Prosit 1 – Burnout !

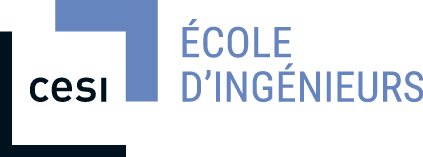


Table des matières

[1. Analyse du contexte 2](#_Toc83824561)

[2. Mots clés 2](#_Toc83824562)

[3. Problématiques 2](#_Toc83824563)

[4. Contraintes 2](#_Toc83824564)

[5. Livrables 2](#_Toc83824565)

[6. Généralisation 2](#_Toc83824566)

[7. Pistes de solution 2](#_Toc83824567)

[8. Plan d’action 2](#_Toc83824568)

[9. Réalisation du plan d’action 2](#_Toc83824569)

[10. Conclusion et retour sur les objectifs 2](#_Toc83824570)

[11. Bilan et synthèse du travail effectué 2](#_Toc83824571)

[12. Réferences bibliographiques 2](#_Toc83824572)

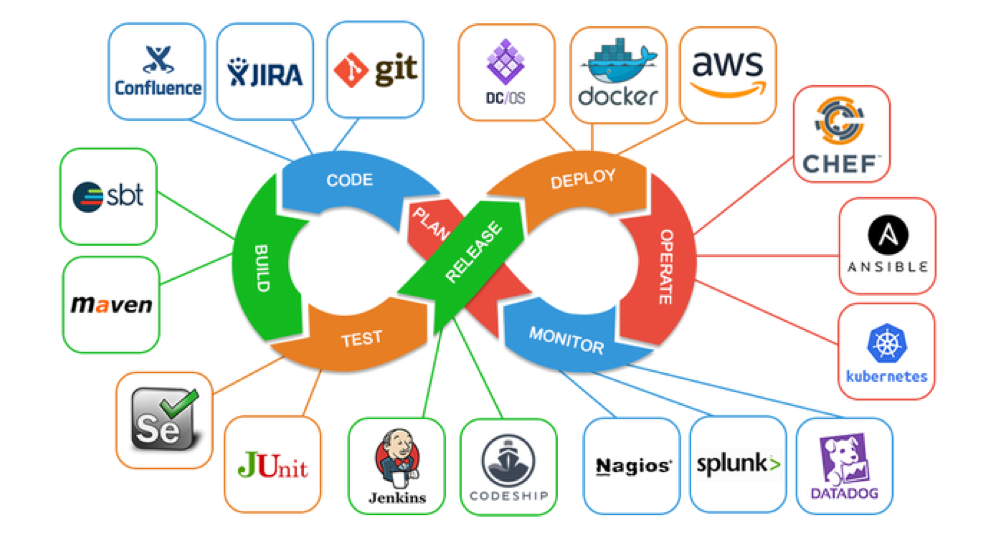
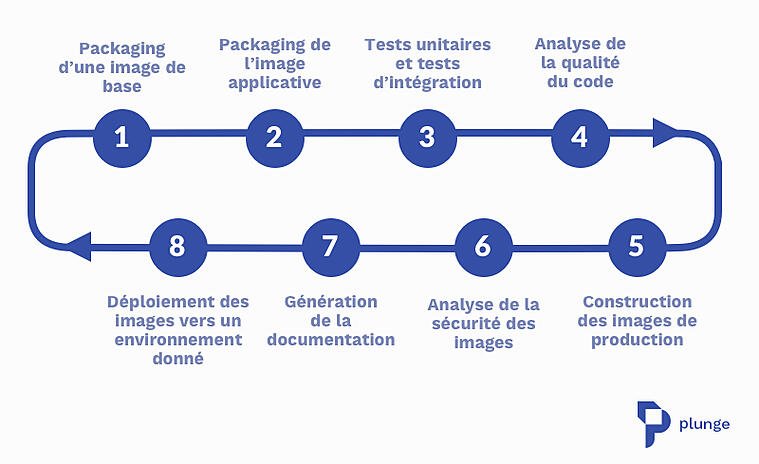
[12.1. Prosit 2](#_Toc83824573)

[12.2. Complémentaires 2](#_Toc83824574)

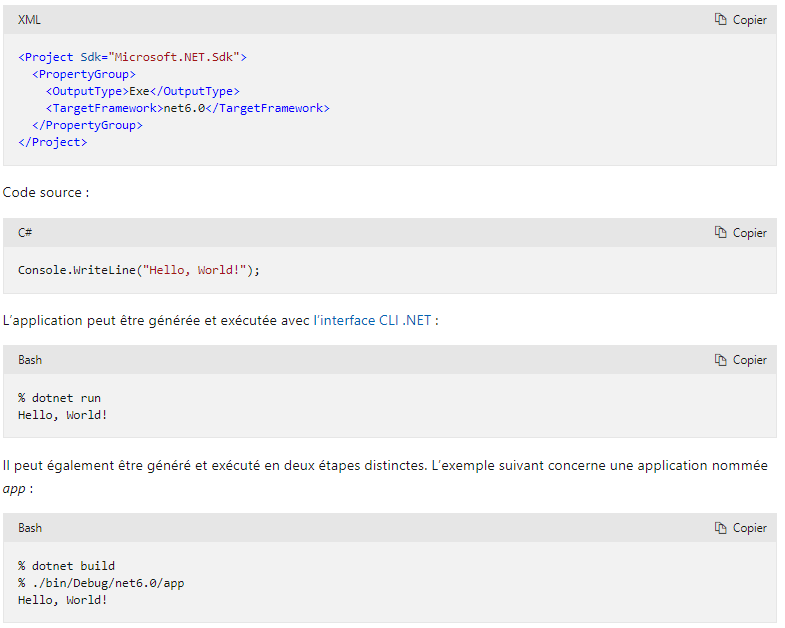
# Analyse du contexte

* Remettre à jour de système de fonctionnement de l’entreprise
* L’entreprise a du s’adapter du à sa croissance

# Mots clés

* Steering of roaming : L’itinérance ou roaming (anglicisme) est la faculté pour l'abonné d'un réseau de téléphonie mobile de pouvoir téléphoner et échanger des données via le réseau de téléphonie mobile d'un autre opérateur, aussi bien pour un appel entrant que pour un appel sortant.
* Roaming engine
* Déploiement
* Test unitaire : En programmation informatique, le test unitaire est une procédure permettant de vérifier le bon fonctionnement d'une partie précise d'un logiciel ou d'une portion d'un programme.
* Versioning : La gestion de versions consiste à gérer l'ensemble des versions d'un ou plusieurs fichiers. Essentiellement utilisée dans le domaine de la création de logiciels, elle concerne surtout la gestion des codes source.
* Gestion des builds
* Ingénieur en assurance qualité logiciel : L'ingénieur qualité est un élément déterminant du fonctionnement de l'entreprise. Il a en effet pour objectif de s'assurer que les produits ou services délivrés sont en parfaite conformité avec les exigences de l'entreprise et des clients. Dans notre cas, un logiciel
* DevOps : DevOps est un ensemble de pratiques et d'outils, ainsi qu'une philosophie culturelle. Son but est d'automatiser et d'intégrer les processus entre les équipes de développement et informatiques. 
* Travail en collaboration : Le Travail d'équipe et collaboration signifie faire partie d'un groupe et travailler en collaboration avec d'autres membres en vue d'atteindre un objectif commun relié au travail.
* Régression : Une régression logicielle est un bug logiciel qui fait qu'une fonctionnalité cesse de fonctionner après un certain évènement
* Chaine d’outil d’intégration continue : L'intégration continue est une méthode de développement de logiciel DevOps avec laquelle les développeurs intègrent régulièrement leurs modifications de code à un référentiel centralisé, suite à quoi des opérations de création et de test sont automatiquement menées. 
* Développement multiplateforme : Développer une application multiplateforme, c'est écrire du code une seule fois et obtenir une app utilisable sur plusieurs systèmes d'exploitation.
* .NET : .NET est uneplateforme de développement gratuite, multiplateforme open source permettant de créer de nombreux types d’applications. .NET repose sur un runtime hautes performances utilisé en production par de nombreuses applications à grande échelle.

Les bibliothèques et les applications .NET sont créées à partir du code source et d’un fichier projet, à l’aide de l’interface CLI .NET ou d’un environnement de développement intégré (IDE) comme Visual Studio.



.NET est gratuit, open source et est un projet.NET Foundation. .NET est géré par Microsoft et la communauté sur GitHub dans plusieurs dépôts.

Microsoft prend en charge .NET sur les systèmes d’exploitation Android, Apple, Linux et Windows. Il peut être utilisé sur les architectures Arm64, x64 et x86. Il est également pris en charge dans les environnements émulés, comme macOS Rosetta 2.

Le Common Language Runtime (CLR) est la base sur laquelle toutes les applications .NET sont basées. Les fonctionnalités fondamentales du runtime sont les suivantes :

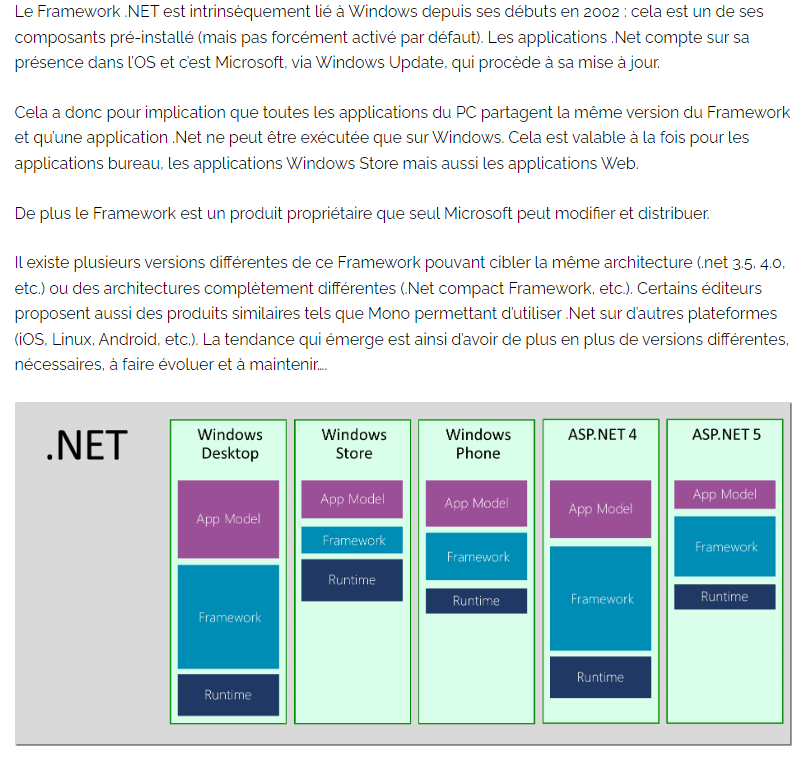
Garbage collection.

Sécurité de la mémoire et sécurité de type.

Prise en charge de haut niveau pour les langages de programmation.

Conception multiplateforme.

Support C#, F# et Visual Basic



* .NET Core :

Il s’agit d’une transformation complète du Framework .Net sous la forme de composants modulaires, utilisables à la demande, cross plateformes et … open-source !

Il est découpé en 2 parties complémentaires :

CoreFx : l’implémentation concrète de .Net Core sous la forme de plusieurs assemblies (DLL)

CoreCLR : correspond au moteur d’exécution de .Net Core ( garbage collector, compilateur JIT, types de base .Net)

Le Framework .Net devient donc une norme (ECMA 335) et Microsoft en propose déjà une implémentation sur plusieurs plateformes (Windows, Linux et Mac).

Open-Source

Les sources de ces deux éléments sont disponibles sur GitHub et il est possible à chacun de télécharger les sources, compiler soit même le Framework et surtout contribuer !

Microsoft joue le jeu de l’open-source à 100%. Cela implique notamment que Mono va pouvoir utiliser tout ou partie de .Net Core pour proposer son implémentation plus rapidement et efficacement.

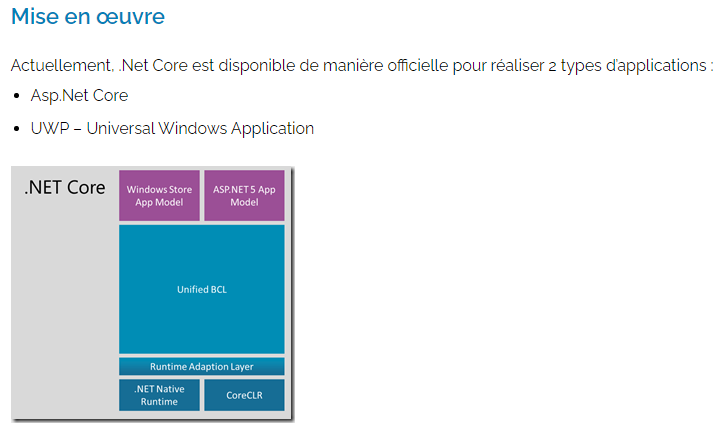
Modulaire et déployé via Nuget

Le modèle de distribution évolue aussi car .Net Core est distribué via Nuget : on ne part donc plus du principe que .Net est installé sur la machine mais on l’embarque dans l’application. Il ne s’agit pas d’un seul package mais de plusieurs disponibles : on passe d’un modèle où tout le Framework est disponible à un modèle où on ne référence (= on ajoute un package Nuget) uniquement que ce dont on a besoin. On parle aussi de modèle “pay-for-play / payer pour jouer”.

En terme de taille de binaire final, cela reste à priori raisonnable puisqu’uniquement ce qui est nécessaire sera embarqué.

Finalement, cette refonte était une opportunité qu’a saisie Microsoft d’ajouter de nouvelles fonctionnalités au Framework : Socket, collections immutables, etc.

Il est bon de savoir que même si le Framework est modulaire, les différents packages ne seront déployés que par “lot entier” après une série de tests de qualité effectués par Microsoft. Cependant, n’étant pas liées à un OS en particulier, les mises à jour pourront être beaucoup plus fréquentes.

 Une image contenant texte

Description générée automatiquement

Une image contenant texte

Description générée automatiquement

* .NET Framework : **.NET Framework** est l’implémentation originale de .NET : il prend en charge l’exécution de sites Web, de services, d’applications de bureau et plus encore sous Windows. Il a pour but de**faciliter la tâche des développeurs**. Comment ? En leur proposant une approche unifiée à la conception d’applications Windows ou Web, tout en introduisant des facilités pour le développement, le déploiement et la maintenance d’applications. Il a en revanche **besoin d’être installé sur la machine de l’utilisateur**.

Une image contenant texte

Description générée automatiquement



<https://www.papergeek.fr/microsoft-net-framework-quest-ce-que-cest-et-pourquoi-est-il-installe-sur-mon-pc-9360>

<https://emyode.com/blogue/ce-que-vous-devriez-savoir-a-propos-de-net-core/>

<https://blogs.infinitesquare.com/posts/divers/net-corequest-ce-que-cest>

* Automatisation : L'automation consiste à utiliser les services d'un logiciel dans une application informatique. Rendre des actions indépendantes s’effectuant d’elles-mêmes
* Conteneurisation (Théo-fil) : La conteneurisation consiste à rassembler le code du logiciel et tous ses composants (bibliothèques, frameworks et autres dépendances) de manière à les isoler dans leur propre « conteneur ».
* Dockeurisation (Théo-fil) Docker est un **système d'exploitation pour conteneurs**. De la même manière qu'une machine virtuelle virtualise le matériel serveur (c. -à-d. qu'il n'est plus nécessaire de le gérer directement), les conteneurs virtualisent le système d'exploitation d'un serveur.t
* Virtualisation (Théo-fil) : La virtualisation consiste, en informatique, à exécuter sur une machine hôte, dans un environnement isolé, des systèmes d'exploitation on parle alors de virtualisation système ou des applications on parle alors de virtualisation applicative.

# Problématique(s)

* Comment mettre à jour les méthodes de fonctionnement et de travail de l’entreprise ?

# Contraintes

* Solution proposée permette fonctionnement multi-plateforme
* Suivre méthode de Mathilde

# Généralisation

* IDE Commun
* Environnement de travail adéquat/adapté
* Elément gestion de projet (bonnes pratique)

# Livrables

* Liste des solutions dans un plan d’amélioration (rapport)
* Liste des logiciels utilisés

# Pistes de solution

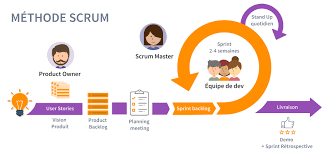
* Utilisation de Git et Docker
* Visualisons studio en IDE
* Docker bien pour versioning
* Outil de gestion de tâche
* Live coding
* Pair programming
* Mettre en place DevOps
* Test de ma régression/ test unitaire

# Plan d’action

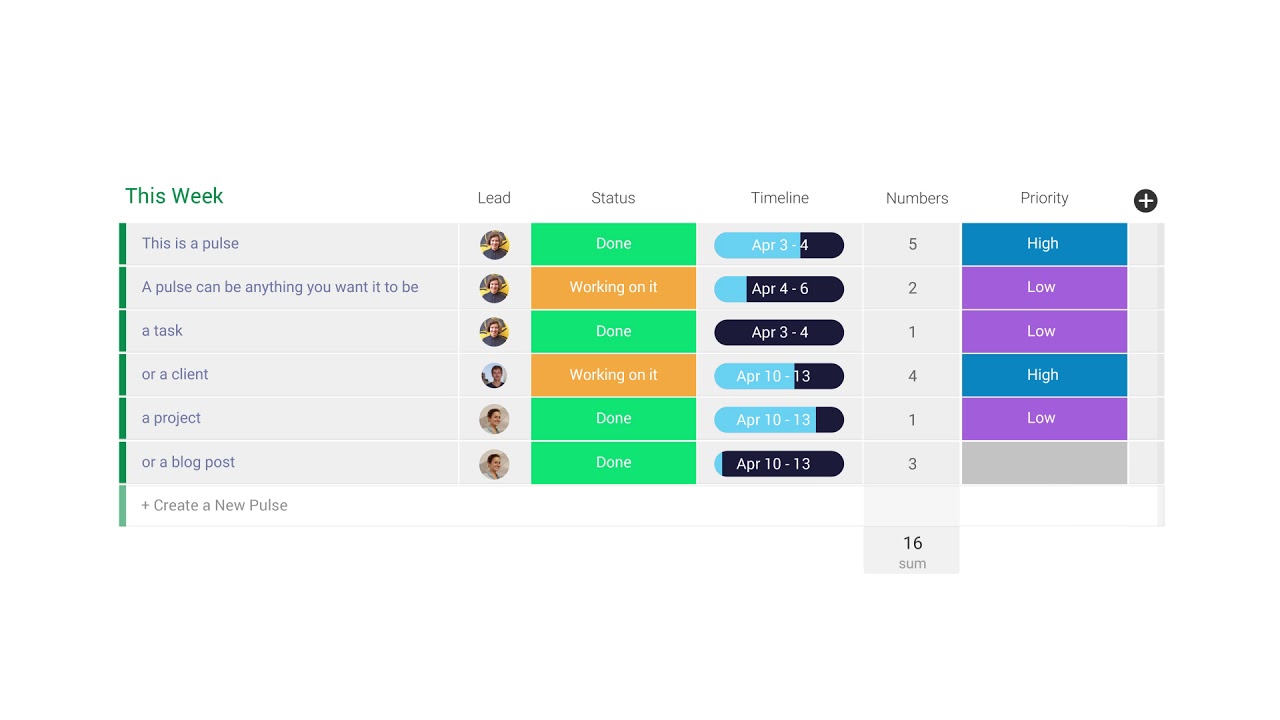
* Proposé plan d’amélioration
* Expliquer nouveaux outil logiciel mis en place et les justifier
* Proposer pratiques et outils

# 9- Réalisation du plan d’action

a) Méthode agile SCRUM : Définition : L'approche SCRUM est une méthode agile consacrée à la gestion de projet. Son objectif phare est d'améliorer la productivité des équipes, tout en permettant une optimisation du produit grâce à des feedbacks réguliers du marché.



Monday est un Work OS simple, mais intuitif, qui permet aux équipes de personnaliser des flux de travail, de s'adapter à l'évolution des besoins, de créer de la transparence, de collaborer efficacement et de supprimer les tâches manuelles et ennuyeuses. **monday**.com optimise le travail d'équipe.



Grâce à ces différentes méthodes de gestion de projet, les développeurs pourront facilement coordonner leur travail et ce grâce au chef de projet (scrum master).

Git permet de faciliter la collaboration en permettant de revenir sur les anciennes versions. La plateforme GitHub va contenir quant à elle les dépôts dans le cloud afin que les développeurs arrivent à travailler sur un même projet et distinguent en temps réel les modifications apportées par les autres développeurs. Dans notre cas, nous utilliserons azureDevOps qui a spécialement été concu pour des solutions .NET

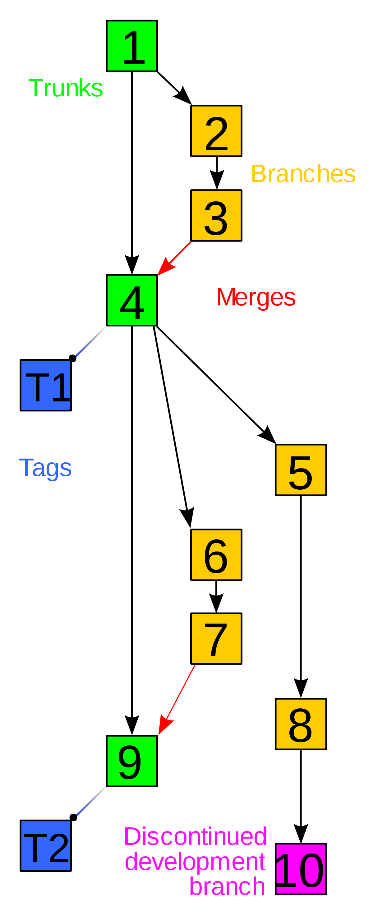
b) En programmation informatique, un environnement de développement est un ensemble d'outils qui permet d'augmenter la productivité des programmeurs qui développent des logiciels.

Git rentre donc en compte dans ce domaine. Nous allons aussi déployer un IDE (integrated développement environnement), ici visual studio car il est très compatible pour le développement .NET

c)

Une image contenant texte, horloge, clipart

Description générée automatiquement Git va nous permettre de réaliser une gestion des versions, tout en permettant un travail collaboratif ?



d) Git est un système de contrôle de version. Il s'agit d'un outil de développement qui aide une équipe de développeurs à gérer les changements apportés au code source au fil du temps.

Grâce à la gestion des branches, on peut enfin travailler sur plusieurs projets en parallèle sans se marcher sur les pieds. Grâce aux algorithmes de fusion (merge), quand un fichier a été modifié par plusieurs personnes en même temps, Git sait s’adapter et choisir un algorithme qui fusionne intelligemment les lignes du fichier qui ont été modifiées. Si par hasard 2 personnes ont modifié en même temps la même ligne (cas rare, mais qui arrive), il y a un conflit et Git laisse des marques dans le fichier pour dire qui a modifié quoi, et vous invite à décider ce que vous gardez. La gestion des versions permet de garder les anciennes versions d’un programme afin pourvoir revenir en arrière en cas d’erreur ou de besoin.

e) Les tets unitaires nous permettront de tester un à un et au fur et à mesure chaque bout de code afin d’arriver à un produit fini totalement fonctionnel et testé. Cela a pour but de valider le bon fonctionnement d’une feature avant de passer au développement de la prochaine feature.