 jours

Prérequis : Aucun.

Public visé : Développeurs, chefs de projets proches du développement, ingénieur scientifique sachant coder

Objectifs : Comprendre les architectures des applications de demain afin de les anticiper

Les systèmes d’information d’hier

* Les clients lourds
* Le Web et les architectures MVC ?
* Les applications téléphoniques et leurs SDK ?

Les architectures Web modernes

* Les bases du Web
* HTML 5 et CSS 3
* L’architecture MVVM (Model View View’s Model) le tueur de MVC
* Les services REST
* Java, Spring, JAX-RS, la mort de JEE
* .NET, .NET Core, Web API, la mort de MVC
* Python, Flask et Django
* Node.js, et le Full Stack JS
* Javascript, ECMAScript et TypeScript
* Les frameworks Javascripts
* Le material design et les responsive design
* Bootstrap et Angular Material
* jQuery, Angular, REACT, …
* Et PHP 7, Symfony ?

La nouvelle gestion de projet technique

* Plus de DBA, plus de Webmaster, plus d’hyper spécialiste JEE : le développeur Full Stack le tueur du développeur hyper spécialisé
* Changement de paradigme dans la conduite des équipes de développement
* Rentabilité d’un développeur Full Stack
* Difficultés : le recrutement, le salaire, …
* Pourquoi le choix d’un langage n’est plus très important : pourquoi JEE n’est plus si populaire et pourquoi tous ces nouveaux langages
* Le développement agile
* SCRUM : pourquoi il n’est plus adapté au développement Full Stack
* TDD et XP (eXtrem Programming)
* Les cycles de vies ultra-courts des nouveaux frameworks et langages
* Comment adapter cela à l’industrie
* Pourquoi CMMI n’est pas mort ?

Les applications universelles (hybrides)

* Les applications pour Smartphones
* Google Android
* Apple iOS
* Les applications universelles
* Apache Cordova et Ionic
* Electron
* Xamarin
* Universal App

Les bases de données relationnelles

* Pourquoi elles existeront encore
* Comprendre l’ACIDité
* PostgreSql, MariaDb, Sql Server, SQLite et les dinosaures (Oracle, DB2, MySql, …)
* Les ORM (Obect Relationnal Model)
* Entity Framework, JPA, …
* Les design patterns (Entity, Repository, Service, …)
* Le développement piloté par les tests (TDD)

Le Big Data

* Qu’est-ce que le Big Data
* No-Sql est-ce seulement une base de données sans SQL
* La perte du concept d’ACIDité
* Les bases de données structurées (MongoDB, Cassandra, Redis, CouchDB)
* JSON et REST
* Qu’est-ce que Apache Hadoop ?
* HDFS
* Hadoop comme DataLake
* Map Reduce
* Yarn
* Hive
* Hadoop comme DataMart
* Comment lier les bases de données relationnelles, structurées avec Hadoop ?

Les gestionnaires de packages et les nouveaux ateliers de développement

* Python PIP
* Node.js et NPM
* C# et NuGet
* Java et Maven
* Les nouveaux IDE basés sur IntelliJ ou sur Electron (Atom, WebStorm, Visual Studio Code, …)

L’internet des objets

* Qu’est que l’IoT ?
* Problématiques
* Stockage
* Analyse des données

Le Cloud

* Différence entre hosting et Cloud
* Amazon Web Service
* Microsoft Azure
* Google
* OVH le cloud français
* Un peu de droit informatique

La sécurité

* Pourquoi la sécurité est l’enjeu majeur de demain et que nous avons un retard abyssal
* Bien comprendre le cryptage, la signature et l’obscurcissement
* Bien comprendre les enjeux, le pirate est dorénavant une intelligence économique qui n’a pas pour but de détruire mais de savoir
* Qu’est que l’OWASP ?
* Qu’est qu’une injection ?
* Comment s’assurer d’une qualité logicielle suffisante afin d’éviter le piratage
* Ne plus jamais entendre dire que mon application est sécurisé car mon réseau est sécurisé

Introduction aux Data Sciences

* Qu’est que la data science ?
* Qu’est que le Machine Learning ?
* Apprentissage supervisé vs non supervisé
* MathPlotLib
* Mise en place d’une machine learning supervisé
* Qu’est qu’une régression
* L’algèbre linéaire avec Numpy
* Le machine Learning avec SciKitLearn
* L’analyse des données avec Pandas
* Présentation de Jupyter
* Les réseaux neuronaux