

1 Introduction:

Ce chapitre est une mise en situation de l'environnement de notre projet de fin d'études. Nous aborderons, dans la première partie, le cadre du projet à travers une présentation de l'organisme d'accueil. Nous enchaînerons avec la présentation du cadre du projet, nous exposerons la problématique et la solution proposée et pour finir, nous ferons le choix de la méthodologie ainsi que la planification du projet.

1.1 Présentation de l'entreprise:

Move est une société de conseil et d'ingénierie spécialisée en technologies Open source. A l'aube d'une révolution industrielle dans le monde de l'IT, la vocation de Move est d'accompagner ses clients dans l'innovation technologique en leur apportant son expertise, ses conseils et ses moyens.

Elle développe des solutions globales qui créent la valeur ajoutée MOVE OS développe ses expertises et ses compétences pour accompagner les entreprises dans l'optimisation de leur performance et leurs enjeux de transformation digitale.

Move Os est un cabinet de conseil IT, créé par des experts et dirigé par des passionnés. +60 employés Siège à Tunis, des partenaires à Paris.

1.1.1 Présentation des services :

Les services de MOVE :

Backend et API: Microservice, JAVA/J2EE, Spring-boot, Rest, Docker

Devops Cloud: Jenkins, Gitlab CI, Azure, Kubernetes, AKS on azure, EKS on AWS, Aws cloud, Azure

Big Data: Stack Elastic, Mongo db, Cassandra, Redis

Full Stack et CMS: Full PHP, Drupal, Symfony, Zend

Front web et mobile : Angular, React, Vue Js, Node Js



Figure 1: Logo de la société

1.2 Présentation du cadre du projet

Dans cette partie, on va présenter notre cadre du projet on commençons par le contexte du projet, et on finissons par sa description.

1.2.1 Contexte du projet

ce travail s'inscrit dans le cadre d'un projet de fin d'études en vue de l'obtention du diplôme de licence en science informatique .

Ce projet est conçue pour les développeur .NET, il s'agit d'une interface web conviviale et une extension Vs code pour l'analyse de projets .NET .

1.2.2 Description du projet

Dans le cadre de ce projet, nous nous engageons à mettre à disposition une interface web conviviale pour faciliter l'analyse des projets. Cette interface permettra aux utilisateurs d'accéder facilement à des outils d'analyse et de visualisation des données pertinentes pour leur travail.

En outre, nous avons pour objectif d'intégrer ces fonctionnalités d'analyse directement dans l'environnement de développement Visual Studio Code, en proposant une extension dédiée. Cette intégration permettra aux développeurs de bénéficier d'un accès rapide et fluide aux outils d'analyse, sans avoir à quitter leur environnement de travail habituel.

Pour faciliter la transition et l'adoption de notre solution, nous proposons une estimation de l'effort de migration en fonction de plusieurs facteurs clés, tels que le nombre de lignes de code, les dépendances du projet et les références utilisées. Cette évaluation personnalisée aidera nos utilisateurs à planifier efficacement leur migration vers notre solution tout en minimisant les perturbations potentielles dans leur flux de travail.”

Le travail consiste à décrire les différentes fonctions principales que doit garantir l'application NetShift Advisor.

l'utilisateur:

- Ouvrir Visual Studio Code.
- Installer l'extension NetShift Advisor depuis le marché d'extensions.
- Ouvrir le projet .NET dans Visual Studio Code.

- Initier l'analyse du projet via l'extension Visual Studio Code.
- L'extension communique avec l'application web NetShift Advisor.
- L'application web effectue l'analyse du code et des dépendances du projet .NET.

1.3 Etude de l'existant

Dans cette section, on va étudier l'existant pour bien comprendre le système actuel, identifier les problèmes à rencontrer, et par la suite poser notre problématique et lui proposer des solutions.

1.3.1 Problématique et critique de l'existant

Actuellement, les processus de migration des applications .NET reposent souvent sur des méthodes manuelles, ce qui peut entraîner des inefficacités, des erreurs et des retards. De plus, la complexité croissante des applications et des environnements technologiques rend le processus de migration encore plus coûteux. Les développeurs sont confrontés à des défis tels que l'identification et la résolution des dépendances obsolètes, la mise à jour du code existant pour assurer la compatibilité avec les nouvelles versions du framework, ainsi que la gestion des tests et des déploiements pour garantir une transition fluide .

Dans ce contexte, comment pouvons-nous concevoir et développer une solution innovante et efficace, telle que l'outil NetShift Advisor, qui simplifie le processus de migration des applications .NET, réduit les coûts et les efforts associés, tout en assurant une communication transparente avec les utilisateurs finaux et en garantissant la qualité et la fiabilité des services offerts .

1.3.2 La solution proposée

Pour résoudre les lacunes et les problèmes rencontrés la solution est de concevoir une plateforme web NetShift Advisor qui facilite l'étude de la migration de projet .NET vers la version récente, l'accès à l'information, ainsi cette plateforme devrait garantir la sécurité et la confidentialité des données personnelles.

1.4 Methodologies de gestion de projet

Adopter une méthodologie de développement est nécessaire pour garantir un niveau de qualité acceptable et éviter tout dépassement. La méthodologie de développement Scrum offre une approche flexible et itérative, parfaitement adaptée à notre projet. En effet, Scrum permet une gestion dynamique des tâches, favorisant ainsi l'adaptation aux éventuels changements et imprévus tout au long du processus.

1.4.1 Méthodes Scrum

Dans le cas de notre projet, nous avons opté pour SCRUM comme méthode de développement informatique car nous avons jugé qu'elle est la méthode la plus adéquate, puisqu'elle dispose d'une organisation adaptée et d'un état d'esprit Agile, qui privilégie l'esprit d'équipe non seulement dans la réalisation technique, mais aussi pour la participation du client à la réalisation du projet.

1.4.2 Les principes de base de Scrum

- Dégager dans un premier lieu le maximum des fonctionnalités à réaliser pour former le product backlog
- En second lieu définir les priorités des fonctionnalités et choisir lesquelles qui seront réalisées dans chaque itération
- Par la suite focaliser l'équipe de façon itérative sur l'ensemble des fonctionnalités à réaliser, dans des itérations appelées Sprints
- Un Sprint aboutit toujours sur la livraison d'un produit partiel fonctionnel appelé incrément

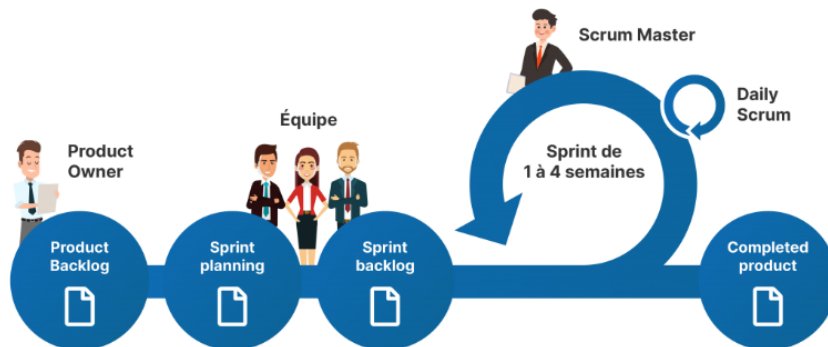


Figure 2: Le processus Scrum

Il est nécessaire de maîtriser d'autres termes pour appréhender la méthode Scrum :

- **Le Product Owner** : est responsable de la vision du produit à développer et collabore avec l'équipe de développement. Il est généralement un expert dans le domaine métier du projet.

- **Le SCRUM Master** : facilite la communication entre le Product Owner et l'équipe de développement en éliminant les obstacles qui pourraient entraver la réalisation des objectifs de chaque sprint. Son rôle est également de garantir le respect des principes et des valeurs Scrum.
- **Les Membres de l'équipe** : qui sont chargés de transformer les besoins exprimés par le Product Owner en fonctionnalités utilisables.

Scrum définit également trois artefacts :

1. Le product backlog (carnet du produit) : est un document qui rassemble les exigences initiales établies avec le client au début du projet et qui sont ensuite hiérarchisées. Toutefois, ce document évoluera au cours du projet en fonction des besoins changeants du client.
2. Le sprint backlog (carnet de sprint) : Au début de chaque sprint, l'équipe définit un objectif et lors de la réunion de sprint, les membres de l'équipe de développement sélectionnent les éléments à réaliser dans le carnet de produit pour atteindre cet objectif. Cette sélection d'éléments constitue le sprint backlog.
3. User story : est une version simplifiée et peu détaillée d'un cas d'utilisation, qui se concentre sur les objectifs métiers du logiciel en cours de développement. Elle est généralement accompagnée de critères d'acceptation tels que des scénarios de test ou des scripts de test, qui sont utilisés pour la valider.

1.4.3 Les principales méthodes agiles existantes :

- SCRUM
- EXtreme Programming (XP)
- Rapide Application Développement (RAD)
- Feature Driven Development (FDD)
- Kanban (lean)
- Dynamic systems development method (DSDM)

1.4.4 Étude comparative des méthodologies

Dans cette partie, nous présentons une étude comparative entre les méthodologies Agiles XP et Scrum, illustré dans le tableau :

Caractéristique	Scrum	XP
Planification	Au début de chaque sprint	Planning game
Estimation de l'effort	Au début de chaque sprint	Pratiques XP
Rôles	1. Scrum master (SM) 2. Product owner (PO) 3. Équipe de développement	1. Équipe de développement 2. Client
Boards/Artifacts	1. Product backlog 2. Scrum board 3. Burndown / Burnup chart	Priorisation par le client, pratiques XP
Quand choisir ?	Équipe dédiée à 100% au projet	Amélioration de la qualité du logiciel critique, prise en compte immédiate des changements
Points forts	1. Méthode leader 2. Vitesse.	1. Qualité code 2. Craftmanship 3. Outillage

Tableau 1: Comparaison des méthodologies

La cérémonie la plus cruciale de Scrum est probablement le Sprint Planning (planification du Sprint) qui a lieu au début de chaque itération.

Le suivi quotidien du projet consiste en :

- Chaque jour une réunion ou Daily Scrum de 15 minutes maximum est organisée avec l'équipe.

Conclusion:

Dans ce chapitre d'introduction, nous avons commencé par examiner les solutions actuellement disponibles, avant de passer en revue les critiques qui leur sont adressées. Nous avons ensuite présenté notre propre solution, ainsi que la méthodologie de développement agile SCRUM.