
Conception et Implémentation du module de la GMAO

Gestion de Maintenance Assistée par Ordinateur

Apprenti

Mohamed

AL NOSSIRAT

Maître d'apprentissage

Thomas

SORREL

Tuteur école

Laurent

GRÉGOIRE

غَرِيبٌ كَمْ يَبْدُو الْمَكَانُ كَمِصِيدَةٍ أحياناً
- حسين البرغوثي

Étrange comme les lieux peuvent parfois ressembler à des pièges

- Hussein AL-BARGHOUTHY

Résumé

L'ère de l'entretien est révolue depuis longtemps et la maintenance structurée dans le secteur tertiaire est aujourd'hui un bonus. Celle-ci, ne peut se faire aisément sans une GMAO paramétrable en standard, adaptable aux flux fonctionnels de chaque organisation. La maintenance des analyseurs et la documentation associée constituent une exigence de la norme NF EN ISO 15189 et de la réglementation en vigueur et, à ce titre, elles doivent être définies dans la politique qualité du laboratoire. La gestion de la maintenance préventive périodique doit être mise en œuvre par les responsables du laboratoire et la maintenance corrective doit être organisée pour ne pas compromettre la continuité des tâches au sein de la structure. Les différentes recommandations concernent l'identification des matériels incluant les analyseurs, les conditions environnementales indispensables pour un bon fonctionnement de l'analyseur, la documentation produite par le fournisseur et celle préparée par le laboratoire incluant toutes les procédures en lien direct avec la maintenance de l'analyseur.

Abstract

The era of maintenance is long gone. And structured maintenance in the service sector is now a must. This cannot be done easily without a CMMS that can be parameterized as standard and adapted to the functional flows of each organization. The maintenance of automatic analyzers and associated documentation taking part in the requirements of the ISO 15189 Standard and the French regulation as well have to be defined in the laboratory policy. The management of the periodic maintenance and documentation shall be implemented and fulfilled. The organization of corrective maintenance has to be managed to avoid interruption of the task of the laboratory. The different recommendations concern the identification of materials including automatic analyzers, the environmental conditions to take into account, the documentation provided by the manufacturer and documents prepared by the laboratory including procedures for maintenance.

Introduction

L'importance de la fonction maintenance dans les laboratoires d'analyse médicale, prend une nouvelle dimension et pousse les dirigeants dans une recherche accrue de compétitivité, de sécurité, de rentabilité, de modernisation et de disponibilité. Ainsi, la nécessité d'informatiser cette fonction apparaît comme une évidence, que ce soit sur les plans économiques, techniques ou bien organisationnels. De nombreuses applications ont vu le jour au fur et à mesure que les besoins se sont fait ressentir avec des spécificités particulières, il faut alors envisager : une conception modulaire où les différentes fonctions seront séparées ; une conception ouverte c'est-à-dire une conception évolutive permettant d'intégrer d'autres fonctions futures ; une conception intégrée avec une base de données unique qui permet d'établir le lien entre les nombreuses informations de maintenance ; un interfaçage possible avec d'autres systèmes ; une utilisation pour tous et sur le terrain ; une possibilité d'accès par code aux différents niveaux du logiciel. La GMAO (Gestion de Maintenance Assistée par Ordinateur) devient alors un outil de référence indispensable pour la gestion et la traçabilité, dans les laboratoires soucieux de gérer la maintenance et le flux d'information engendré par celles-ci. Mon travail consiste à réaliser une application de GMAO dont le but sera d'optimiser la gestion de la maintenance afin de remédier à tous les problèmes rencontrés par les services concernés, et d'atteindre les objectifs fixés par les responsables. Le premier chapitre présente des généralités sur la maintenance : définition de la maintenance, différentes politiques, niveaux, formes organisationnelles, outils, organisation des opérations et coûts de cette dernière. Le deuxième chapitre introduit et définit la GMAO d'une façon générale au départ puis propose, par la suite, un exemple d'une GMAO appliquée dans les laboratoires ainsi que ses différents modules analysés. Le troisième chapitre présente les démarches à suivre pour l'informatisation de la maintenance puis se consacre en particulier à la présentation de l'application.

Table des figures

| | | |
|------|---|----|
| 1.1 | Résultats du sondage Agilent Technologies | 3 |
| 2.1 | Types de maintenances | 9 |
| 2.2 | Frameworks back end les plus populaires en Janvier 2023 | 11 |
| 2.3 | Schéma de l'architecture MVT de Django | 12 |
| 2.4 | Schéma de l'architecture REST API | 13 |
| 2.5 | Description d'un REST API | 14 |
| 2.6 | Le modèle de maturité de Richardson | 15 |
| 2.7 | Schéma simplifié d'une application virtualisée VS application conteneurisée . . . | 18 |
| 2.8 | Frameworks Frontend les plus populaires en Janvier 2022 | 20 |
| 2.9 | Workflow de laboratoire | 22 |
| 2.10 | L'ancienne interface de paramétrage de maintenances - MCA | 23 |
| 2.11 | Modèle entité-association de la solution GMAO | 25 |
| 2.12 | Formulaire de saisie de matériel | 27 |
| 2.13 | Formulaire de saisie d'opérations | 28 |
| 2.14 | Le calendrier des interventions | 30 |
| 3.1 | Exemple d'une issue GitLab | 33 |
| 3.2 | Le cycle de vie d'une issue Gitlab | 33 |
| 3.3 | Exemple de Board GitLab | 34 |
| 4.1 | Graphique de synthèse présentant les différentes compétences techniques acquises | 36 |

Table des matières

| | | |
|----------|--|----------|
| 1 | Rapport d'action | 1 |
| 1.1 | Les problématiques | 1 |
| 1.2 | La maintenance : une tâche chronophage | 3 |
| 1.3 | Éléments pour répondre aux exigences de la norme NF en ISO-15189 | 5 |
| 1.4 | G.M.A.O | 6 |
| 1.5 | Les objectives fixées | 7 |
| 1.6 | Les résultats obtenus | 7 |
| 2 | Rapport d'étude | 8 |
| 2.1 | Définition de la maintenance | 8 |
| 2.2 | Differents type de maintenances | 8 |
| 2.2.1 | La maintenance curative | 8 |
| 2.2.2 | La maintenance préventive | 9 |
| 2.3 | L'organigramme de la maintenance | 9 |
| 2.4 | Les objectives de la maintenance | 10 |
| 2.5 | Pourquoi une maintenance informatisée | 10 |
| 2.6 | Présentation de la situation initiale | 11 |
| 2.6.1 | État de l'art | 11 |
| 2.6.1.1 | Architecture backend | 11 |
| 2.6.1.2 | Architecture REST | 13 |
| 2.6.1.3 | Le modèle de maturité de Richardson | 15 |
| 2.6.1.4 | Docker | 18 |
| 2.6.1.5 | Architecure Frontend | 19 |
| 2.6.1.6 | VueJS | 21 |
| 2.6.2 | L'ancien processus de la maintenance | 22 |
| 2.7 | Proposition d'une nouvelle solution | 24 |
| 2.7.1 | Gestion de l'inventaire matériel | 24 |

| | | |
|----------|--|-----------|
| 2.7.2 | Planification des activités de maintenance | 24 |
| 2.7.3 | Gestion des interventions | 24 |
| 2.7.4 | Suivi des interventions | 24 |
| 2.7.5 | Historique des matériels | 24 |
| 2.7.6 | Indicateurs de performance | 24 |
| 2.7.7 | Conception de base de données | 25 |
| 2.8 | Le développement | 27 |
| 2.8.1 | Gestion de l'inventaire matériel | 27 |
| 2.8.2 | Planification des activités de maintenance | 28 |
| 2.8.2.1 | Les opérations : | 28 |
| 2.8.2.2 | Les interventions | 29 |
| 2.8.3 | Le calendrier des interventions | 30 |
| 3 | La gestion de projets | 31 |
| 3.1 | L'équipe projet | 31 |
| 3.2 | Outils de gestion de projet | 32 |
| 4 | Bilan personnel du projet | 35 |
| 4.1 | Les compétences technique | 35 |
| 4.2 | Les compétences générales | 37 |
| 5 | Conclusion | 38 |

Chapitre 1

Rapport d'action

1.1 Les problématiques

L'ISO 15189 est une norme internationale publiée par l'ISO en 2012 qui spécifie les exigences de qualité et de compétence propres aux laboratoires de biologie médicale (LBM1). Son titre est "Laboratoire de biologie médicale. Exigences concernant la qualité et la compétence" Cette norme est spécifique aux LBM à la différence de la norme ISO/CEI 17025 qui concerne tous les laboratoires d'étalonnages et d'essais. Le système de management de la qualité des laboratoires accrédités 15189 est fondé sur la norme ISO 9001 :2008. La version de décembre 2012 constitue une révision technique de la version précédente, de 2007.

En France, le Cofrac accrédite sur demande les laboratoires d'analyses de biologie médicales selon un référentiel qui leur est propre. Cette accréditation est une reconnaissance des compétences techniques. Dans le cadre de la nouvelle loi HPST, cette accréditation est devenue obligatoire.

En 2013, tout laboratoire, public ou privé, devait démontrer qu'il avait entrepris une démarche d'accréditation et en 2016, tous les laboratoires publics ou privés de France devaient être accrédités ISO 15189 sur au moins 50 Par ailleurs, depuis 2011, un laboratoire de biologie médicale qui souhaiterait ouvrir ses portes ne pourra pas invoquer de délai et devra être accrédité à l'ouverture. Concernant les équipements de laboratoires médicaux, et dans le chapitre 3.5 intitulé « Matériel de laboratoire » de la norme. Extrait du guide de bonne exécution des analyses de biologie médicale : *« Les appareils doivent être périodiquement et efficacement inspectés, nettoyés, entretenus et vérifiés selon la procédure en vigueur. L'ensemble de ces opérations ainsi que les visites d'entretien et de réparation du constructeur ou de l'organisme de maintenance doivent être consignées par écrit dans un registre de maintenance affecté à chaque instrument. Le biologiste-responsable du laboratoire doit s'assurer de la mise en œuvre des moyens métrolo-*

giques nécessaires à leur vérification usuelle. Les notices d'utilisation et de maintenance d'appareils doivent être mises en permanence à la disposition du personnel utilisateur et respectées. Le fonctionnement des appareils doit être vérifié selon la fréquence préconisée par le fabricant. Des procédures de remplacement doivent être prévues en cas de dysfonctionnement d'un automate : mise en œuvre d'autres techniques ou transmission des échantillons biologiques à un autre laboratoire de biologie médicale. Ces procédures doivent prendre en compte la transférabilité des résultats afin de conserver un historique cohérent des données du patient. Responsabilité de la personne chargée de l'assurance de qualité. L'organisation du système d'assurance de qualité du laboratoire de biologie médicale peut être déléguée par le biologiste-responsable du laboratoire à un biologiste médical ou à une personne chargée de la gestion du système d'assurance de qualité qui devra avoir la formation, la compétence et l'expérience nécessaires pour accomplir la tâche qui lui sera confiée. Elle doit notamment s'assurer : [...] – de la maintenance, du bon fonctionnement des appareillages ; [...]. »[1]

1.2 La maintenance : une tâche chronophage

Dans un sondage réalisé par Agilent Technologies¹ demandait à des responsables de laboratoire de classer les tâches chronophages courantes qui ont le plus d'impact sur leurs analyses, nous pouvons constater les résultats suivants :

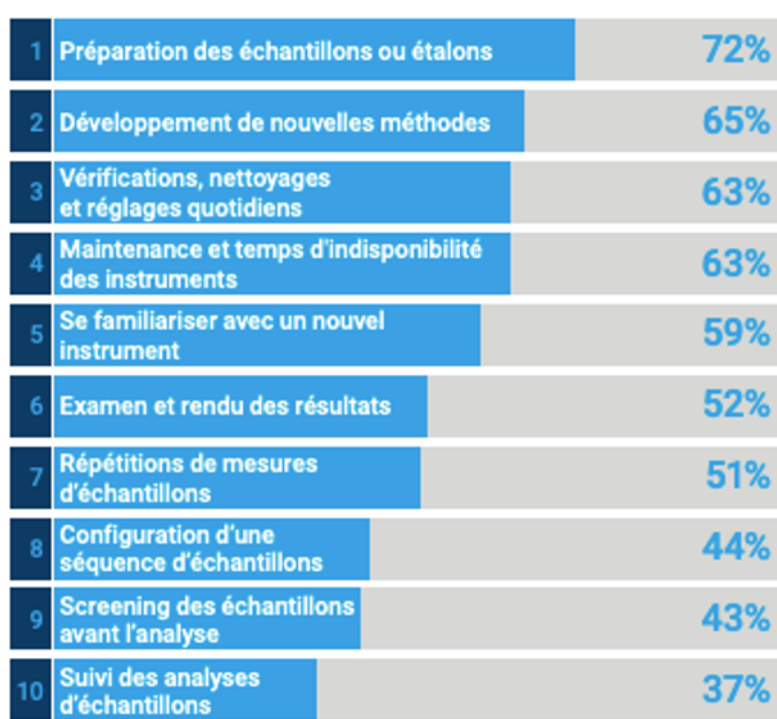


FIGURE 1.1 – Résultats du sondage Agilent Technologies

On pense souvent à tort que les instruments sont complexes, chronophages et coûteux à entretenir. Certains utilisateurs pensent également que leurs instruments analytiques continueront de fonctionner, jour après jour, sans aucun entretien ni attention. Quant aux laboratoires, ils sont nombreux à placer le temps d'indisponibilité des instruments parmi leurs plus grandes frustrations. Pourtant, les ingénieurs de maintenance constatent souvent qu'il suffit de nettoyer l'instrument ou d'effectuer un réglage de routine pour résoudre le problème. Ces tâches simples peuvent être effectuées par les analystes s'ils y sont formés. Avec des charges de travail élevées et une pression constante pour maximiser la productivité, les laboratoires commerciaux devraient mettre en place un calendrier de maintenance régulière pour garantir des performances instrumentales optimales et éviter les petits problèmes qui provoquent le temps d'indisponibilité des instruments pendant les analyses. Une bonne stratégie consiste à exécuter chaque jour un test de performances automatisé de l'instrument avant les analyses et à l'issue des analyses sans surveillance pendant la nuit. Les vérifications des performances permettent de confirmer l'état de

1. Agilent Technologies est une société qui développe et industrialise des instruments de mesure. Son siège se trouve à Santa Clara en Californie, <https://www.agilent.com/>

l'instrument avant de démarrer les analyses. Cette vérification réduit les risques de devoir arrêter l'analyse et de remesurer les échantillons si les performances se dégradent pendant la journée.

De nombreux laboratoires incluent la maintenance et le nettoyage des instruments dans leur routine quotidienne ou leurs procédures opérationnelles normalisées. Mais le moment et la fréquence de ces activités peuvent être basés sur des défaillances d'instruments, d'anciens instruments ou des tâches effectuées pour différents types de solutions ou sur d'autres techniques d'analyse de métaux. À l'inverse, il est possible que les calendriers de nettoyage et de maintenance documentés dans les procédures de laboratoire soient omis ou négligés, notamment lorsque le laboratoire est soumis à des contraintes de temps. Ne pas exécuter ces tâches risque d'impacter les résultats, puisqu'il faudra résoudre le problème et éventuellement remesurer les échantillons, induisant une perte de temps.

Le sondage a montré que Près de 10% des appels pour dépannage concernent un nettoyage de routine qui n'a pas été réalisé. Pour certains laboratoires, il est évident qu'une meilleure programmation de l'entretien courant peut éviter les pertes de temps liées à l'attente inutile d'un ingénieur de maintenance.

L'informatisation relative à la maintenance préventive aide les analystes à gérer leur calendrier de maintenance en leur permettant de configurer des alertes pour les tâches d'entretien courant. Cette dernière permet ainsi de paramétrer les fréquences de la maintenance et des vérifications quotidiennes pour qu'elles soient le mieux adaptées à la fréquence des maintenances requise pour tous types de solutions particuliers et à ne pas perdre de temps à réaliser une maintenance inutile. L'autre grand avantage de l'informatisation de la maintenance est le fait qu'elle peut être utilisée comme preuve lors d'un audit. En fait, les laboratoires peuvent décider de mettre complètement de côté les calendriers de maintenance basés sur le temps et de supprimer les enregistrements de maintenance sur papiers. La maintenance informatisée conserve toutes les données et s'occupe d'établir le calendrier de maintenance. Autre fonction utile : l'accès aux guides d'utilisation directement sur les moniteurs de maintenance. Un excellent moyen de gagner du temps et de s'assurer que les actions de maintenance sont effectuées correctement.

1.3 Éléments pour répondre aux exigences de la norme NF en ISO-15189

La gestion de l'équipement est l'un des points essentiels du système de gestion de la qualité. Une bonne gestion de l'équipement au laboratoire est nécessaire pour assurer la justesse, la fiabilité et la pertinence des analyses.

Les bénéfices d'une bonne gestion des équipements sont nombreux :

- Aide à maintenir un haut niveau de fonctionnement du laboratoire
- Réduit les variations entre les résultats des tests, et augmente la confiance du technicien dans la justesse des résultats
- Diminue les frais de réparation : moins de réparations seront nécessaires sur un équipement bien entretenu
- Augmente la durée de vie des instruments
- Réduit les interruptions de fonctionnement dues à des pannes et des défauts
- Augmente la sécurité pour les employés
- Permet une meilleure satisfaction du client

Le laboratoire doit définir clairement une politique relative à la gestion de la maintenance et celle de la documentation afférente. La durée de conservation des enregistrements en fonction de leur nature est précisée dans un document. Ces éléments peuvent être classés en trois groupes :

- Identification des matériels incluant les analyseurs ;
- Documentation produite par le fournisseur, conformément à la directive européenne ;
- Documentation préparée par le laboratoire incluant toutes les procédures en lien direct avec la maintenance de l'analyseur.

1.4 G.M.A.O

GMAO signifie la *Gestion de la Maintenance Assistée par Ordinateur*. C'est un progiciel qui contient toutes les données des opérations de maintenance curatives ainsi que préventives réalisées et en prévision. Mais surtout, il facilite la communication et les prises de décisions, grâce à une mémoire informatisée et des flux paramétrés. Le logiciel GMAO vient soutenir les équipes de maintenance dans leur travail. Il facilite le suivi des activités, fiabilise les machines, gère les risques, permet de connaître le nombre d'arrêts et leurs causes, de planifier les maintenances. En d'autres termes, il permet de mieux gérer la production.

Les bénéfices d'une GMAO :

- o Assurer une visibilité optimale des équipements
- o Assurer la suivie des opérations et des interventions sur les équipements.
- o Assurer l'enregistrement et la traçabilité des opérations.
- o Reporting et analyse des statistiques.
- o Meilleure communication interne

Le rôle d'une GMAO au sein de laboratoire :

- o Connaître et identifier et gérer les équipements à maintenir : inventaire, localisation,
- o Gestion de l'information relative à l'équipement : document constructeur, contrat de prestation etc.
- o Piloter la maintenance : préventive, curative, corrective, améliorative,
- o Planifier les interventions,
- o Gérer les demandes d'intervention,
- o Coordonner le personnel et les plannings : activités, métiers, plan de charge, prévisions, etc.

1.5 Les objectives fixées

Le premier objectif de ce projet concerne la rédaction du cahier des charges de toutes les évolutions à effectuer au cours de ce projet. Ce cahier des charges contient tous les éléments à reprendre pour la maintenance préventive et curative, ainsi que toutes les évolutions à apporter à MCA pour répondre aux exigences techniques de la partie précédente.

L'objectif, en termes de délai, est de finaliser les développements fin septembre. Le développement concerne les quatre grandes parties du cahier des charges, soit, l'inventaire des matériels, les planifications, la gestion de la maintenance périodique et l'aperçu de l'agenda des techniciens.

1.6 Les résultats obtenus

Comme prévu, le cahier des charges et les issues sont entièrement rédigés à la fin du mois de mai 2022. Les développements ont commencé au début du mois de juillet de la même année.

J'ai démarré ensuite le développement du nouveau module. Sur une période d'une semaine et demi, j'ai initié l'architecture de la base de données ainsi que la gestion des droits, la récupération des données présentes en base de données et introduis la charte graphique de la partie Frontend du module.

Arrivé à ce point du développement, j'ai rencontré diverses difficultés qui nécessitaient des éclaircissements ainsi que des réponses à mes questions de la part de mes collègues. J'ai donc fait une pause sur cette application, et me suis mis à me documenter, faire des recherches sur bonnes pratiques à adopter. Par la suite, j'ai commencé à affiner les développements sous la tutelle du chef produit et d'un collègue plus compétent sur l'application MCA.

Actuellement, une version beta de ce module est en cours de déploiement auprès des clients de Clarisys Informatique pour passer la recette de test et collecter les dysfonctionnements potentiels pour améliorer les fonctionnalités de ce module.

Chapitre 2

Rapport d'étude

2.1 Définition de la maintenance

Rappelons la norme NF X 60-010 qui définit la maintenance comme étant « *l'ensemble des activités destinées à maintenir ou rétablir un bien dans un état ou dans des conditions données de sûreté de fonctionnement, pour accomplir une fonction requise. Ces activités sont une combinaison d'activités techniques, administratives et de management.* »[2]

Par bien, il faut entendre un équipement de production, un bâtiment, un logiciel, un véhicule ou bien le regroupement ordonné de tout cela en vue d'assurer une fonction donnée, le plus souvent productrice de valeur. La position du management de ce ou de ces biens doit être de les tenir dans un état tel qu'ils puissent, de façon optimale, effectuer les tâches et remplir les fonctions pour lesquelles ils ont été conçus.

2.2 Différents type de maintenances

2.2.1 La maintenance curative

Dont le but est la remise en état après défaillance ou dégradation. Selon que la réparation est provisoire ou définitive, on parlera de palliatif (dépannage) ou de curatif(correctif). Il se pourra même que l'on procède à des modifications (améliorations) en vue d'éviter que la panne ne se reproduise ou en vue de faciliter les interventions.

2.2.2 La maintenance préventive

Qui s'attache par l'ensemble des travaux d'actions de surveillance à réduire la probabilité d'apparition de la défaillance ou de la dégradation du service.

Si les activités de préventif sont réalisées selon un calendrier fixe, il s'agit d'un préventif systématique. Si, en revanche, ces activités sont fonction d'observations ponctuelles faites sur l'état du bien, il s'agira de maintenance conditionnelle.

Enfin, si c'est l'évolution d'un phénomène que l'on suit afin de prévoir quand il faudra intervenir ou prédire dans le temps quand la dégradation irréversible se produira ou deviendra intolérable, on parlera de prévisionnel ou prédictif.

Toutes ces opérations servent à produire de la disponibilité, à donner au bien des points de rendement, à le conserver en état de bon fonctionnement. Parce que ces opérations devaient être assurées au meilleur coût, l'entreprise fut amenée progressivement à gérer la fonction maintenance de façon plus précise et plus rigoureuse. Bien gérer, c'est connaître et décider en toute connaissance.

2.3 L'organigramme de la maintenance

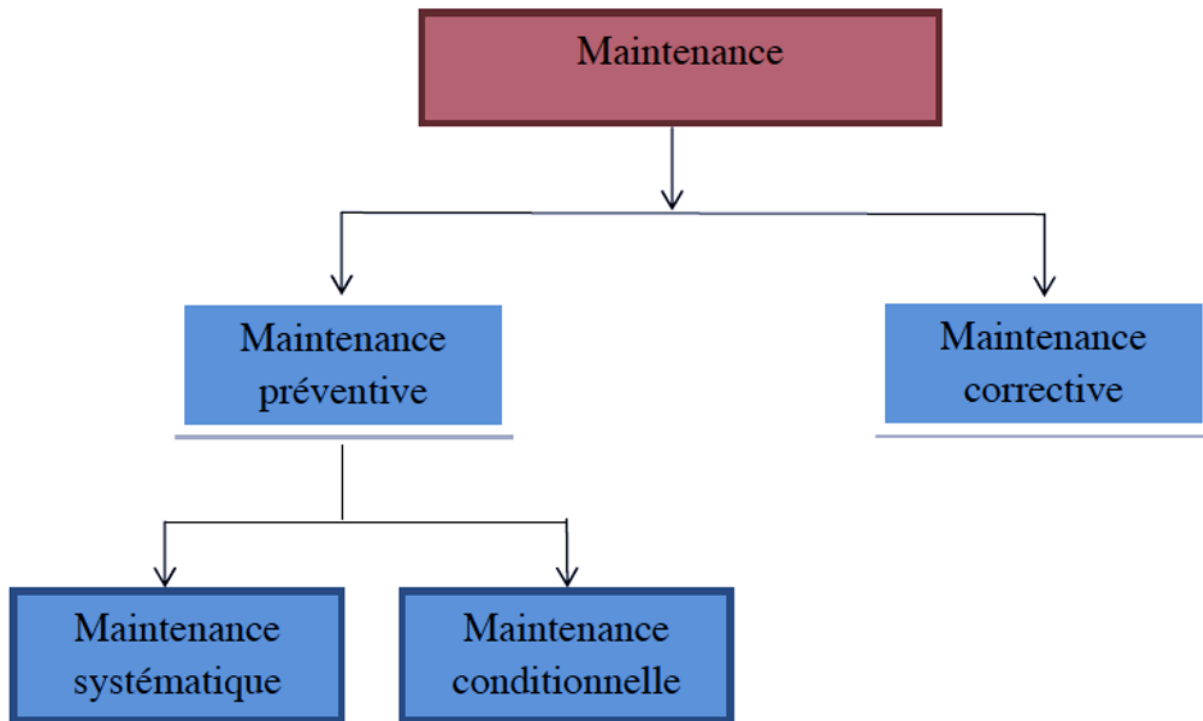


FIGURE 2.1 – Types de maintenances

2.4 Les objectives de la maintenance

Les objectifs de la fonction maintenance sont :

- Contribuer à assurer la production prévue.
- Veiller au respect des délais.
- Contribuer à maintenir la qualité du produit fabriqué.
- Rechercher les coûts optimaux.
- Préserver l'environnement de production.

Ces objectifs généraux assignés à la maintenance impliquent des missions multiples, d'où l'attribution de moyens importants : leur pleine rentabilisations amène très souvent la maintenance à assurer des missions qui, entre autres, peuvent dépasser le cadre strict de cette fonction.

2.5 Pourquoi une maintenance informatisée

L'inefficacité de la méthode papier, en raison de son incapacité à transmettre l'information en temps réel, a poussé les acteurs industriels à étudier de nouveaux concepts de gestion plus exacte et à intégrer l'informatique au sein des entreprises industrielles afin d'optimiser l'information.

Concrètement, la mise en place des systèmes de gestion de maintenance informatisée entraîne :

- Une méthode plus écologique
- La réduction du temps consacré à la maintenance préventive
- La diminution des heures supplémentaires (pannes réparées en dehors des heures normales)
- La diminution des pertes de productions dues aux pannes (manque à gagner)
- La diminution du temps consacré à la gestion administrative du service maintenance
- Prolongation de la durée de vie des équipements grâce à une maintenance préventive mieux gérée

2.6 Présentation de la situation initiale

2.6.1 État de l'art

2.6.1.1 Architecture backend

Parmi les solutions informatiques que CLARISYS INFORMATIQUE édite, nous nous intéressons, dans ce rapport, au progiciel MCA « Manager Connexion Automate ».

MCA est un middleware² expert, conçu en application web, accessible via un navigateur web. Il assure le pilotage et le traitement de l'activité analytique d'un laboratoire, y compris celle de bactériologie. Il se charge également d'établir les différents types de connexions entre les différents analyseurs et de traiter les envois de demandes d'analyse vers ces derniers et le retour de résultats.

MCA est développé en utilisant le Framework Django³ pour sa partie « Backend⁴ » ; En effet, il s'agit d'un Framework robuste, qui, s'il est correctement exploité, permet de produire un projet solide en très peu de temps. Malgré le fait qu'il ait vu le jour en 2003, Django figure aujourd'hui largement parmi les plus utilisés. Régulièrement mis à jour, c'est un outil qui, pour le moment, rivalise aisément avec la plupart des autres outils existants.

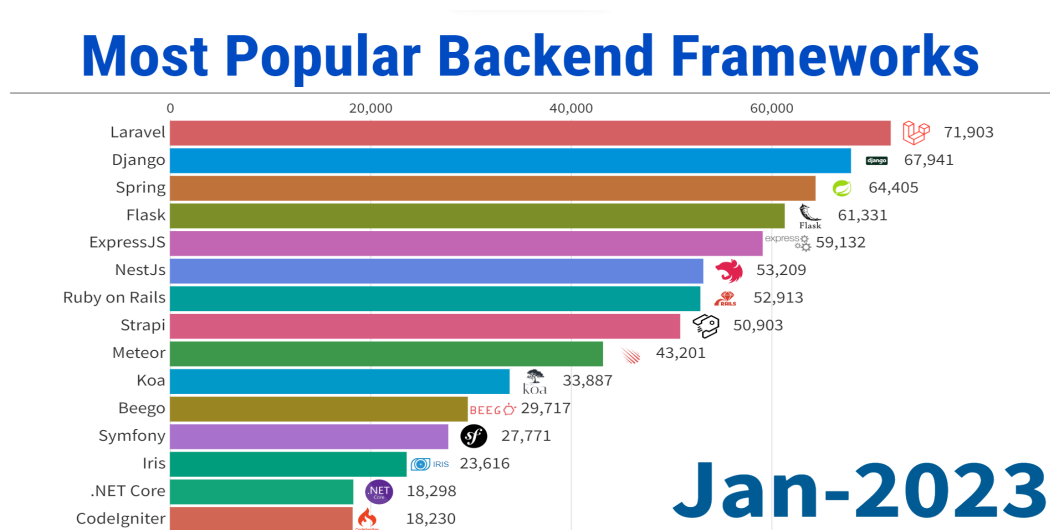


FIGURE 2.2 – Frameworks back end les plus populaires en Janvier 2023

2. En architecture informatique, un middleware ou intergiciel est un logiciel tiers qui crée un réseau d'échange d'informations entre différentes applications informatiques.

3. Django est un cadre de développement web open source en Python. Il a pour but de rendre le développement web 2.0 simple et rapide.

4. En informatique, un back-end est un terme désignant un étage de sortie d'un logiciel devant produire un résultat. On l'oppose au front-end qui lui, est la partie visible.

Django possède de nombreux avantages. Il est facile à prendre en main, ce qui représente un avantage pour de nombreux développeurs. Riche en fonctionnalités, Django peut donc s'adapter à n'importe quel type de projet web. Autre élément crucial pour CLARISYS INFORMATIQUE, il s'agit d'un outil sécurisé, qui aide ses utilisateurs à éviter de nombreux problèmes de sécurité comme l'injection SQL⁵. Django repose sur l'architecture MVT (Model, View, Template), elle-même inspirée du modèle MVC (Model, View, Controller) plus largement connu et utilisé par la plupart des autres frameworks web.

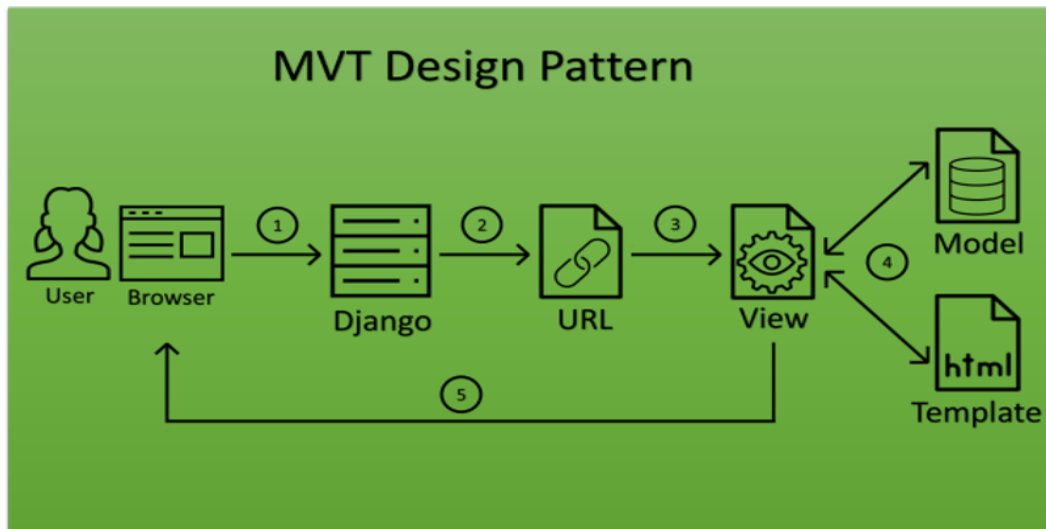


FIGURE 2.3 – Schéma de l'architecture MVT de Django

L'architecture MVT de Django est composée de trois couches :

- o Le Modèle : Le Modèle définit la structure des données et les représente sous forme d'objets ou classes d'objets. C'est la couche qui interagit avec la base de données via un ORM⁶.
- o La Vue : La Vue contrôle ce que l'utilisateur peut consulter. Elle récupère les requêtes web et renvoie la réponse correspondant à chaque requête.
- o Le Template : Le Template définit l'interface graphique que consulte l'utilisateur. Il permet de contrôler l'interaction avec l'utilisateur et gère la façon dont la réponse à une requête est retournée à celui-ci.

5. La faille SQLi, abréviation de SQL Injection, soit injection SQL en français, est un groupe de méthodes d'exploitation de faille de sécurité d'une application interagissant avec une base de données.

6. ORM signifie Object-Relational Mapping. Un ORM est un ensemble de classes permettant de manipuler les tables d'une base de données relationnelle comme s'il s'agissait d'objets.

2.6.1.2 Architecture REST

Dans l'architecture Backend de MCA, nous pouvons constater qu'au-delà de l'architecture classique de Django, il existe de nombreux modules applicatifs de MCA qui sont fournis aussi en « REST API⁷ ». En effet, il s'agit de la librairie Django REST Framework écrite en Python. DRF se base sur le Framework Django pour concevoir des APIs REST. L'avantage principal de l'architecture REST de Django REST Framework comparé à Django MVT, est que l'architecture REST permet de séparer la partie Backend (la partie concernant le serveur, les données, l'authentification. . .) de la partie Frontend⁸ (la partie visible de l'application). REST étant une approche distribuée, les applications client et serveur doivent être complètement découplées (indépendantes) les unes des autres, quel que soit l'endroit d'où proviennent les demandes afin de minimiser les interactions.

La seule information que l'application client doit connaître est l'identifiant de ressource uniforme (URI) de la ressource demandée. Elle ne peut interagir avec l'application serveur d'aucune autre manière.

Toute demande effectuée par un consommateur sera, soit acceptée, soit rejetée par le serveur. De même, une application serveur doit permettre aux applications clientes d'accéder aux données demandées via HTTP sans aucune modification.

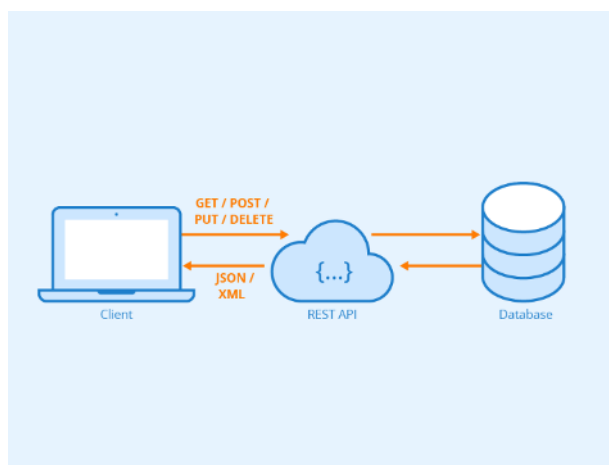


FIGURE 2.4 – Schéma de l'architecture REST API

Ce découplage de deux parties de l'architecture d'une application web permet d'intégrer plusieurs clients qui peuvent consommer le REST API. Le rôle premier d'une API REST est de servir d'intermédiaire entre le client et le serveur. C'est-à-dire que c'est elle qui réceptionne

7. L'API REST (Representational State Transfer - Application Program Interface) est un style architectural qui permet aux développeurs de créer des services web. REST utilise des méthodes HTTP pour récupérer et publier des données entre un appareil client et un serveur.

8. Le développement frontend correspond aux productions HTML, CSS et JavaScript d'une page internet ou d'une application qu'un utilisateur peut voir et avec lesquelles il peut interagir directement.

les requêtes émises par le client, les transmet à l'entité demandée au serveur, prend les réponses données par ce dernier puis les retransmet au client. En effet, en faisant appel à une URI, nous pouvons récupérer une quantité importante de données. Cela offre de grands avantages pour les deux parties : les clients peuvent augmenter la performance de leurs applications et de leurs sites. Quant aux serveurs, ils gagnent en performance car ils ne sont plus amenés à charger l'interface graphique de l'application web. En revanche, le serveur s'occupe de transmettre les données demandées en format JSON⁹ ou XML¹⁰ par la requête de client et ce dernier s'occupe de l'affichage graphique. Il faut également savoir qu'une seule API REST peut facilement alimenter plusieurs applications. Ce qui veut dire que le temps et la ressource utilisés sont fortement réduits.

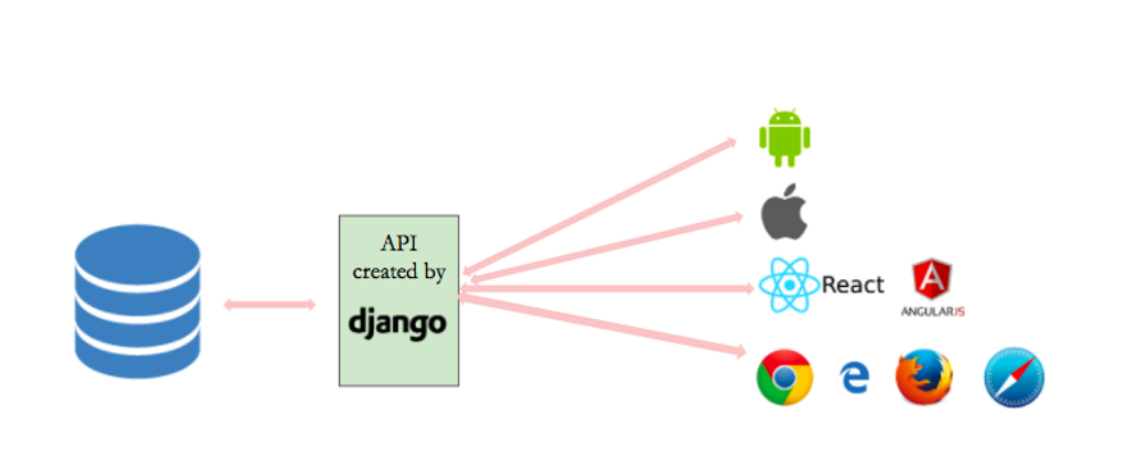


FIGURE 2.5 – Description d'un REST API

De plus, c'est une manière idéale de sécuriser les applications pour les deux parties concernées car les interactions directes pouvant entraîner de fausses manipulations sont évitées au maximum.

9. JavaScript Object Notation est un format de données textuelles dérivé de la notation des objets du langage JavaScript. Il permet de représenter de l'information structurée.

10. XML (eXtensible Markup Language) est en quelque sorte un langage HTML amélioré permettant de définir de nouvelles balises. Il s'agit effectivement d'un langage permettant de mettre en forme des documents grâce à des balises.

2.6.1.3 Le modèle de maturité de Richardson

Développé par Leonard Richardson, ce modèle permet de découper les contraintes de REST en trois étapes principales afin de mettre en application la théorie de REST en tant que service web.

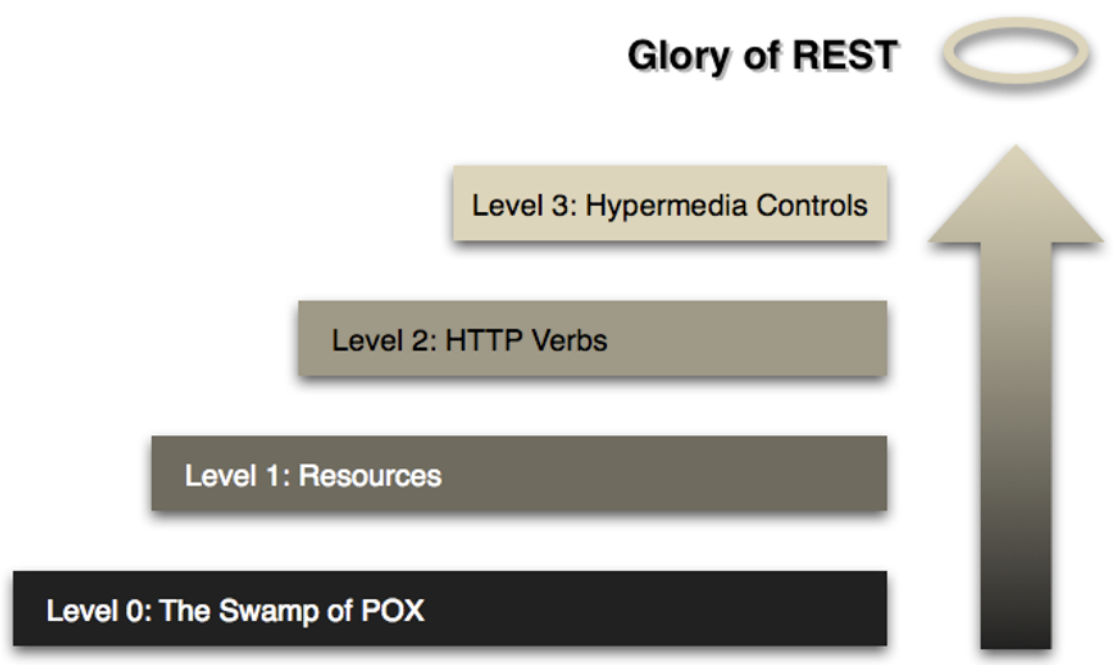


FIGURE 2.6 – Le modèle de maturité de Richardson

Dans son article intitulé « REST APIs must be hypertext-driven » [3], Roy T. Fielding¹¹ a décrit les points fondamentaux d'un REST API après avoir indiqué que les principes de REST ne sont pas toujours respectés. Nous pouvons trouver parmi ces points la notion de « Hypermedia-drive API's » ou plus précisément HATEOAS (Hypermedia as the Engine of Application State). Pour bien utiliser l'HATEOAS, il est important de comprendre deux principes essentiels :

- les URLs Immutables, ou les liens qui retournent toujours les mêmes ressources
- la navigation API, ou la modélisation en arbre avec des transitions.
- Les states sont les pages web et les transitions sont les liens hypertextes.

L'idée derrière le HATEOAS est donc très simple. Pour chaque réponse du serveur nous incluons des liens vers les prochaines requêtes pour explorer les autres ressources.

11. Roy Thomas Fielding : (né en 1965) est un informaticien américain, l'un des principaux auteurs de la spécification HTTP et à l'origine du style architectural Representational State Transfer (REST). Il fait autorité en matière d'architecture de réseau informatique et a cofondé le projet Apache HTTP Server .

Prenons l'exemple d'une réponse JSON d'un serveur suite à une requête de récupération des utilisateurs (Réponse classique non-HATEOAS).

```
[
  {
    "uid": 1,
    "age": 18,
    "name": "John Doe",
    "order": 1
  },
  {
    "uid": 2,
    "age": 29,
    "name": "Bob Marley",
    "order": 2
  }
]
```

Et si un client qui consomme cet API veut en savoir plus sur les utilisateurs ? Doit-il préparer à l'avance ses URLs qui mènent vers les ressources de détails de chaque utilisateur ? Et si l'API évolue au cours du temps et les URLs changent ? Que faire ?

La solution serait donc que l'API fournisse au client des liens dans la réponse, semblables à des hyperlinks dans un site web. Le premier avantage est que le client sera complètement dissocié de l'API de service et qu'il n'a pas à hard-coder les URLs pour interagir avec l'API et l'application state.

```
[
  {
    "uid": 1,
    "age": 18,
    "name": "John Doe",
    "order": 1,
    "href": "/user/1"
  },
  {
    "uid": 2,
    "age": 29,
    "name": "Bob Marley",
    "order": 2,
    "href": "/user/2"
  }
]
```

C'est exactement la même réponse qu'avant, mais avec un champ href en plus, contenant un lien direct vers les ressources détails d'un utilisateur.

Donc, avec cette idée, le client n'a pas à se rappeler des valeurs des urls, notamment quand l'API utilise les URLs immutables qui changent à chaque révision. La solution est donc d'ajouter des informations supplémentaires dans notre réponse JSON via l'attribut href qui a comme tâche de nommer les URLs.

Ainsi, notre réponse embarque toutes les URLs permettant d'explorer l'API pour un utilisateur et l'attribut href permet de donner un sens aux URLs et facilite la gestion côté client.

Les avantages des HATEOAS sont nombreux :

- o Réduction des erreurs de codage - Côté client : En raison de l'accent mis sur l'utilisation des URI (identificateurs de ressources uniques), environ 90% des bogues peuvent être éliminés grâce à HATEOAS. En général, les erreurs sont commises en omettant des segments de chemin d'accès ou en les plaçant dans le mauvais ordre. Parfois, même les codeurs oublient de coder les URL. Ces erreurs humaines courantes peuvent être minimisées en utilisant une approche HATEOAS

- o Mise à niveau détaillée de l'application sans rupture du client : les clients d'une API REST sont généralement programmés avec certaines hypothèses concernant le fonctionnement du système et ses limites. Les API peuvent évoluer assez rapidement si le développeur documente correctement une restriction à laquelle il faut prêter attention et cibler spécifiquement certains aspects du système. Cela s'accompagne d'une logique solide côté serveur pour l'ajout de choses à un stade ultérieur. Un bon développeur peut également s'assurer que tout cela ne perturbe pas les comportements précédemment codés. Grâce à cette approche, les API peuvent évoluer assez rapidement

Cela permet également de s'assurer que tous les clients restent en état de marche. Elle permet également de réduire la charge de travail du développeur, en évitant de multiples versions d'une API sur le serveur. C'est sans doute l'un des principaux avantages de la mise en œuvre d'API conforme à HATEOAS.

2.6.1.4 Docker

Au-delà du Framework Django, qui donne un cadre pour le développement, MCA utilise d'autres technologies, notamment la plateforme de conteneurs Docker qui a récemment été mise en place afin de simplifier son installation.

Docker est une technologie qui permet de distribuer et de déployer des application en les plaçant dans des conteneurs qui se comportent comme des machines virtuelles.

A la différence de ces dernières, les conteneurs docker s'exécutent nativement sur les systèmes et sont donc très efficaces en ressources utilisées. Ils sont auto-suffisants, ce qui signifie qu'ils contiennent tout ce dont l'application a besoin pour fonctionner, sans faire d'hypothèse sur la configuration de la machine qui les héberge.

Docker permet donc de rassembler dans un conteneur l'ensemble des fichiers et services dont l'application a besoin pour fonctionner.

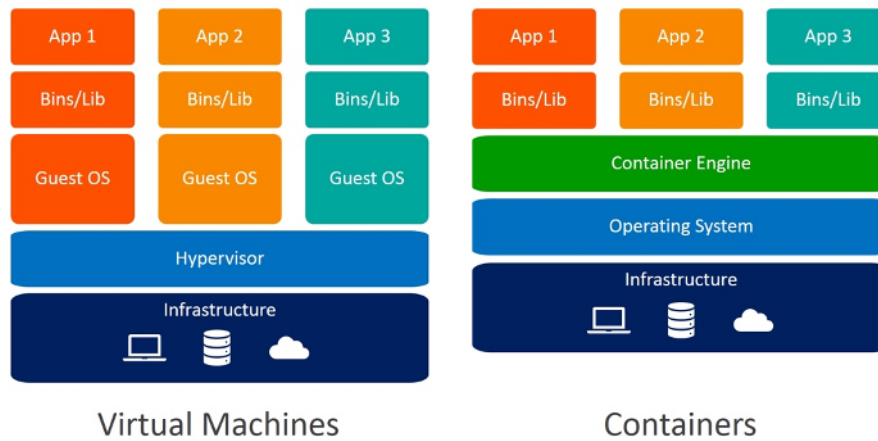


FIGURE 2.7 – Schéma simplifié d'une application virtualisée VS application conteneurisée

Lancé en 2013, Docker permet de facilement créer des conteneurs et des applications basées sur les conteneurs. Même s'il existe d'autres solutions, Docker est largement plus utilisé, notamment car il est plus facile à utiliser et à déployer que ses concurrents.

Ainsi, grâce à la conteneurisation, l'installation de MCA est devenue bien plus facile puisqu'il suffit d'effectuer quelques commandes pour pouvoir accéder à un environnement complet ainsi qu'une base de données et un logiciel fonctionnel. Cela permet également au développeur de créer, supprimer et recréer des conteneurs aisément.

2.6.1.5 Architecture Frontend

Au cours de la vie de MCA, l'interface graphique a subi plusieurs refactorisations¹² concernant les technologies du Frontend. En effet, le développement Frontend a connu une forte hausse notamment ces dix dernières années.

Ainsi, les technologies et les Frameworks aident les développeurs à accélérer le développement et la mise en forme des interfaces des applications web grâce à des écosystèmes robustes.

Les premières versions de MCA n'utilisaient pas de Frameworks de Frontend et se contentaient de la traditionnelle association HTML, CSS, JavaScript. Depuis, en vue d'optimiser l'application ainsi que l'expérience utilisateur, l'équipe technique de Clarisys Informatique a étudié l'avantage d'inclure le Frontend dans un Framework plus moderne et performant.

Les Frameworks de Frontend permettent aux développeurs de se concentrer uniquement sur la partie métier puisque toutes les couches techniques sont déjà intégrées dans le Framework. C'est un gain de temps pour le développement et donc aussi pour le projet.

L'architecture permet la séparation des couches techniques logiques afin de faciliter le développement en équipe, la maintenance et l'évolution. Par exemple, un développeur ne travaillera pas sur la même couche qu'un intégrateur. Cela permet de paralléliser les tâches et d'éviter les conflits dans la gestion des sources.

La maintenance et l'évolution du framework sont gérées par l'organisme fondateur. Ce n'est pas l'équipe de développement qui aura la charge de le maintenir. Tout ce temps économisé pourra être dépensé en recherche et développement et apporter de la valeur ajoutée au projet.

Le seul inconvénient réel qui pourrait freiner l'utilisation d'un Framework est le temps d'apprentissage. C'est le prix à payer car la prise en main peut parfois s'avérer fastidieuse. Mais une fois le Framework maîtrisé, elle ne peut qu'améliorer les développements.

12. La refactorisation ou le réusinage de code est l'opération consistant à retravailler le code source d'un programme informatique sans toutefois y ajouter des fonctionnalités ni en corriger les anomalies

Les critères à prendre en compte dans le choix avant même de tester le framework (ces critères valent pour le choix de n'importe quelle technologie) :

- La maturité du framework : il vaut mieux choisir des technologies qui sont là depuis longtemps et qui ont déjà fait leurs preuves plutôt qu'un « petit nouveau » qui va peut-être devoir résoudre des défauts de jeunesse. Autre avantage : une technologie mature a eu le temps de rassembler une grande communauté d'utilisateurs/développeurs prête à venir en aide aux membres en difficultés grâce aux forums de discussion, à la documentation, aux ressources
- La rapidité de mise en œuvre, c'est-à-dire le temps nécessaire pour l'installation du framework et celui entre l'installation et le moment où le framework est opérationnel
- La courbe d'apprentissage, qui est le temps nécessaire pour apprendre à maîtriser le framework
- L'acceptation par le framework des conventions de programmation et des best practices
- La capacité du framework à s'adapter à l'évolution des technologies : autrement dit, mature ne veut pas dire « fossilisé ». Il faut que le framework évolue avec son temps
- La capacité du framework à s'adapter à la charge, c'est-à-dire à l'augmentation du nombre d'utilisateurs

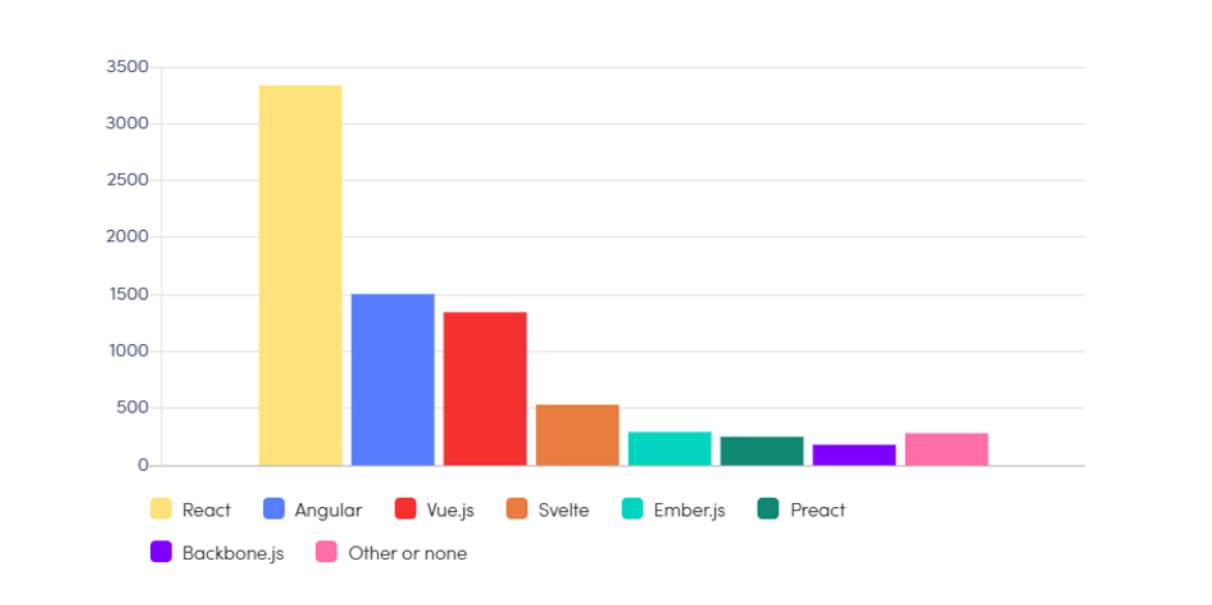


FIGURE 2.8 – Frameworks Frontend les plus populaires en Janvier 2022

2.6.1.6 VueJS



Le module de la maintenance est un nouveau module à développer en MCA. L'équipe technique de Clalrisys Informatique a choisi le Framework VueJS pour le développement de la nouvelle interface utilisateurs de MCA pour les nouvelles versions à sortir. Notamment la versions 9.0.

Evan You¹³ a créé VueJS alors qu'il travaillait chez Google en tant que technologue créatif. Ainsi, à certains égards, il est similaire à AngularJs, le framework JavaScript de Google. VueJS est décrit comme « un framework progressif pour la création d'interfaces utilisateur ». Ce qui différencie ce framework des autres frameworks de développement web JavaScript est que son architecture est conçue pour être adaptable de manière incrémentielle.

L'écosystème VueJS se compose d'une bibliothèque de base, de frameworks et d'autres outils qui permettent un développement front-end facile et rapide. Bien que Vue, lui-même, soit léger et minimaliste, il donne tous les outils aux développeurs pour créer des applications web très fonctionnelles.

Le plus grand avantage de VueJS est sa facilité d'intégration dans un projet Web existant. Le passage à ce framework est donc très facile et avantageux pour un environnement de développement rapide.

L'un des principaux avantages de VueJS est sa petite taille. L'utilisateur n'a pas besoin de beaucoup de temps pour télécharger et utiliser ce framework car il ne fait que 18-21 Ko. Toutefois, cette petite taille ne signifie pas une vitesse inférieure car, au contraire, il bat tous ses rivaux volumineux comme Angular, React.JS et Ember.JS.

13. Evan you est un développeur open source indépendant, un designer et un codeur créatif. Il est l'auteur de Vue.js, un framework JavaScript permettant de créer des interfaces web modernes avec des composants réactifs.

2.6.2 L'ancien processus de la maintenance

La première version du module de la maintenance ne prend en compte que les analyseurs (automates) dans un laboratoire. Étant limité aux analyseurs, ce module ne répond plus aux besoins des laboratoires. L'informatisation des processus d'analyses médicales a engendré de nouveaux besoins qui s'interconnectent. Aujourd'hui, de nouveaux équipements rentrent en jeu dans la procédure d'analyses médicales tels que les imprimantes, les scanners, les microscopes, les équipements informatiques, etc.

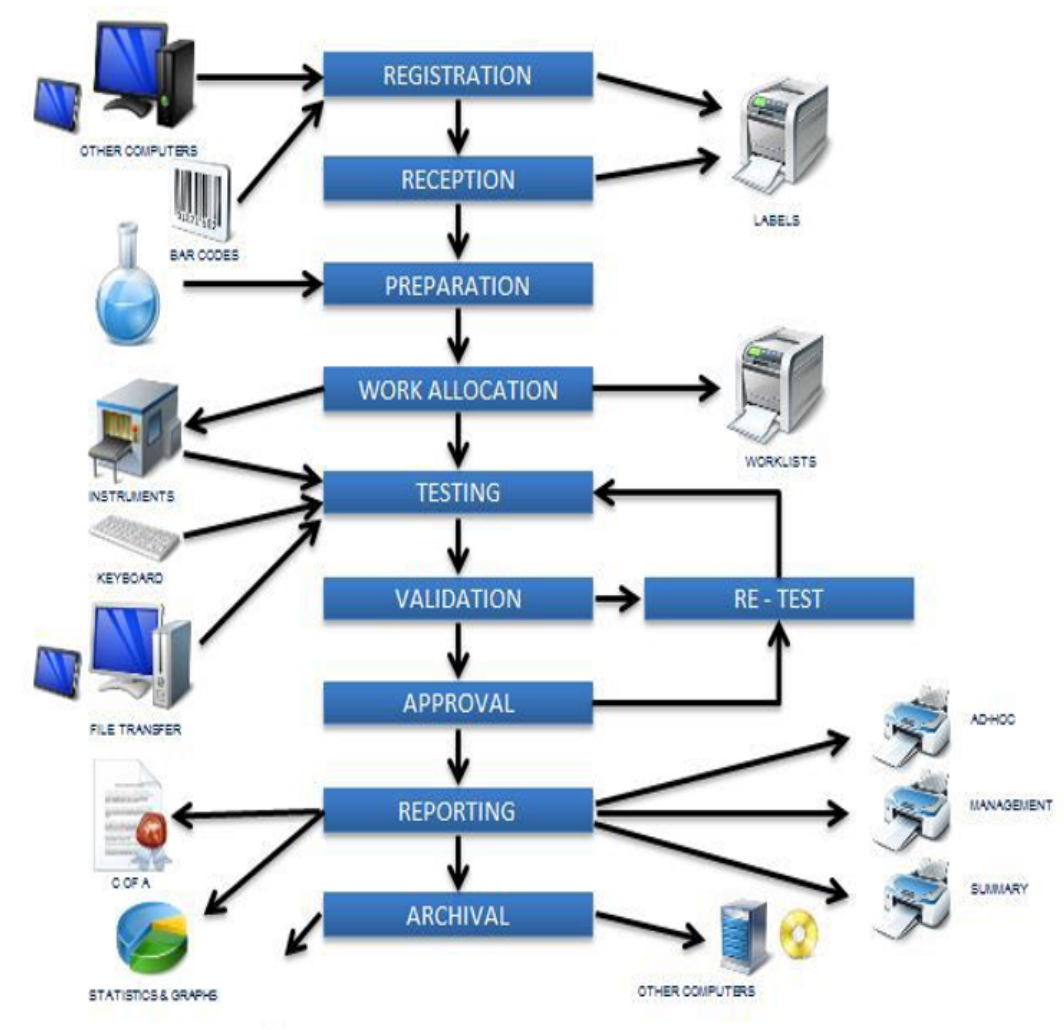


FIGURE 2.9 – Workflow de laboratoire

Le schéma précédent montre que dans un workflow de laboratoire d'analyses médicales, plusieurs équipements médicaux et non-médicaux sont nécessaires pour assurer le bon déroulement de ce dernier. À l'instar des analyseurs, tous ces équipements demandent un entretien particulier pour assurer leur fonctionnement, la sûreté de leurs résultats ainsi que leurs disponibilités en permanence.

C'est pour cette raison que l'évolution du module de la maintenance devient une nécessité pour l'équipe technique de Clarisys Informatique. Avec ce module MCA, elle peut gérer d'une manière plus générale les opérations de maintenance concernant les équipements auxquels il se connecte. Actuellement, si nous voulons planifier une opération de maintenance pour un analyseur, nous pouvons nous rendre à l'interface de planifications de maintenances dans le menu de paramètres de MCA.

Anciennement, si nous voulons planifier une opération de maintenance pour un analyseur, nous pouvons nous rendre à l'interface de planifications de maintenances dans le menu de paramètres de MCA.

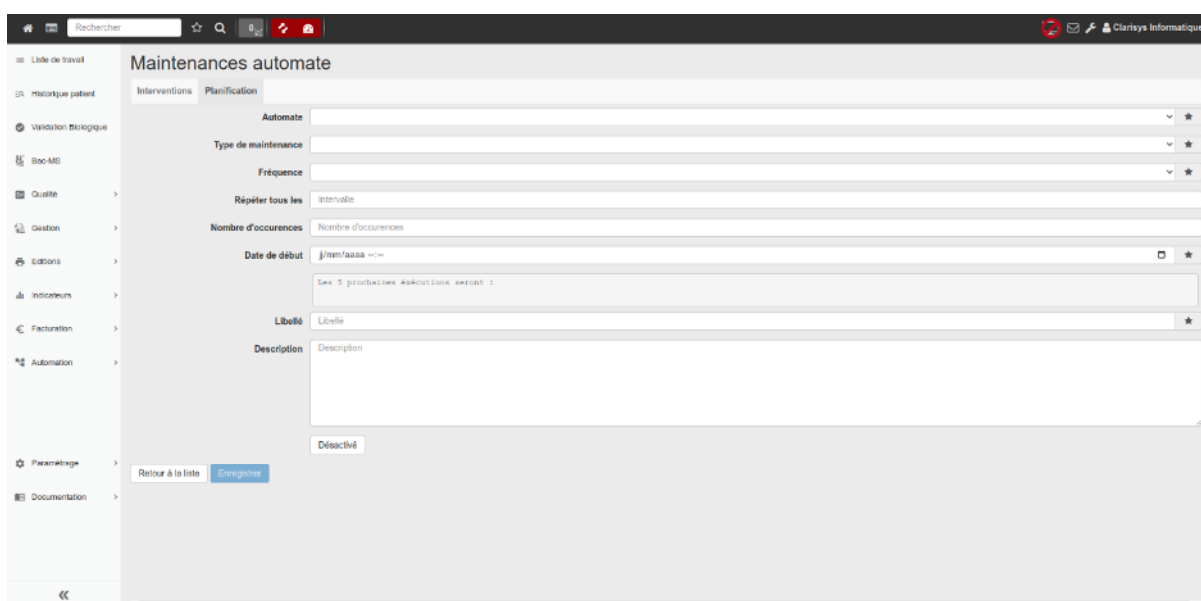


FIGURE 2.10 – L'ancienne interface de paramétrage de maintenances - MCA

Cette interface, ne permet pas à l'agent de maintenance de consulter les informations concernant les analyseurs telles que leurs marques, leurs fournisseurs, leurs contrats de prestations, le contact de l'entreprise de maintenance si prestation externe, etc. De plus, aucune information concernant l'état de l'analyseur n'est consultable. Par exemple : la date de la dernière intervention de maintenance faite sur l'analyseur, les interventions à venir et leurs dates, les interventions précédentes et leurs informations. Ces informations sont très importantes pour établir ce que l'on appelle le registre de vie ou le cahier de vie d'analyseur, dans lequel, on retrouve les registres ou grilles de maintenances mis à jour par les techniciens de maintenances.

En conclusion, une évolution de ce module permet, d'une part, de mieux comprendre le fonctionnement du laboratoire d'analyses médicales et d'autre part d'aider les agents de maintenance à organiser les opérations de maintenance non seulement pour les automates mais pour tout type d'équipements de laboratoire.

2.7 Proposition d'une nouvelle solution

2.7.1 Gestion de l'inventaire matériel

La première étape consiste à créer une base de données exhaustive de tous les équipements qui nécessitent une maintenance. Chaque équipement doit être enregistré avec des informations telles que le nom, le modèle, le numéro de série, la date d'installation, les caractéristiques techniques, etc.

2.7.2 Planification des activités de maintenance

Une fois que l'inventaire des équipements est établi, les activités de maintenance préventive et corrective doivent être planifiées. La GMAO permet de créer des calendriers de maintenance régulière pour chaque équipement en fonction des recommandations du fabricant, des normes industrielles ou des exigences spécifiques de l'entreprise.

2.7.3 Gestion des interventions

Lorsqu'une intervention de maintenance est nécessaire sur un équipement, une demande d'intervention est générée dans la GMAO. Cette demande est ensuite assignée à un technicien ou une équipe de maintenance appropriée.

2.7.4 Suivi des interventions

Les techniciens utilisent la GMAO pour enregistrer les détails de chaque intervention effectuée sur un équipement. Cela inclut les actions entreprises, les pièces de rechange utilisées, le temps passé sur l'intervention, etc.

2.7.5 Historique des matériels

La GMAO conserve un historique complet de toutes les interventions et activités de maintenance effectuées sur chaque matériel. Cela permet d'analyser les tendances de défaillance, de prévoir les besoins futurs en maintenance et d'optimiser les processus.

2.7.6 Indicateurs de performance

Une GMAO peut fournir des rapports et des tableaux de bord pour suivre les indicateurs de performance clés (KPI) liés à la maintenance des équipements. Ces KPI peuvent inclure le taux de disponibilité des équipements, le temps moyen entre les défaillances, le taux de réussite des interventions, etc.

2.7.7 Conception de base de données

Pour bien comprendre ce qui a été fait et savoir d'où je dois partir, il m'a fallu analyser les différentes tables présentes en base de données.

Pour mettre au propre cette partie, j'ai modélisé un diagramme d'entité association qui reprend partiellement les tables de l'ancien module GMAO. J'ai réalisé seulement une portion du diagramme d'entité association puisque le module GMAO contient beaucoup de tables dont un certain nombre qui ne concernent pas les évolutions que je dois y apporter.

De plus, le diagramme n'est pas complet. J'ai donc reproduit le diagramme d'entité association uniquement pour la partie qui concerne la mise en place de la gestion de l'inventaire matériel.

L'objectif de ce projet est de fournir un système destiné à gérer le parc matériels des laboratoires médicaux. Ce système fait partie de l'application MCA. Il est accessible via un navigateur.

Le module permet d'avoir une vue globale sur le parc matériels, de gérer et de contrôler le travail des techniciens sur le terrain tout en centralisant les informations sur les matériels se trouvant dans les différents sites de laboratoire.

Le module permet aux techniciens sur le terrain de renseigner leurs activités, d'avoir les informations sur les matériels et aux usagers de déclarer les pannes.

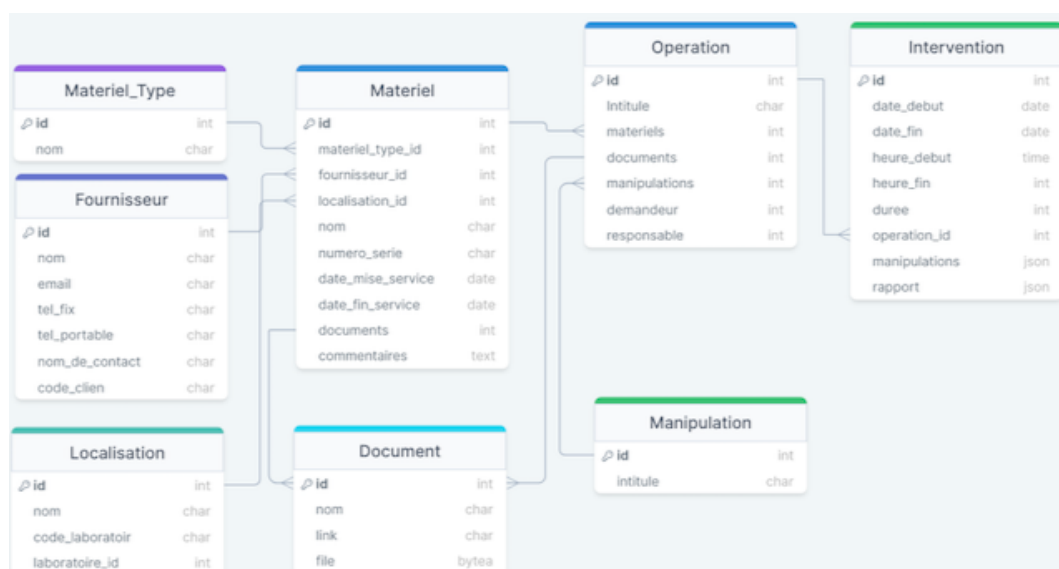


FIGURE 2.11 – Modèle entité-association de la solution GMAO

Les matériels sont attachés à un type pour faciliter le filtrage et pour regrouper les différents matériels à une famille de type. Les informations des fournisseurs sont aussi attachées aux matériels.

Les agents de maintenance peuvent planifier plusieurs maintenances en même temps, ainsi, ils peuvent intervenir sur plusieurs maintenances et générer les rapports d'interventions.

Aussi, il est possible de télécharger plusieurs fichiers qui seront utiles au moment de la maintenance : guide d'utilisation de matériel, documentation technique de constructeur, etc.

Dans le module GMAO, les techniciens de laboratoire pourront ajouter, modifier ou supprimer des matériels.

Il faudra développer une interface composée d'un tableau qui reprend tous les matériels enregistrés en base de données en indiquant leur type, leur fournisseur et les documents associés.

De cette manière, les utilisateurs verront rapidement l'état de l'inventaire matériel et toutes les informations associées.

Une fois que cette gestion de l'inventaire sera correctement implémentée dans le module GMAO, il me faudra adapter la fonctionnalité de la maintenance. En effet, lors d'une demande de maintenance sur un matériel géré dans le module GMAO, l'utilisateur devra être capable d'indiquer l'indisponibilité de ce matériel sur lequel la maintenance sera effectuée.

2.8 Le développement

2.8.1 Gestion de l'inventaire matériel

La gestion de l'inventaire est l'un des aspects essentiels de la GMAO. Elle consiste à suivre et à contrôler les matériels afin de maintenir les actifs en bon état de fonctionnement. Les responsables et les techniciens de la maintenance peuvent, à l'aide d'un formulaire dédié, de faire rentrer dans le système les nouveaux matériels.

Pour cela, un module a été créé pour que les administrateurs puissent ajouter, modifier ou supprimer des matériels. Ce module est constitué d'un formulaire dans lequel l'administrateur saisira les informations nécessaires à la création d'un matériel. Ces informations sont le numéro de série, la famille d'appartenance, le nom, le modèle de matériel auquel il est rattaché, le fournisseur, une description, la localisation dans laquelle ce matériel se situe au sein des différents sites de laboratoire et les dates d'expiration (si pertinent)

Ajouter un nouveau matériel

| | | | |
|-----------------------------------|--------------------------------|------------------------------|---------------------|
| Type de Matériels + Imprimante | Nom de matériel HP Lte 4500 | Numéro de série 963258741 | Fournisseur HP + |
| 01/07/2023 | 31/07/2023 | | |
| Localisation Casamozza | 2 files | | |

Commentaires

Normal B I U S = = = = = = A

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed ullamcorper augue augue, a elementum quam lobortis a. Suspendisse tincidunt elit quis ex finibus suscipit. Vestibulum sollicitudin mattis erat at rhoncus. Quisque lobortis porta leo posuere vehicula. Cras orci tellus, scelerisque non massa quis, sollicitudin tristique mauris. Duis euismod interdum tincidunt. Aenean ullamcorper lacinia massa at sagittis. Maecenas molestie libero non urna tempus, eget vehicula lacus sollicitudin. Donec cursus lectus mollis, ornare leo et, elementum dolor. Vivamus a urna ac est gravida pharetra. Nam non sem eu urna ullamcorper egestas in ut risus. Integer eleifend varius risus vitae suscipit. Nam vel mauris vitae urna congue lobortis vel sit amet mauris. Pellentesque luctus luctus nunc, non elementum arcu vehicula at. Donec lobortis scelerisque ex, id fringilla ligula fringilla quis. Aliquam scelerisque quam a eleifend tincidunt. Sed enim ligula, convallis sit amet ultrices vel, porta eu ligula. Nulla maximus nibh eget lorem consectetur vestibulum. Ut sit amet nulla varius, pretium libero sed, vulputate velit. Nulla facilisi.

FERMER SAUVEGARDER

FIGURE 2.12 – Formulaire de saisie de matériel

2.8.2 Planification des activités de maintenance

La planification des activités de maintenance dans une GMAO est un processus complexe qui nécessite une attention méticuleuse aux détails et une collaboration étroite entre les différents acteurs impliqués dans la gestion de la maintenance. Une bonne planification permettra d'assurer la disponibilité et la fiabilité des équipements, ce qui contribuera à améliorer l'efficacité opérationnelle et à réduire les coûts de maintenance.

2.8.2.1 Les opérations :

Une opération est un document électronique qui décrit les détails d'une tâche ou d'une activité de maintenance à effectuer. Il s'agit d'une instruction formelle donnée à une équipe de maintenance ou à un technicien pour exécuter des travaux spécifiques sur un équipement, une machine ou un système.

les éléments généralement inclus dans une opération d'une GMAO :

- Identifiant : Un numéro ou une intitulé unique attribué à l'opération pour le suivi et la référence.
- Description de la tâche : Une explication détaillée des travaux à effectuer, y compris les étapes spécifiques et les instructions nécessaires pour accomplir la tâche.
- Équipement concerné : Le nom, le numéro de série et/ou l'identification de l'équipement ou de la machine nécessitant la maintenance.
- Commentaires et remarques : Des notes supplémentaires sur l'état d'avancement des travaux ou d'autres informations pertinentes.

choisir entre Matériel HP LTE 450 ou Analyseur

Intitulé
Nettoyage imprimante

☐ Contrôle Qualité ✖ ☐ Bloquante ✖

Filtrer les manipulations
Aucun resultat trouvé ? Saisissez l'intitulé de la manipulation et appuyez sur Entrer!

| | |
|--------|-------------------------------------|
| Test 1 | <input checked="" type="checkbox"/> |
| test 2 | <input checked="" type="checkbox"/> |
| test 3 | <input checked="" type="checkbox"/> |

Description

Normal **B** **I** **U** **S** **≡** **≡** **≡** **≡** **≡** **A** **✖**

Cras ut ex dolor. In felis orci, luctus in eleifend sit amet, sodales efficitur orci. Nam a tellus sed libero cursus ultrices eget non felis. Phasellus posuere, nunc in posuere gravida, felis risus dictum neque, viverra fermentum libero erat eu diam. Sed sit amet libero vehicula augue dignissim rhoncus. Vestibulum commodo aliquam tortor ac pellentesque. Sed vel iaculis ante, vel viverra velit. Nullam hendrerit metus erat, id tincidunt ligula facilisis a. Donec iacus felis, facilisis non est quis, varius pulvinar nulla. Morbi mollis orci eget lorem posuere tempor. Morbi nec felis sit amet ante malesuada pretium sit amet et lacus. Vestibulum auctor nec ligula nec dapibus. Suspendisse sit amet malesuada nibh, sed tincidunt est. Integer nec volutpat orci.

FERMER **SAUVEGARDER**

FIGURE 2.13 – Formulaire de saisie d'opérations

Ces opérations permettent de suivre et de gérer efficacement les activités de maintenance, d'assurer la communication entre les membres de l'équipe et de garantir que les travaux sont effectués de manière cohérente et méthodique.

2.8.2.2 Les interventions

Les interventions dans une GMAO permettent d'assurer la maintenance préventive et curative des équipements, d'améliorer leur disponibilité et leur efficacité, tout en réduisant les temps d'arrêt non planifiés et en augmentant la durée de vie des actifs de l'entreprise.

Les interventions font référence aux différentes actions de maintenance réalisées sur les équipements, machines ou installations d'une entreprise. Ces interventions sont enregistrées, suivies et gérées par le système GMAO pour assurer un suivi efficace de la maintenance.

Les étapes typiques d'une intervention dans une GMAO :

- Création de la demande d'intervention : Lorsqu'un problème est détecté sur un équipement ou lorsqu'une maintenance préventive doit être effectuée, une demande d'intervention est créée dans la GMAO. Elle peut être générée automatiquement par le système en fonction des plannings de maintenance ou saisie manuellement par un opérateur.
- Assignation de l'intervention : La demande d'intervention est attribuée à un technicien ou une équipe de maintenance appropriée. L'affectation peut être basée sur les compétences requises, la disponibilité du personnel et la localisation géographique.
- Planification : Une fois l'intervention attribuée, elle est planifiée dans le calendrier de maintenance. La date et l'heure de l'intervention sont définies en tenant compte des priorités, de l'urgence et de la disponibilité des ressources.
- Exécution de l'intervention : Le technicien ou l'équipe se rend sur le site de l'intervention et effectue les tâches de maintenance requises. Cela peut inclure des réparations, des remplacements de pièces, des vérifications, des réglages, etc.
- Saisie des données : Pendant l'exécution de l'intervention, le technicien enregistre les informations pertinentes dans la GMAO. Cela peut inclure les heures de travail, les pièces utilisées, les observations sur l'état de l'équipement, les actions effectuées, etc.
- Validation : Une fois l'intervention terminée, le travail est vérifié et validé par un responsable ou une équipe de contrôle de qualité pour s'assurer que toutes les tâches ont été correctement exécutées.

2.8.3 Le calendrier des interventions

Les techniciens ou les équipes de la maintenance peuvent visualiser, selon leurs droits d'utilisateur ou leur affectations, un calendrier regroupant les tâches à effectuer dans la journée, la semaine, le mois etc.

Ce calendrier permet la suivie de l'avancements de toutes les tâches de la maintenance en cours, ainsi que une vision globale sur toute les retards potentiels ou les changements de dates heures etc..

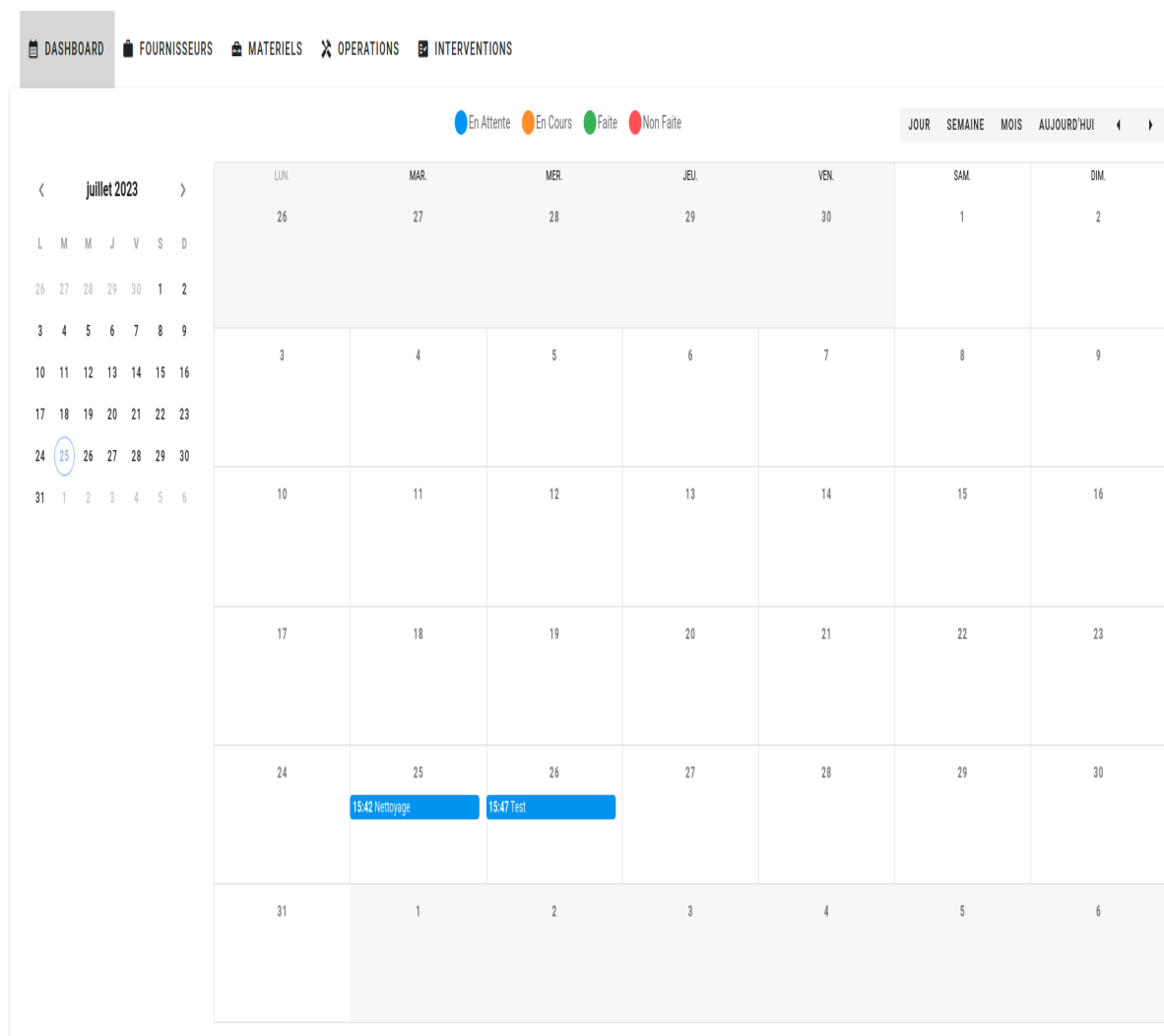


FIGURE 2.14 – Le calendrier des interventions

Chapitre 3

La gestion de projets

3.1 L'équipe projet

Clarisys Informatique est une entreprise mêlant des personnes issues du domaine de l'informatique et de la biologie médicale. Elle est divisée en deux équipes principales qui collaborent sur la production de deux logiciels commercialisés par l'entreprise, auxquelles s'ajoutent la direction et l'administration, les chefs produits et projets ainsi que les administrateurs systèmes.

Tout d'abord, on retrouve l'équipe support, principalement constituée d'anciens salariés de laboratoires qui s'occupe de l'installation, de la formation ainsi que du support et du dépannage des clients. Il s'agit de l'équipe la plus orientée vers le métier de l'entreprise et celle qui est le plus en contact avec le produit ainsi qu'avec les clients au quotidien.

La deuxième équipe, plus technique, est l'équipe R&D. Composée de personnes ayant reçu une formation en informatique, ses membres sont davantage éloignés des aspects fonctionnels. Il s'agit de l'équipe au sein de laquelle j'opère en tant qu'alternant.

Puisque Clarisys possède deux produits principaux, et même si ceux-ci utilisent des technologies communes, l'équipe de développement est scindée en deux groupes. L'un d'eux travaille sur MCA et l'autre sur Clarilab. J'effectue mes missions d'alternance au sein de l'équipe chargée du développement de MCA et son écosystème.

Pour ce projet, j'ai été amené à contacter différentes parties prenantes de l'entreprise. Tout d'abord, le chef produit afin d'obtenir les spécifications sur les fonctionnalités métier qu'il m'incombait de développer. J'ai également organisé des réunions de brainstorming avec mon maître d'apprentissage, le Lead Dev, ainsi que le directeur de la stratégie produits, anciennement biologiste, maître d'œuvre des projets.

J'ai donc eu, au cours de ce projet, l'occasion de travailler souvent en collaboration avec les autres membres de l'équipe notamment avec mon maître d'apprentissage et le Lead Dev.

3.2 Outils de gestion de projet

L'équipe de développement ainsi que l'équipe support utilisent la plate-forme GitLab, qui fournit plusieurs modules de gestion de projet.

GitLab est une solution collaborative open-source de développement qui permet d'héberger un projet, facilitant la gestion du code-source ainsi que des outils qui favorisent une approche agile de la gestion de projet.

Aujourd'hui, GitLab est devenu un outil central au sein de Clarisys Informatique. En effet, il offre la possibilité, en plus d'héberger le code et de permettre à chaque personne ayant les droits de le modifier, de créer des issues.

Lorsqu'un problème est soulevé, comme une régression par exemple, une amélioration est imaginée. Si une fonctionnalité doit être ajoutée, une issue est créée.

Celle-ci doit décrire précisément le problème à résoudre et peut comporter des captures d'écran, des commentaires ou encore des références vers d'autres issues.

Une fois l'issue créée, elle passe par différentes étapes jusqu'à être intégrée au reste du code du projet et mise en place en production. Tout d'abord, elle doit être analysée. C'est-à-dire que le développeur ou le chef de projet doit définir précisément ce qu'il y a à faire, ainsi que le temps estimé pour l'issue et éventuellement une date de rendu.

L'estimation est réalisée à titre indicatif afin de donner une idée générale du type d'issue dont il s'agit (durée plutôt courte d'une heure ou alors gros projet s'étendant sur plusieurs jours voire plusieurs mois).

Une fois l'analyse faite, l'issue passe au statut "À développer". Cela signifie qu'elle doit être prise en charge par l'un des développeurs de l'équipe. Suivant sa criticité et le deadline imposée, elle sera traitée plus ou moins vite.

Lorsqu'elle est prise en charge, l'issue prend le statut "Développement en cours", cela implique qu'un des membres de l'équipe est en train de la traiter.

Le développement sera ensuite en attente de relecture lorsqu'il sera fini et passera au statut "À revoir".

Pris en charge par un autre membre de l'équipe chargé d'apporter un œil extérieur sur le travail effectué, de corriger d'éventuelles erreurs ou de proposer des pistes d'améliorations, l'issue passera alors en "Revue en cours".

Enfin, lorsque la relecture est finie, le développement est ajouté au reste du projet et placé en attente d'un environnement de test jusqu'à sa mise en production. L'issue est alors close.

Evolution module de maintenance



Demande de Siemens : Bonjour,

Suite à la demande de 2 clients en cours de prospection, nous souhaiterions connaître la faisabilité d'évolution du module de maintenances qui n'est à ce jour pas assez performant pour une utilisation de routine.

MPL, la solution en place chez ces clients est notre principal concurrent et nous devons apporter une réponse rassurante sur notre capacité à gérer les maintenances.

Voici donc les évolutions que nous souhaiterions apporter au module : Définir grâce à un calendrier des actions à faire : Gestion des actions quotidiennes, hebdomadaires, mensuelles, annuelles Avoir la possibilité d'avoir un rappel de l'action à faire quelques jours avant Avoir une alerte si l'action n'a pas été faite Avoir une vue générale des actions planifiées ou réalisées sur une période choisie

Pouvoir stocker un fichier (PDF, WORD ou autre) afin de l'afficher lors de la maintenance à faire, sinon pouvoir décrire dans l'application la procédure en texte libre

Pouvoir tracer l'action réalisée, qui l'a faite, quand et saisir un commentaire éventuellement

Pouvoir enregistrer des actions particulières (pannes automates, actions impactant l'automate, mise à jour automate, etc ...)

Prévoir la possibilité d'export en fichier type tableau des actions réalisées sur une période choisie

Tous ces éléments doivent être accessibles à partir d'un tableau de bord. L'accès à la fonction doit être paramétrable dans les droits utilisateurs et l'affichage des maintenances automates doit être limité aux automates du périmètre utilisateur. Si vous confirmez la faisabilité, nous aurons également besoin d'une évaluation du temps de développement.

[2020.04.24__Module_maintenances.docx](#)

Edited 2 years ago by Karl ARACTINGI

↑ Drag your designs here or [click to upload](#).

Linked issues 0 +

FIGURE 3.1 – Exemple d'une issue GitLab

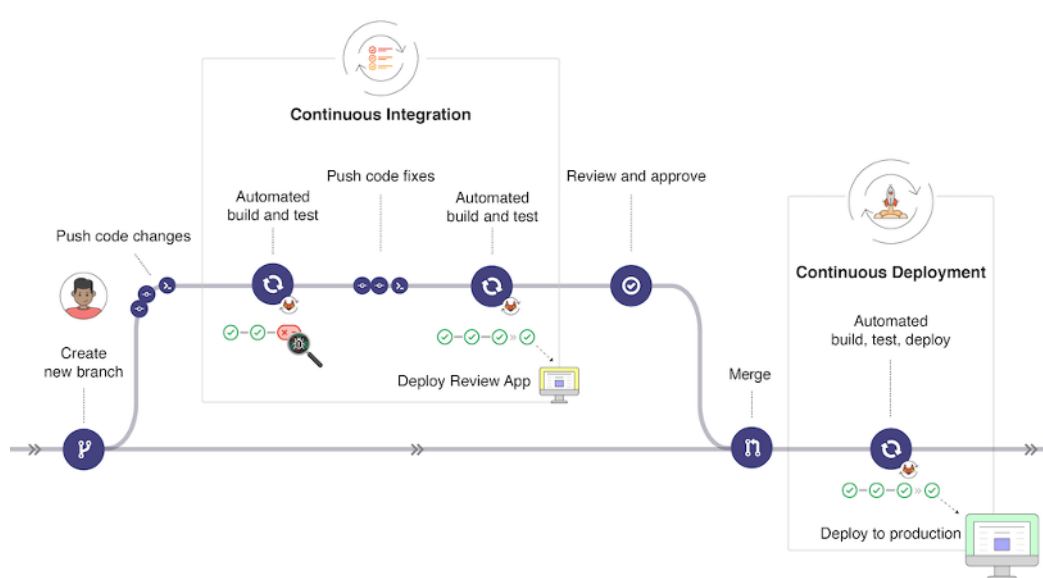


FIGURE 3.2 – Le cycle de vie d'une issue Gitlab

Afin de faciliter une bonne coordination au sein de l'équipe, GitLab propose un outil permettant de regrouper l'ensemble des issues ouvertes. Il s'agit du board. C'est autour de lui que s'articule le Daily Meeting, ayant lieu chaque matin, au cours duquel sont passées en revues les issues.

En plus de donner une vue d'ensemble des issues ouvertes, il permet de voir rapidement à quels membres de l'équipe elles sont affectées, le temps estimé pour chaque issue et le temps réel qui y est passé ainsi que tous les deadlines à prendre en compte.

Utilisé par l'équipe de développement mais également par l'équipe support ainsi que la chef-ferie de projet et de produit, cet outil simplifie grandement la communication entre les collaborateurs.

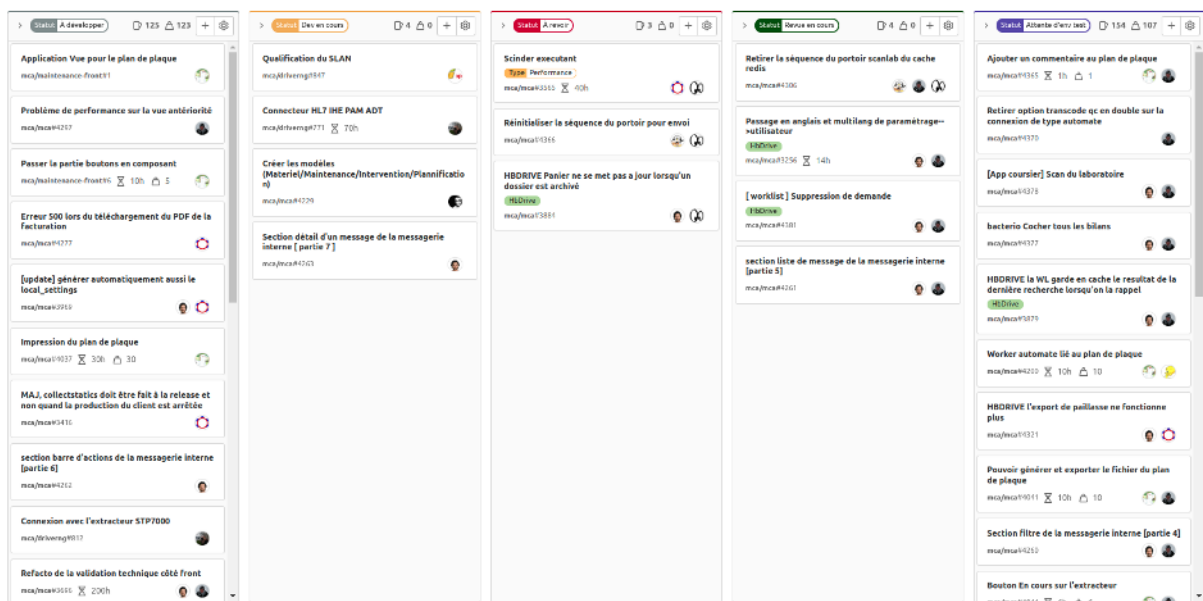


FIGURE 3.3 – Exemple de Board GitLab

Le board proposé par GitLab pourrait être comparé à ceux utilisés dans la méthode agile Kanban. En effet, il s'agit pour Kanban d'un tableau consultable par l'ensemble des membres de l'équipe regroupant les tâches à faire, celles en cours ainsi que les tâches terminées.

Chapitre 4

Bilan personnel du projet

4.1 Les compétences technique

L'intégration d'une équipe performante, possédant de solides connaissances et compétences, m'a permis de me perfectionner dans la maîtrise du Framework Django

J'ai notamment pu en apprendre davantage sur l'architecture des API, avec lesquelles je n'avais que très peu travaillé lors de mon projet l'année précédente.

Le fait de travailler avec cette architecture m'a également permis d'utiliser et de parfaire les connaissances en base de données que j'ai pu acquérir au cours de ma formation.

En effet, la création des vues permettant les communications entre le client et le serveur, m'a offert la possibilité de regarder de plus près le Framework Django, le Framework Django REST Framework ainsi que le Framework VueJs.

Au cours de ce projet, j'ai pu faire face à des problématiques d'intégrations de nouvelles technologies dans une architecture préexistante ; MCA s'étant construit bien avant mon arrivée à Clarisys Informatique.

Au cours de cette année, il a aussi fallu que j'apprenne à travailler avec un nouvel environnement, à savoir Docker. C'était la première fois que j'avais l'opportunité de travailler sur un environnement entièrement conteneurisé.

De manière générale, j'ai pu au cours de cette année en apprendre beaucoup sur MCA et son fonctionnement, notamment au sujet des processus de mise à jour. Même s'il s'agit d'un logiciel extrêmement vaste et parfois complexe, je me sens chaque jour plus à l'aise avec son architecture ainsi qu'avec les termes métiers qui lui sont associés.

Ce projet m'a globalement appris à améliorer mes capacités de recherche lorsque je rencontrais un problème et qu'il fallait parcourir la documentation existante par exemple, mais également lorsqu'il fallait chercher des éléments de réponse directement dans le code de MCA. J'ai également

pu mettre en place, concrétiser, tester des solutions de résolution face à des problèmes et mobiliser les compétences acquises via ma formation ainsi que mon expérience d'alternant.

Le projet de cette année a donc été riche d'enseignements techniques et m'a offert la possibilité de monter en compétence et de gagner en autonomie dans le domaine du développement Web FullStack.

J'ai pu explorer davantage les possibilités offertes par le Framework Django ainsi que le Framework Vuejs et le développement Frontend de manière générale. Enfin, j'ai pu utiliser et perfectionner mes connaissances en base de données et en apprendre plus sur le fonctionnement du logiciel sur lequel je travaille au quotidien.

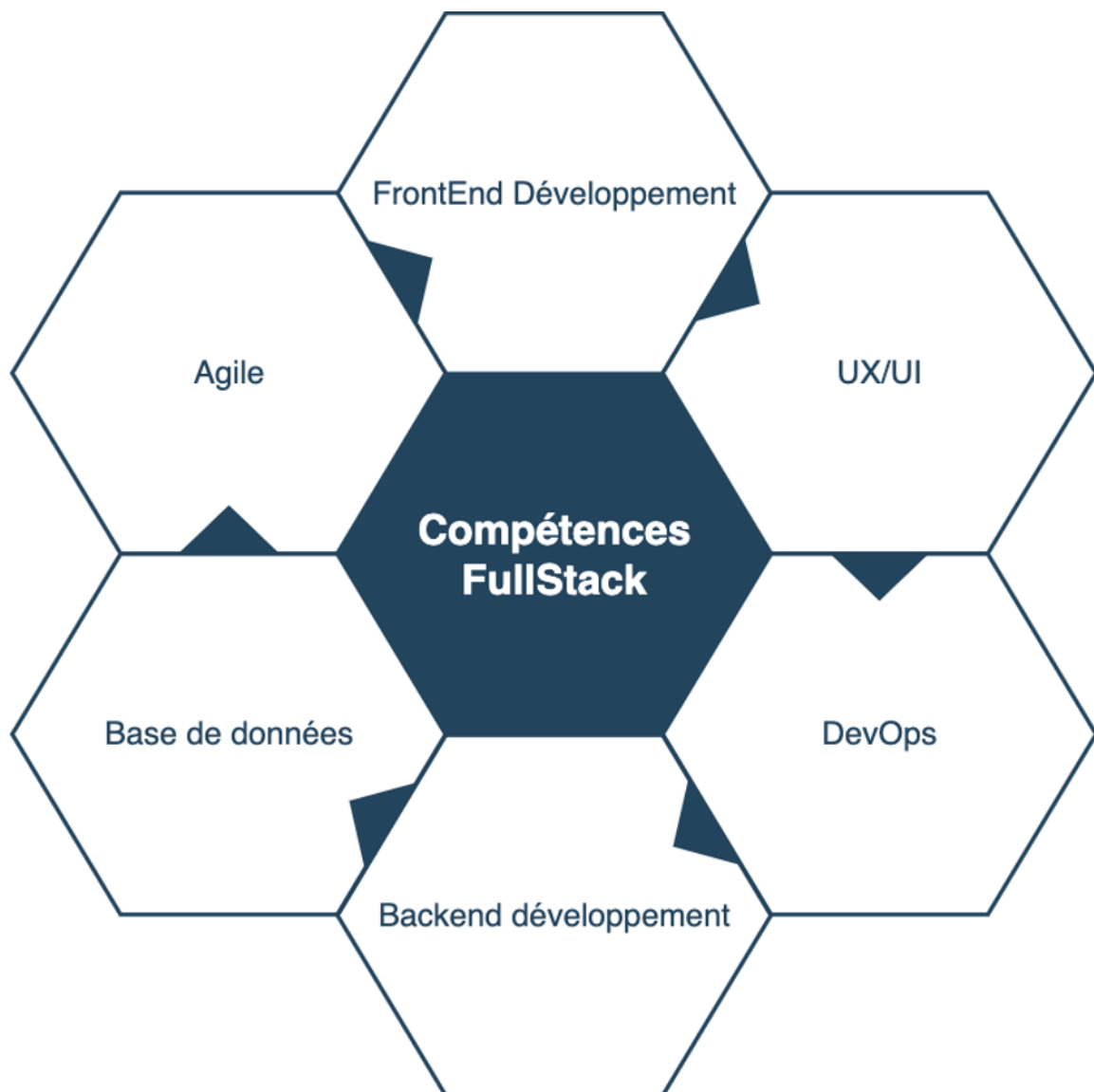


FIGURE 4.1 – Graphique de synthèse présentant les différentes compétences techniques acquises

4.2 Les compétences générales

Au cours de ce projet, j'ai pu acquérir bien plus que des compétences techniques et informatiques.

En effet, j'ai pu développer un grand nombre de compétences dites Soft Skills, indispensables pour tout futur ingénieur.

Le fait de parcourir de nombreuses pages de documentation a non seulement amélioré mes capacités à rechercher une solution technique, mais cela m'a également permis de développer ma capacité à synthétiser et trier les informations à disposition.

Cette compétence informationnelle est pour moi essentielle lorsqu'on souhaite exercer le métier d'ingénieur puisqu'il s'agit d'une profession au sein de laquelle il est fréquent d'être confronté à l'inconnu.

En raison de l'alternance entre présentiel et télétravail, j'ai dû apprendre à m'adapter, à communiquer autrement avec mes collègues lorsqu'ils étaient en distanciel et je pense que cette adaptabilité va constituer une véritable force dans ma future carrière professionnelle.

Dans ce contexte particulier de crise sanitaire, j'ai également pu mieux mesurer les enjeux et besoins du domaine de la santé.

En effet, si la crise du COVID-19 a pu impacter de manière négative de nombreuses entreprises, Clarisys Informatique, en raison de son statut d'éditeur de logiciel pour les laboratoires d'analyses médicales, a vu sa quantité de travail significativement augmenter.

Plus que jamais, j'ai dû faire en sorte que mon travail soit conforme aux exigences et aux normes d'efficacité et de sécurité.

A l'occasion du travail en distanciel, j'ai également pu acquérir une certaine autonomie au cours de ce projet.

Même si mon maître d'apprentissage s'est rendu très disponible, j'ai également eu l'occasion de souvent travailler seul ce qui m'a permis d'apprendre à mieux m'organiser.

De plus, cela m'a aussi permis de mieux me connaître et de mettre en évidence mes qualités et mes défauts.

Enfin, j'ai bien sûr pu améliorer mes capacités relationnelles, en ayant des échanges réguliers avec mon maître d'apprentissage ainsi qu'avec l'ensemble de mes collègues. J'ai pu m'affirmer en présentant des réunions telles que le Daily Meeting ou encore les réunions de présentation du projet ayant lieu lors de l'amélioration continue.

Lors de cette deuxième année d'alternance et la première au sein de Clarisys Informatique, j'ai donc pu améliorer mes compétences techniques mais également d'autres aspects clés plus formels et rationnels.

Chapitre 5

Conclusion

Cette deuxième année d'apprentissage au sein de la société Clarisys Informatique m'a permis de développer des compétences techniques, particulièrement sur les langages Python et JavaScript.

Grâce aux différents développements que j'ai pu effectuer dans le cadre de l'évolution du middleware MCA, j'ai pu acquérir une certaine expérience dans le développement web FullStack.

De ce fait, la plupart des tâches qui me sont confiées aujourd'hui sont dans la continuité de mon projet.

J'ai donc effectué de nombreuses modernisations de code et des nouvelles interfaces pour une meilleure expérience utilisateur, notamment via ce nouveau projet qui est davantage orienté vers l'UX.

De plus, j'ai pu acquérir à la fois de solides connaissances sur le monde de la biologie médicale et les enjeux généraux du système de santé français mais aussi les compétences nécessaires à un ingénieur, telles que la capacité d'analyse, la mobilisation de ressources techniques, l'utilisation d'outils de modélisation...

Tout au long de ce projet, j'ai pu avoir un aperçu du métier de développeur FullStack, des enjeux de sécurité, de l'interopérabilité ainsi que le point de vue législatif de la biologie médicale.

D'autre part, en ce qui concerne mon futur au sein de Clarisys Informatique, j'aspire à me spécialiser sur le Framework Vuejs. L'équipe technique l'a élu en tant que le Framework Frontend à adopter pour la nouvelle interface utilisateur.

Pour ce qui est de mon projet professionnel à plus long terme, je souhaiterais idéalement rester dans le domaine de la santé et être au plus proche du métier, ce qui est le cas à Clarisys Informatique, tout en travaillant en tant que développeur FullStack.

J'aimerais également être amené à travailler avec des outils modernes et en perpétuelle évolution tels que Django et VueJs.

Bibliographie

- [1] Guide de bonne execution des analyses des biologie medicale. *JOURNAL OFFICIEL DU 11 DECEMBRE 1999*, 1999.
- [2] Anfor. Maintenance - concepts et définitions des activités de maintenance. 1994.
- [3] Roy T. Fielding. Rest apis must be hypertext-driven,<https://roy.gbiv.com/untangled/2008/rest-apis-must-be-hypertext-driven>. 2008.