

RAPPORT DE STAGE

Année universitaire 2019/ 2020

Nom et Prénom de l'étudiant :

PINTORE Angeline

Année d'études :

- L3 INFO L3 INFO APPR. L3 MATHS
 M1 MIAGE M1 INFO. DECISION M1 INFO APPR. M1 MATHS

Organisme d'accueil : Laboratoire Lorrain de Recherche en Informatique et ses Applications (LORIA)

Titre du rapport : Rapport du développement d'une application d'analyse d'entretiens entre des professionnels de santé et leurs patients, réalisé par Angeline PINTORE

Tuteur de stage : AMBLARD Maxime

Dates du stage : 27/05/2020 au 28/07/2020 (8 semaines)

Remerciements

Je tenais premièrement à remercier Madame Maude Manouvrier, enseignante référente, pour le temps dévoué, autant dans la recherche, dans les procédures ou consignes à respecter du stage.

Je remercie aussi vivement l'organisation du LORIA de m'avoir accueillie au sein de leurs équipes pendant 8 semaines et notamment l'équipe Sémagramme pour leur bienveillance.

Plus particulièrement, je remercie Monsieur Maxime Amblard, mon tuteur professionnel et membre du projet, pour sa confiance, sa disponibilité et sa compréhension. Ce fut une expérience très enrichissante, autant du point de vue personnel que professionnel.

Enfin, j'exprime toute ma reconnaissance à Monsieur Pierre Lefebvre, ingénieur de l'équipe, pour m'avoir patiemment guidée, tous les jours, durant ces semaines. Je ressors de cette expérience avec plus de connaissances et surtout, une meilleure façon d'appréhender le travail dans le domaine informatique.

Sommaire

1	Introduction	2
2	Organisation de la recherche au sein du LORIA	3
2.1	Présentation du LORIA	3
2.1.1	Activités	3
2.1.2	Caractéristiques humaines et gestion du personnel	5
2.1.3	Organisation et structure	7
2.1.4	Histoire et culture du LORIA	11
2.1.5	Présentation du service : Sémagramme	11
2.2	Analyse de l'association entre le LORIA, INRIA et personnels de santé au sein du projet ODiM	13
2.3	Analyse de la communication	15
2.3.1	Description des outils de communication	15
2.3.2	Positionnement par rapport aux propos d'Elisabeth Charpentier	16
2.3.3	Maux de la communication interne	16
2.4	Conclusion : Une organisation innovatrice dans son travail et sa communication	17
3	Développement d'un logiciel permettant l'analyse d'entretiens	18
3.1	Présentation générale des missions réalisées	18
3.2	Explication des missions	22
3.2.1	Affichage du numéro de version	23
3.2.2	Complétion du formulaire du patient	24
3.2.3	Affichage des informations du formulaire	26
3.2.4	Correction de code	28
3.2.5	Analyse d'entretiens	29
3.2.6	Adaptation pour plus de 2 personnes	31
3.2.7	Tri des entretiens	32
3.3	Procédure d'intégration du code avec Git	32
3.4	Conclusion : Le développement d'une application à travers divers outils	33
4	Conclusion générale : Le développement d'un logiciel au sein d'une organisation innovatrice	34
5	Webographie	36

1 Introduction

Étant en troisième année de licence informatique, j'ai dû réaliser un stage dans ce domaine afin de travailler concrètement sur des notions étudiées au cours de cette année scolaire et en découvrir de nouvelles. En effet, du 27 mai au 28 juillet 2020, j'ai eu la chance de travailler pour le Laboratoire Lorrain de Recherche en Informatique et ses Applications (ci-après LORIA), situé à Villers-lès-Nancy dans la région Grand-Est. Au vu des circonstances particulières du COVID-19, mon stage s'est entièrement déroulé en télétravail.

Le LORIA est une organisation assez importante, puisqu'elle regroupe environ 400 personnes, séparées en plusieurs équipes, elles-mêmes structurées en divers départements. Je l'ai donc choisi pour réaliser ce stage, du fait de sa renommée dans le monde de l'informatique en Lorraine ; mais surtout, car la mission proposée éveillait particulièrement mon intérêt.

Ma mission consistait au développement d'une application d'analyse d'entretiens entre des professionnels de santé et leurs patients. Lors de mon arrivée, une première version existait. Cette application permettait notamment de recenser des informations concernant le patient, d'ajouter des entretiens enregistrés et d'effectuer des tests neurocognitifs. J'ai donc dû y ajouter des fonctionnalités, voire améliorer certaines existantes. À l'heure actuelle, celle-ci est encore en développement mais devrait constituer un réel support de travail pour les psychologues, du fait de sa capacité d'analyse.

Au sein de l'équipe Sémagramme, en charge de ce projet, j'ai été encadrée par Monsieur Maxime Amblard, avec qui je m'entretenais hebdomadairement et par Monsieur Pierre Lefebvre, de manière quotidienne. Ce dernier me transmettait des missions et m'aiguillait en cas de problème.

Afin que vous compreniez mieux le contexte de ce stage ainsi que ma place dans le développement du logiciel, je présenterai dans un premier temps l'organisation du LORIA ; puis, dans un second temps, le projet informatique réalisé pour vous faire part de mon travail.

2 Organisation de la recherche au sein du LORIA

2.1 Présentation du LORIA

Le LORIA est un laboratoire de recherche spécialisé en sciences informatiques, créé en 1997. Il s'agit d'une Unité Mixte de Recherche, c'est-à-dire qu'il repose sur la collaboration de plusieurs établissements, à savoir le Centre National de la Recherche Scientifique (ci-après CNRS), l'Université de Lorraine et l'Institut National de Recherche en Informatique et en Automatique (ci-après INRIA).

2.1.1 Activités

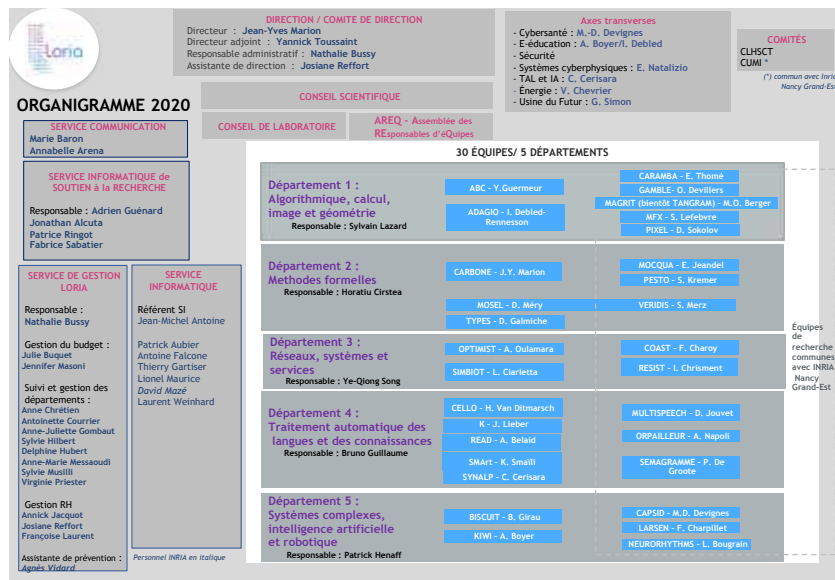


FIGURE 1 – Organigramme du LORIA ¹

L'Université de Lorraine comporte deux missions essentielles. La première concerne la formation et donc l'éducation dans le supérieur. La seconde concerne, quant à elle, la recherche. On retrouve 10 pôles concernant les différents sujets de recherche, à savoir :

- A2F : Agronomie, Agroalimentaire et Forêt ;
- **AM2I** : Automatique, Mathématiques, Informatique et leurs Innovations, dont fait partie le LORIA ;
- BMS : Biologie, Médecine, Santé ;
- CLCS : Connaissance, Langage, Communication, Sociétés ;

1. Source : loria.fr

- CPM : Chimie et Physique Moléculaire ;
- EMPP : Énergie, Mécanique, Procédés, Produits ;
- M4 : Matières, Matériaux, Métallurgie, Mécanique ;
- OTELo : Observatoire Terre et Environnement de Lorraine ;
- SJPEG : Sciences Juridiques, Politiques, Économiques et de Gestion ;
- TELL : Temps, Espace, Lettres, Langues.

Ces pôles sont gérés par le conseil scientifique de l'Université de Lorraine.

Le LORIA fait donc partie du pôle AM2I, composé de 6 autres laboratoires de recherche comme, par exemple, le Centre de Recherche en Automatique de Nancy (abr. CRAN) et l'Institut Elie Cartan de Lorraine (abr. IECL), qui forment, à eux 3, la Fédération Charles Hermite, spécialisée en mathématiques et STIC (sciences et technologies de l'information et de la communication). La Fédération Charles Hermite est la fédération de recherche la plus importante de Lorraine, comptant environ 710 personnes.

L'activité du LORIA repose sur l'informatique, mais se divise précisément en 6 thématiques :

- Intelligence artificielle
- Santé numérique
- Robotique
- E-éducation
- Cybersécurité
- Fabrication additive

De plus, le laboratoire est séparé en cinq départements, regroupant ces différentes thématiques :

1. Algorithmes, calcul, géométrie et images
2. Méthodes formelles
3. Systèmes, réseaux, services
4. Traitement automatique de la langue et des connaissances
5. Systèmes complexes, intelligence artificielle et robotique

Au sein du LORIA demeure une certaine hiérarchie. En effet, en haut de celle-ci se place l'équipe de direction, constituée du directeur, Monsieur Jean-Yves Marion, du directeur-adjoint, Monsieur Yannick Toussaint, de la responsable administratif, Madame Nathalie Bussy, et de l'assistante de direction, Madame Josiane Reffort. Cette équipe est chargée de la direction du conseil scientifique du laboratoire. Ce dernier est composé du directeur, d'un directeur adjoint et des responsables des 5 départements. Il a pour mission d'assister les décisions de la direction et permet une communication ascendante, c'est-à-dire qu'il permet aux personnels d'échanger avec la direction. Le conseil scientifique se réunit toutes les deux semaines.

Il existe aussi un conseil de laboratoire. Ce dernier est composé de membres élus pour 4 années et de membres nommés. Ce conseil a pour but de représenter et

donc de prendre des décisions pour les membres du laboratoire. Les décisions peuvent concerner, par exemple, l'organisation du laboratoire, le budget et les formations. Le conseil de laboratoire se réunit environ une fois tous les deux mois.

La recherche, au sein du LORIA, s'effectue par équipes. Chaque équipe comporte un responsable, repose sur une thématique et appartient donc à un département. En regroupant tous les départements, il y a au total 30 équipes dont 16 sont communes avec INRIA. Cependant, ces 5 thématiques, propres aux départements, occultent d'autres thèmes forts, en corrélation avec les sujets de recherche. À ce propos, il existe des axes transverses, regroupant des sujets traités dans différents départements. À la différence des départements, les axes transverses ne font pas l'objet d'équipes. En effet, ils ne sont pas organisationnels mais permettent uniquement de mettre en avant d'autres thématiques.

Tous les responsables d'équipes constituent, quant à eux, l'Assemblée des Responsables des Équipes, nommée AREQ.

Au total, plus de 430 personnes travaillent dans ce laboratoire, ce qui fait de lui le deuxième plus grand laboratoire de Lorraine.

Le LORIA en chiffres (2019) ²

- 29 équipes de recherche (30 désormais)
- 210 chercheurs et enseignants chercheurs
- 120 doctorants
- 14 start-ups
- 600 publications dans les revues
- 60 nationalités représentées
- 13 millions d'euros pour les contrats
- 2 membres IUF (Institut Universitaire de France)
- 9 lauréats ERC (European Research Council)

2.1.2 Caractéristiques humaines et gestion du personnel

Effectifs globaux et leurs évolutions Selon l'évaluation du HCERES en 2016 : "*Le LORIA compte 168 chercheurs et enseignants-chercheurs permanents, 113 doctorants et chercheurs contractuels, 28 personnels permanents en support (dont 9 administratifs, 13 informaticiens et 1 en communication), et mélange 48 nationalités.*"

Toujours selon l'évaluation du HCERES, il semblerait qu'il y ait eu plusieurs évolutions entre 2016 et 2018. En effet, le nombre d'enseignants-chercheurs titulaires et assimilés serait passé de 105 à 119. Le nombre de chercheurs des EPST (Établissement Public à caractère Scientifique et Technologique) ou EPIC (Établissement Public à caractère Industriel et Commercial) titulaires et assimilés serait passé de 63 à 67. Enfin, le nombre d'autres personnels titulaires (appui à la recherche et/ou n'ayant pas d'obligation de recherche) aurait augmenté de 1, passant de 28 à 29.

2. Source : loria.fr

Par ailleurs, on peut lire dans l'évaluation du HCERES de 2016 que le LORIA possédait, à ce moment, seulement 28 équipes alors qu'il en compte désormais 30.

Ces diverses informations montrent que les effectifs du LORIA sont en constante évolution.

Répartition par catégories

Services Le LORIA est composé de 3 services :

- Gestion
- Communication
- Moyens informatiques (en commun avec INRIA)

Le service de gestion est propre au LORIA et est en collaboration avec le service correspondant d'INRIA. Ce service est lui-même séparé en 3 pôles : pôle de gestion, pôle budgétaire et pôle de ressources humaines. Selon la même étude du HCERES que précédemment, cette configuration semble optimale car considérant autant les intérêts du laboratoire, des départements, des équipes et d'INRIA.

Le service de communication a été créé en 2011 et comporte un responsable, son unique personnel, depuis 2015. De même, il semblerait que ce service soit une réussite. En effet, la communication est faite de manière à, non seulement, s'adresser à des chercheurs du laboratoire, mais aussi, au grand public. Ce service permet la considération du LORIA en tant que tel, indépendamment de INRIA. Le service informatique est le seul service commun à INRIA. Son personnel varié, provenant de l'Université de Lorraine, du CNRS ou du LORIA, permet une efficacité indéniable.

Personnels Au sein du LORIA, il existe deux types de personnels.

Les chiffres et données suivants reposent sur l'évaluation du HCERES faite en 2016. À ma connaissance, aucun autre document plus récent ne recense autant d'informations.

Le premier type de personnel est appelé ITA/BIATSS. Il s'agit des personnels ingénieurs, administratifs, techniques, sociaux et de santé et des bibliothèques. En 2016, il existait 23 emplois permanents de ce domaine, dont 6 concernant l'Université de Lorraine, 12 le CNRS et 5 INRIA.

Le deuxième type de personnel porte sur les chercheurs et enseignants-chercheurs permanents. En 2016, le LORIA comptait 105 enseignants-chercheurs et 63 chercheurs de statut CNRS ou INRIA ; respectivement 119 et 67, en 2018. Ces deux postes dépendent de la recherche. Cependant, les enseignants-chercheurs permettent davantage la formation de nouveaux doctorants, alors que, les chercheurs s'investissent d'autant plus dans cette branche.

Turn-Over Le Turn-Over est très différent selon les membres du LORIA. En effet, le laboratoire est composé de membres permanents, notamment les enseignants-chercheurs, et de membres non-permanents tels que les stagiaires, les doctorants ou les ingénieurs. Concernant les membres permanents, le Turn-Over est presque nul, l'équipe est très sédentaire et les départs sont très rares. Au contraire, les membres non-permanents ne sont présents que pour une courte durée : durée d'un stage, d'une thèse, voire d'un projet. De fait, le Turn-Over des membres non-permanents est bien plus élevé. Le LORIA recherche le renouvellement et recrute son personnel spécifiquement afin d'avoir "*une politique équilibrée de promotion/renouvellement*", comme le souligne l'évaluation du HCERES.

Évolution professionnelle Il n'est pas évident d'évoluer au sein du LORIA. La première façon d'évoluer est de passer de membre non-permanent à membre permanent. Pour cela, il faut obtenir une thèse puis trouver une place adéquate, au sein du laboratoire. Puis, les membres permanents sont, au début, chargés de recherche et maîtres de conférences. Ils peuvent alors devenir professeurs d'université et être habilités à diriger des recherches, en passant un concours assez difficile. Par ailleurs, toutes les places de dirigeants font l'objet d'élections. L'évolution professionnelle n'est donc pas systématique.

2.1.3 Organisation et structure

Les configurations de Mintzberg Afin d'identifier au mieux l'organisation et la structure du LORIA, nous allons utiliser les configurations de Mintzberg. Celles-ci reposent sur 4 éléments, faisant différer les structures.

Mécanisme de coordination Il définit la stratégie utilisée pour coordonner les effectifs afin d'arriver à l'accomplissement d'une activité. Dans les configurations de Mintzberg, on retrouve 3 mécanismes :

- L'*ajustement mutuel* repose sur la communication informelle, c'est-à-dire une communication non-officielle, voire décontractée.
- La *supervision directe* repose, a contrario de l'ajustement mutuel, sur une communication bien plus formelle. Dans le cas d'une supervision directe, il existe une hiérarchie.
- La *standardisation* se décline notamment en 4 formes :
 - Standardisation des procédés de travail : uniformisation des méthodes de travail et du comportement des salariés
 - Standardisation des résultats : spécification des résultats attendus
 - Standardisation des qualifications et du savoir : détermination de la formation et des qualités nécessaires pour effectuer un travail
 - Standardisation des normes : alignement des règles s'appliquant à l'organisation

Dans le cas du LORIA, on remarque qu'il n'existe pas forcément de standardisation, quelle qu'elle soit. En effet, au sein de chaque département, puis au sein

de chaque équipe, les méthodes de travail, les résultats, les qualifications ainsi que les normes ne sont pas forcément identiques. Bien qu'elles pourraient l'être, il ne s'agit pas d'une volonté puisque chaque équipe est indépendante. Et, par conséquent, chaque département l'est aussi. Le mécanisme de coordination utilisé pourrait s'apparenter à la supervision directe. En effet, comme nous avons pu le voir précédemment, le LORIA est un établissement très structuré où il existe et demeure une hiérarchie. Par exemple, chaque département contient un dirigeant, chaque département contient des équipes et chaque équipe contient un chef d'équipe. Cependant, ma présence au sein de l'équipe Sémagramme m'a permis de réaliser qu'une ambiance assez amicale et accueillante régnait lors des réunions. De plus, tous les membres pouvaient donner leur avis et les décisions ne sont pas uniquement prises par le chef d'équipe. Enfin, au vu des conditions spécifiques au COVID-19, à savoir le télétravail, divers moyens de communication informels étaient utilisés, comme par exemple le logiciel Discord, dont nous reparlerons lors de l'analyse de la communication. Par conséquent, le mécanisme de coordination semblerait être l'*ajustement mutuel*.

Composantes d'une organisation Mintzberg en a identifié 5, propres à l'organisation d'une structure :

- Le *sommet stratégique* comporte les dirigeants prenant les décisions stratégiques.
- La *ligne hiérarchique* permet d'établir le lien entre le sommet stratégique et le centre opérationnel. Elle est principalement constituée de managers.
- La *technostructure* est composée de spécialistes. Celle-ci est située en dehors de la ligne hiérarchique.
- Le *support logistique* fournit au reste de l'organisation divers services internes.
- Le *centre opérationnel* rassemble toutes les personnes effectuant le travail de production de biens ou de services.

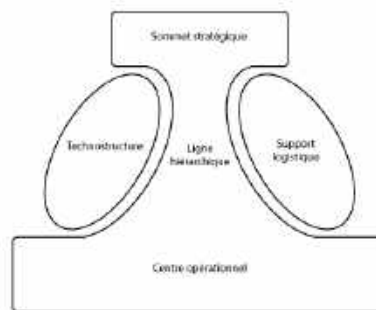


FIGURE 2 – Les composantes d'une organisation³

3. Source : espacesoignant.com

Dans l'analyse de Mintzberg, le but est de savoir quelle composante joue un rôle clé dans l'organisation. Dans notre cas, le support logistique a un rôle prépondérant. En effet, c'est lui qui permettra de peaufiner voire finaliser les projets. Il est indispensable à la bonne production de services.

Facteurs de contingence Ils constituent des variables qui sont propres à chaque organisation et qu'il convient donc d'analyser indépendamment pour comprendre la structure du LORIA.

— *Environnement* : Il semble être stable. En effet, comme déclaré sur le site du LORIA : "*Tout au long de ces années, l'environnement du laboratoire a contribué à maintenir dans l'unité des objectifs d'excellence à la fois théoriques et appliqués, et à développer harmonieusement les recherches pluridisciplinaires et les applications*", l'emploi du terme "maintenir" souligne la notion de stabilité du point de vue des objectifs et des missions. De plus, cela fait maintenant plusieurs années que la même structure existe, décrite notamment par la présence des 5 départements. Enfin, cette notion de stabilité est renforcée par le caractère très sédentaire des membres permanents. Cependant, au vu du Turn-Over élevé des membres non-permanents, l'environnement peut paraître bien moins stable. En effet, les effectifs et donc la création de nouvelles équipes sont en constante évolution, démontrant d'ailleurs un certain dynamisme et une motivation des équipes. Par ailleurs, bien que l'organisation du LORIA soit clairement définie, elle peut paraître compliquée, au vu du nombre de services et du manque de lien direct entre eux par moment.

Son environnement se veut alors difficile à analyser car il est autant simple que compliqué et autant stable que changeant, voire évoluant.

- *Taille* : Il s'agit plutôt d'un grand laboratoire en Lorraine, puisqu'il constitue le deuxième plus grand laboratoire de la région. En effet, il comporte environ 400 membres.
- *Âge* : Le laboratoire est relativement récent puisque créé en 1997.
- *Système technique* : Il s'agit d'un système diversifié selon les différentes attentes, mais essentiellement automatisé. On peut parler d'un système sophistiqué et à la pointe. En effet, ce laboratoire a pour but de proposer des innovations en informatique et se doit donc d'être actuel dans ses manières d'opérer. Par exemple, au cours de l'année 2020, le laboratoire a créé un robot permettant de suivre et de filmer le vol des insectes. Ce projet a nécessité l'utilisation de divers câbles, caméras et d'un contrôleur. En outre, tous les projets informatiques concernant des logiciels dépendent d'outils numériques.
- *Stratégie* : Bien que le sommet stratégique ait une place importante, celle-ci peut provenir de plusieurs sources, les dirigeants prennent en compte les différents avis.
- *Pouvoir de décision* : Il semble d'avantage appartenir aux experts.

Paramètres de conception L'analyse de ceux-ci repose sur plusieurs points.

- Conception des postes de travail
 - *Spécialisation du travail* : Le but est de savoir si, au sein du LORIA, il s'agit d'une spécialisation verticale ou horizontale. La spécialisation verticale, comme son nom l'indique, témoigne d'une notion de hauteur. Ainsi, elle évoque la division du travail reposant sur la hiérarchie. À l'inverse, la spécialisation horizontale dénote la notion de largeur et donc concerne la division du travail selon les différents domaines d'activité. Dans ce laboratoire, la division du travail se fait plutôt de façon horizontale. En effet, lors d'un projet, l'ensemble d'une équipe se met au travail, peu importe leur niveau dans la hiérarchie. Ici, les dirigeants ne donnent pas seulement des ordres mais participent activement au bon déroulement de celui-ci.
 - *Formation et socialisation/endoctrinement* : Les formations ne semblent pas être un frein, au sein du LORIA. En effet, beaucoup de formations sont faites pour permettre l'avancée de projets.
 - *Formalisation des comportements* : Elle se caractérise par une standardisation des méthodes de travail à l'aide de règles, instructions voire fiches de travail. Celle-ci n'existe pas vraiment au sein du LORIA. En effet, le résultat est très largement favorisé à la manière dont le personnel y arrive. On relève plutôt des informalités.
 - *Taille des unités* : Bien que le LORIA comporte beaucoup de personnels, ceux-ci sont séparés selon leur activité. Les chercheurs, par exemple, sont séparés en 30 équipes, d'une dizaine de personnes. Les unités sont donc plutôt petites.
- Conception des liens latéraux
 - *Système de contrôle des performances et planification des actions* : De mon point de vue, le système de contrôle semble exister, mais à moindre mesure. Les actions à réaliser sont planifiées et organisées; cependant, l'accent est davantage mis sur la bonne réalisation et non le temps pris. Certaines missions peuvent s'avérer urgentes mais, dans l'ensemble, la planification semble limitée.
 - *Mécanismes de liaison* : Il semble y avoir beaucoup de mécanismes de liaison, notamment dans la communication entre les différents membres.
- Conception du système de prise de décision
 - *Centralisation/Décentralisation* : Il s'agit d'analyser de quelle manière est prise la décision. D'une part, il se peut qu'elle le soit uniquement par le sommet stratégique. Dans ce cas, on parlera de centralisation. D'autre part, le pouvoir de décision peut être réparti. On parlera alors de décentralisation. De manière similaire à la spécialisation du travail, les directives ne sont pas seulement données par le sommet stratégique. La prise de décision peut donc être décentralisée.

Suite à l'examen de ces différents critères, on peut en conclure que le LORIA constitue une **organisation innovatrice**. En effet, malgré sa structure très précise et la hiérarchie qu'il y demeure, il contient davantage de critères correspon-

dant à une organisation innovatrice. On relèvera notamment la décentralisation de la prise de décision, la division du travail horizontale, le manque de formalisation au profit d'informalités, le système technique très sophistiqué, l'ajustement mutuel demeurant au sein des équipes et surtout la volonté d'innover.

2.1.4 Histoire et culture du LORIA

L'informatique s'est démocratisée en Lorraine en 1958 notamment grâce au professeur Jean Legras, de part l'analyse numérique et les mathématiques appliquées.

Au début des années 1970, suite à la création de 3 nouvelles universités à Nancy, des enseignants-chercheurs sont embauchés. À côté de l'enseignement, ils effectueront de la recherche au sein du Centre de Recherche en Informatique de Nancy (abr. CRIN) qui est le laboratoire affilié à ces 3 universités. Quelques années plus tard, le CRIN sera alors reconnu par le CNRS comme laboratoire associé.

En 1986, l'INRIA Lorraine et l'Institut de l'Information Scientifique et Technique (abr. INIST) sont créés à l'aide du CNRS, afin de participer au "*renouveau économique de la région*", comme explicité sur le site officiel du LORIA.

Suite à la création de ces différents laboratoires et notamment l'évolution du CRIN, c'est en 1997 que le LORIA va être conçu. De ceci naîtra en réalité une Unité Mixte de Recherche, reposant sur la collaboration du CNRS, d'INRIA et de l'Université de Lorraine, dont fait partie le LORIA. Les objectifs et réalisations de ce dernier sont clairement énoncés sur le site du LORIA : "[...] *maintenir dans l'unité des objectifs d'excellence à la fois théoriques et appliqués, et développer harmonieusement les recherches pluridisciplinaires et les applications.*"

Suite à sa conception, des nouveaux sujets de recherche vont être étudiés, comme par exemple la bioinformatique, les grilles et calculs distribués, la sécurité informatique ou encore la théorie des virus. De plus, le LORIA va sans cesse essayer de s'adapter à l'ère dans laquelle il demeure. En effet, la période de sa création correspond à une période où les actions médicales avaient une grande importance. À ce propos, les chercheurs du laboratoire s'intéresseront à des questions d'infoéthique.

En conclusion, Nancy, de part notamment la conception du LORIA, influence très fortement la recherche en informatique, depuis maintenant plusieurs années.

2.1.5 Présentation du service : Sémagramme

Comme dit précédemment, le LORIA se décompose en équipes de recherche. Lors de mon stage, j'ai intégré l'une d'entre elle : Sémagramme.

Sémagramme appartient au quatrième département qui étudie le traitement automatique des langues et des connaissances. Il s'agit d'une équipe commune

avec INRIA, dirigée par Monsieur Philippe De Groot. Elle travaille principalement sur l'analyse d'énoncés et de discours en langue naturelle. Pour cela, elle peut être amenée à réaliser des modélisations logiques. Plus particulièrement, les projets de Sémagramme s'organisent autour de 3 axes :

- *Les interfaces syntaxiques-sémantiques* : Le projet Sémagramme repose essentiellement sur l'analyse de la sémantique. La sémantique correspond à l'analyse des signifiés dans un discours, c'est-à-dire comprendre ce qui a vraiment voulu être transmis par les interlocuteurs. Néanmoins, l'analyse sémantique est presque indissociable de l'analyse syntaxique, puisque la syntaxe représente un outil très important pour celle-ci. C'est pourquoi, les chercheurs travaillent sur un modèle mêlant les deux.
- *Les dynamiques de discours* : Bien que l'analyse d'un discours prend en compte le contexte de celui-ci, les dynamiques doivent être considérées. En effet, une intonation peut changer le sens d'une phrase.
- *Les ressources lexicales et grammaticales de bases communes* : Ce dernier thème fait référence à la création d'arbres syntaxiques permettant l'analyse. Ceux-ci sont induits des différentes grammaires, de lexiques et d'algorithmes d'analyse.

Selon l'étude du HCERES, le quatrième département dont fait partie Sémagramme comportait, le 1er janvier 2018, 33 enseignants-chercheurs titulaires et assimilés, 18 chercheurs des établissements publics à caractère scientifique et technologique (abr. EPST) ou des établissements publics à caractère administratif (abr. EPIC) titulaires et assimilés et 2 autres personnes titulaires, travaillant dans la recherche ou non. Toujours selon cette étude, du 1er janvier 2011 au 30 juin 2016, 35 thèses ont été soutenues, il y a eu 41 post-doctorants ayant passé au moins 12 mois dans l'unité et 4 habilitations à diriger les recherches ont été soutenues. Sémagramme est, à ce jour, composée de 4 membres permanents, présents depuis plusieurs années. L'équipe est complétée par 2 autres membres permanents, rattachés secondairement à l'équipe, 1 membre permanent en délégation et une dizaine de membres non-permanents, tels que des doctorants, des ingénieurs et des stagiaires.

Pour ce stage, j'ai participé au projet ODIM. Il s'agit d'un projet interdisciplinaire, à l'interface de la psychiatrie-psychopathologie, de la linguistique, de la sémantique formelle et des sciences du numérique. Il tend à remplacer le paradigme des Troubles du Langage et de la Pensée (abr. TLP) tel qu'on l'utilise dans le secteur de la santé mentale par un modèle sémantico-formel des Troubles du Discours (abr. TDD). Le but est de traduire ces troubles en signes diagnostiques ainsi que de dépister des personnes vulnérables dites "à risques". Le projet ODIM réunit activement 5 personnes, propres à la recherche au sein du LORIA : membres permanents, ingénieurs et doctorants. Des stagiaires, comme moi, viennent ponctuellement aider à l'avancement du projet. Cependant, le développement du logiciel ne concerne qu'un ingénieur informatique, avec qui j'ai donc travaillé durant ce stage.

2.2 Analyse de l'association entre le LORIA, INRIA et personnels de santé au sein du projet ODiM

Certaines équipes sont propres au LORIA, d'autres résultent de l'association du LORIA et de INRIA, c'est le cas de Sémagramme. Dans le cadre du projet ODiM, ces deux entités doivent, en plus, collaborer avec du personnel de santé comme par exemple des psychologues. L'association de personnels provenant d'organisations différentes peut parfois poser problème. En effet, il se peut qu'elles n'aient pas les mêmes objectifs voire les mêmes stratégies. Parfois, c'est plutôt les façons de travailler, les normes et donc, plus globalement, la culture qui sera source de conflits. Dans notre cas, INRIA et le LORIA sont deux laboratoires de recherche de Lorraine, qui souhaitent apporter leur aide aux psychologues. Par conséquent, leurs intérêts semblent être similaires. Cependant, la présence de cultures différentes peut potentiellement révéler des failles, d'où notre analyse.

La culture est une notion assez complexe à définir. On peut la décrire comme étant le regroupement de divers thématiques comme la connaissance, l'art, les croyances, la morale, les coutumes et toutes les habilités acquises par l'Homme, en demeurant au sein de la société. On pourrait aussi la qualifier comme étant un ensemble de règles, d'interdits ou même de normes, se transmettant de générations en générations. Il existe cependant un postulat : l'entreprise est un lieu autonome de création culturelle. Au sein de chaque entreprise demeure de la culture. On différenciera deux types spécifiques : la culture d'entreprise et la culture dans l'entreprise. Au vu des cultures propres à INRIA et au LORIA, le but est de savoir si celles-ci entrent en conflit au sein de l'équipe.

La culture d'entreprise Il s'agit d'un dispositif de gestion. Elle constitue l'identité de l'entreprise et repose sur l'ensemble des normes, des règles, des interdits et des coutumes propres à celle-ci. Le but de la culture d'entreprise est de faire réussir l'entreprise, lui faire atteindre ses objectifs. Bien souvent, les responsables souhaitent que cet effort provienne de tous les salariés. C'est pourquoi, cet ensemble de normes aura pour but de fédérer les effectifs, afin que ces derniers travaillent ensemble, de la même manière, dans un but commun. On remarquera que cette culture pourrait presque s'apparenter à de l'instrumentalisation culturelle.

La culture dans l'entreprise Elle correspond à une "sous-culture" de l'entreprise, propre à un groupe de personnes. En effet, chaque groupe invente sa propre culture et son propre cadre de références culturelles pour faire face à des problèmes ou obligations. Cette notion a été définie comme microculture par Michel Liu. De plus, elle dispose de 3 dimensions :

- Symbolique
- Technologique

Bien que la culture dans l'entreprise soit bien moins formelle que la culture d'entreprise, elles peuvent parfois être similaires.

Au vu de mes conditions particulières de travail, à savoir le télétravail, j'ai dû interroger certaines personnes de l'équipe pour comprendre comment se déroulent les conditions de travail. Il semblerait que le personnel soit plutôt séparé selon leur titre. Par exemple, les informaticiens ne sont pas quotidiennement en contact avec les psychologues. En général, certains informaticiens entrent en contact avec eux, une fois par semaine, par le biais d'une réunion. C'est plus rarement que toute l'équipe se réunit entièrement pour faire le bilan. Lors de ce genre de réunion, tous les sujets seront abordés, comme notamment la psychiatrie, l'informatique, les entretiens dans les hôpitaux et le protocole d'expérimentation. Ce sont, de plus, des réunions qui sont plus longues en terme de durée. En effet, tous les participants ne sont pas forcément familiers avec les sujets abordés. C'est alors le responsable, ayant des aptitudes en informatique et en psychologie, qui jouera le rôle de traducteur entre les deux parties.

Plus globalement, au sein de Sémagramme, il y a souvent des réunions entre personnels, travaillant sur le même sujet. Cependant, une fois par semaine, l'équipe se réunit pour discuter du travail de chacun. En effet, ce sont des réunions où tout le monde se présente et explique ce qu'il a fait durant la semaine. À cette réunion, tous les membres de l'équipe sont conviés, que ce soit des doctorants, des enseignants-chercheurs, des stagiaires, des ingénieurs etc. De plus, il semblerait que l'équipe ait prévu, à la rentrée, de réaliser des réunions où chaque membre parlerait du sujet de son choix.

Par ailleurs, l'équipe semble assez libre. Bien qu'elle se réunisse pour les réunions, certains membres ont d'autres activités extérieures, comme par exemple l'enseignement ou des conférences à organiser. Ils ne sont alors pas toujours en contact et n'ont pas vraiment de rituels ou de coutumes entre eux, bien que l'entente semble très bonne lors des réunions. Ceci n'est pour autant pas un prétexte de cassure sociale. En effet, même si les enseignants-chercheurs peuvent s'absenter, leur place reste intacte à leur retour. J'ai pu remarquer que les membres faisaient en sorte que le contact entre eux soit gardé. En effet, outre les réunions, divers moyens de communication ont été mis en œuvre pour pouvoir discuter simplement. Par exemple, un salon informatique se nommant "*Machine à café*" a été créé, permettant de garder l'esprit des regroupements autour de la machine à café et d'avoir des discussions bien moins formelles qu'en réunion.

Enfin, il semblerait que tout le monde ait les mêmes façons de travailler ou du moins, qu'elles ne soient pas conflictuelles. En effet, les réunions permettent des échanges, peuvent faire l'objet de recommandations mais ne sont pas des jugements de valeur sur les façons de travailler de chacun.

Suite à cette analyse, on remarque, dans un premier temps, que l'équipe se réunit souvent. En effet, que l'on soit un stagiaire ou ingénieur, nous avons le droit à la parole et nous présentons notre travail de la semaine. À priori, il

n'existe pas vraiment de distinction entre les différentes personnes. En interrogeant un membre de l'équipe, j'ai pu comprendre que les organisations de provenance des membres n'impactaient pas les relations. L'équipe Sémagramme semble être unie. Les différents membres semblent être en très bonne relation et n'ont aucun mal à travailler entre eux, malgré leur domaine d'étude parfois différent. Il semblerait que les membres de Sémagramme ait construit leur propre culture commune, reposant sur des réunions et des échanges réguliers, des façons de travailler diverses et une bienveillance commune, sans pour autant avoir des coutumes ancrées, comme pourraient en avoir d'autres entreprises. En conclusion, l'association du LORIA, d'INRIA et des psychologues semblent être parfaitement réussie.

2.3 Analyse de la communication

2.3.1 Description des outils de communication

Il existe divers moyens de communiquer au sein des organisations. Ceux-ci vont dépendre du type de communication. D'une part, la communication externe correspond à tous les dispositifs qui vont être mis en place par l'entreprise pour construire sa propre image. D'autre part, la communication interne renvoie aux moyens de communiquer au sein de l'entreprise. Dans les deux cas, il y a 3 types d'outils : ascendants, descendants et transversaux.

Outils	Ascendants	Descendants	Transversaux
<i>Internes</i>	<ul style="list-style-type: none"> - Bulletins d'informations - Messagerie intranet Inria - Réunions - Newsletter 	<ul style="list-style-type: none"> - Enquêtes/sondages - Suggestions - Réunions - Points de situation interne 	<ul style="list-style-type: none"> - Réseaux sociaux (Discord, Slack) - Intranet Inria - Réunions/Conventions - Réunions informelles
<i>Externes</i>	<ul style="list-style-type: none"> - Conférences sur divers sujets de recherche - Présentations de projets accomplis 	<ul style="list-style-type: none"> - Site internet - Page Wikipédia - Partenariats avec des universités du monde entier - Formations dans l'enseignement 	<ul style="list-style-type: none"> - Réseaux sociaux (Facebook, Twitter, LinkedIn) - Communication financière

FIGURE 3 – Les outils de communication

Selon moi, tous les outils mis à disposition sont pertinents. En effet, certains outils sont communs à une majorité d'entreprises, comme par exemple les bulletins d'informations, la newsletter et les réunions ; d'autres sont en adéquation aux laboratoires de recherche, comme notamment les conférences et les partenariats avec diverses universités.

Si je devais choisir l'outil le plus pertinent, je dirais qu'il s'agit de Discord. C'est un logiciel permettant la communication entre divers individus. Initialement, Discord a été créé pour les joueurs et n'est donc, à priori, pas fait pour la

communication professionnelle. Cependant, du fait du confinement, le LORIA a su en faire un usage autant professionnel que ludique. En effet, c'est à travers Discord que je communiquais quotidiennement avec mon encadrant, Pierre Lefebvre. Par ailleurs, un serveur commun à l'équipe Sémagramme existe (*Machine à café*) et permet à tous les membres de communiquer entre eux sur différents sujets, peu importe leur place hiérarchique dans l'équipe. Bien que Discord était uniquement propre aux personnels, il m'a permis de me sentir mieux intégrée au sein de l'organisation.

Au contraire, l'outil le moins pertinent, selon moi, est Slack. En effet, je n'ai pas compris ce qu'il apportait de plus que Discord et cela en devenait presque perturbant. De plus, je ne l'ai jamais vraiment utilisé.

2.3.2 Positionnement par rapport aux propos d'Elisabeth Charpentier

Elisabeth Charpentier a déclaré : "*La communication uniquement descendante, c'est fini. Nous donnons la parole aux équipes, organisons des « chats » réguliers avec la direction générale. Il faut être prêt à tout entendre ! Mais on y trouve notre compte, en termes d'idées et d'innovations*".

Je pense personnellement que le LORIA est au-delà de cette posture. En effet, mon encadrant me demandait souvent mon avis quant au travail que j'avais à faire, il prenait en compte mes idées voire même parfois mes envies, malgré le fait qu'il soit ingénieur et moi stagiaire de fin de licence. De plus, j'ai pu remarquer lors de réunions que, bien que celles-ci soient présidées par le dirigeant de Sémagramme, ce dernier était très à l'écoute des membres. Et, grâce à l'utilisation de Discord, la communication entre différents membres, de hiérarchie différente, se faisait très facilement, même pour une stagiaire comme moi. Je n'ai donc pas eu l'impression que la communication était uniquement descendante.

2.3.3 Maux de la communication interne

Dans son texte, Michel Augendre parle de plusieurs maux altérant la communication interne.

L'absence de dispositifs formalisés Elle peut parfois être source de rumeurs et donc de manque de communication. Le LORIA n'est pas touché par celle-ci. En effet, au sein des équipes, il y a beaucoup de réunions permettant à tous les membres de communiquer. Cependant, il est vrai qu'en dehors de notre équipe, nous ne pouvons pas toujours savoir sur quoi travaillent les autres équipes. Cependant, cela ne semble pas être source de conflits.

La communication formelle prenant le pas sur les relations humaines

Le LORIA favorise les réunions, les interactions sociales ne sont donc en aucun cas remplacées. De plus, ayant assisté à des réunions, j'ai pu remarquer

qu'elles étaient assez détendues et pouvaient, par moment, s'éloigner du sujet initialement prévu. Enfin, j'ai pu remarquer une certaine entente entre différents membres.

La faiblesse ou l'absence de communication ascendante Au sein de mon équipe, la communication ascendante était bel-et-bien présente. Il était parfaitement possible d'avoir des interactions avec des personnes situées plus haut hiérarchiquement, assez facilement.

La présence de maillons faibles De mon point de vue, il n'y a pas l'air d'y avoir de maillons faibles.

Les décalages de vitesse entre les circuits d'information À priori, il n'y a pas de décalages de vitesse entre les circuits d'information.

La discordance des sources Il n'y a pas de discordance des sources, au sein de l'équipe Sémagramme. Je ne sais pas ce qu'il en est pour les autres équipes.

L'inflation ou la pénurie des échanges Par moment, je recevais beaucoup de mails : enquêtes, réglementations, bulletins d'informations, newsletter ou points de situation interne. Bien que ceux-ci me permettaient de me sentir intégrée dans l'équipe, ils pouvaient parfois me faire passer à côté de mails bien plus pertinents et importants et, il était compliqué de pouvoir tous les lire.

2.4 Conclusion : Une organisation innovatrice dans son travail et sa communication

En réalisant ces analyses, j'ai pu remarquer que le LORIA met un point d'honneur à l'innovation, autant dans son organisation que dans ses recherches. En effet, sa façon de travailler et de communiquer entrent totalement dans l'air du temps. Cela se traduit notamment par une communication très facilitée, un ajustement mutuel et une décentralisation de la prise de décision. De plus, il semblerait qu'il s'agisse d'une organisation très équilibrée, puisque je n'ai pas remarqué de problèmes, spécifiquement dans la communication. Même en tant que stagiaire, je me suis sentie intégrée et respectée. Il conviendrait de se demander si les entreprises devraient, désormais, toutes s'organiser de la sorte, c'est-à-dire plus librement, au vu de la réussite du LORIA.

3 Développement d'un logiciel permettant l'analyse d'entretiens

Dans le cadre du projet ODiM, des membres de l'équipe Sémagramme ont développé une application nommée SLODiM. Celle-ci est destinée à des praticiens et permet l'analyse formelle des entretiens entre les professionnels et leurs patients. Partant de cette analyse, elle permet de présenter des indices issus de la linguistique sur différents aspects du dialogue, que les praticiens pourront ensuite interpréter dans le champ de la psychiatrie en vue de l'établissement d'un diagnostic.

Avant mon arrivée, l'application permettait une gestion basique des utilisateurs (les praticiens), de leurs patients, et entretiens, ainsi qu'une première étape d'analyse de ces entretiens. Tout en étant en étroite collaboration avec l'ingénieur de l'équipe, ma mission principale consistait à enrichir l'application avec différentes fonctionnalités. Mon travail était, plus tard, intégré à la version officielle, s'il était validé par mes responsables.

L'application est basée sur une architecture client serveur. Le client est celui qui va envoyer des requêtes, alors que le serveur essaiera d'y répondre. Plus simplement, tout ce qui est "côté client" correspond globalement à ce qui est observable pour l'utilisateur, contrairement au serveur. Le client et le serveur se démarquent notamment par les langages informatiques utilisés.

3.1 Présentation générale des missions réalisées

Dans un premier temps, j'ai dû installer sur ma machine l'environnement adéquat et m'y familiariser. En effet, l'application possédait déjà plusieurs fonctionnalités, j'ai donc dû découvrir toutes les spécificités qui en découlaient. Pour cela j'avais notamment accès à GitLab. GitLab est une forge logicielle, permettant notamment l'hébergement et la gestion de projets, ainsi que l'intégration continue de code. Cela m'a permis principalement d'accéder au code écrit préalablement et au Wiki, ensemble de pages décrivant l'environnement et le processus de développement de l'application. De plus, il était indispensable d'apprendre à travailler avec Git, "*logiciel de gestion de versions décentralisé*"⁴. Chacune de mes missions était développée sur ma propre branche, puisque cela permet de pouvoir développer sans impacter la branche stable du projet.

Affichage du numéro de version Ma première mission consistait à afficher le numéro de version de l'application. Celui-ci devait pouvoir être accessible et perçu par les utilisateurs de SLODiM en cliquant sur un bouton *À propos*, accessible dans le menu déroulant principal, présent sur toutes les pages.

4. Source : Wikipédia



FIGURE 4 – Bouton *À propos*

Il s’agissait d’une mission d’introduction afin de mieux comprendre la façon de développer, les langages utilisés et le lien client-serveur. Le but était que j’avance et que je comprenne, dans la mesure du possible, seule. La mission aura duré environ 4 jours.

Complétion du formulaire du patient Ma deuxième mission était concentrée côté client : il s’agissait de compléter la création d’un formulaire concernant le patient. En effet, lors de la création d’un patient, l’utilisateur (praticien) doit remplir obligatoirement son nom et son prénom. Cependant, par la suite, il est possible de modifier les informations déjà entrées. J’ai donc ajouté d’autres champs disponibles lors de la modification des informations du patient : les initiales, le numéro de sujet, le sexe, la date de naissance, la ville de naissance, la date d’inclusion, le mode de vie, le parcours professionnel et la profession. Ces champs peuvent être des champs de texte, des date pickers pour les dates, un bouton radio pour le sexe ou encore des champs à sélectionner pour le parcours professionnel et la profession. De plus, les champs faisaient l’objet de tests. Par exemple, le sexe doit faire partie des champs proposés et les initiales du sujet ne doivent pas dépasser le nombre de 3 caractères. Si l’utilisateur se trompe, alors il n’est pas possible de mettre à jour les informations et un message écrit en rouge apparaît.

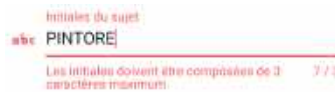


FIGURE 5 – Champ *Initiales* du formulaire

Ce projet nécessitait de travailler avec d’autres technologies méconnues au préalable par moi-même, comme notamment *Vuetify*, que nous développerons plus tard. Il a fait l’objet de nombreux tests ; sa réalisation a donc été plus longue, soit environ 9 jours de travail.

Affichage des informations du formulaire Suite à cela, ma troisième mission consistait à afficher les informations entrées dans la formulaire de mise à

jour, dans la page d'accueil du dossier patient. Pour cela, il fallait récupérer ces informations, puis les afficher côté client. Il était nécessaire de veiller à ce que les informations affichées soient uniquement celles qui avaient déjà été entrées par l'utilisateur. Par exemple, si la date d'inclusion n'avait pas été renseignée, il ne faut pas que celle-ci apparaisse. Enfin, dans un souci esthétique, le genre du patient devait être symbolisé par une icône et non un champ de caractères. La réalisation de cette mission a duré environ 6 jours. Au cours de celle-ci, j'ai rencontré des difficultés notamment dans l'affichage des dates, ce qui fera l'objet de ma prochaine mission.

Correction de code Mes prochaines missions avaient pour but d'améliorer certaines parties du code déjà existantes. Tout d'abord, les dates faisaient toujours l'objet de conversions car celles-ci n'étaient pas écrites dans le bon format. Une des missions a donc été de généraliser la conversion de celles-ci pour pouvoir, à chaque manipulation de dates, utiliser un même procédé et donc éviter la duplication de code.

Par ailleurs, l'application n'étant pas terminée, certains boutons, notamment présents dans l'analyse des entretiens, n'étaient pas utilisables et ne faisaient pas le lien avec d'autres pages. J'ai donc dû les rendre inaccessibles pour l'utilisateur en les désactivant.

Ces petites corrections auront pris environ 2 jours.

Analyse d'entretiens Mes prochaines missions concernaient plus précisément les entretiens. Tout d'abord, j'étais en charge de la rédaction d'expressions régulières, permettant la lecture, puis le chargement d'entretiens transcrits. Les expressions régulières constituent un ensemble de caractères décrivant les caractères possibles dans une expression.

A screenshot of a terminal window with a dark background and light-colored text. The text is arranged in three lines: the first line is '0001 R rouge', the second line is '00:00:01.566 - 00:00:02.241', and the third line is '(2.19)'. The text appears to be a regular expression pattern used for parsing speech data.

FIGURE 6 – Exemple de prise de parole

Une interaction est modélisée par 3 lignes :

- La première ligne comporte un identifiant (0001), une chaîne de caractères représentant la personne qui parle (**R**) et le discours tenu (**rouge**).
- La deuxième ligne correspond au début (00:00:01:156) et à la fin (00:00:02:241) de la prise de parole, en terme de temps.
- La troisième ligne est facultative et symbolise la durée de silence (2:19).

Les expressions régulières permettent donc à la fonction analysant les entretiens, de savoir sur quelle ligne elle est. Une fois qu'elle le sait, elle pourra identifier l'orateur, l'identifiant ou le discours par exemple, en les stockant dans des variables.

Une fois ces expressions écrites, ma mission consistait à compléter les algorithmes d'analyse et à écrire les tests associés. Par exemple, dans notre cas, un test consisterait à vérifier que l'identifiant vaut bien 0001.

Ces notions étaient nouvelles pour moi, il était difficile de m'y acclimater. J'ai finalement terminé au bout de 7 jours.

Adaptation pour plus de 2 personnes Actuellement, SLODiM permet l'analyse d'entretiens entre deux personnes. Le but de cette mission était d'écrire un code, permettant l'analyse d'entretiens avec plus de 2 personnes. L'idée était donc de réduire le nombre de personnes des textes comportant plus de 2 personnes, pour finalement arriver à 2. Concrètement, cela revient à écrire un code permettant de modifier l'attribut du nom de l'orateur pour chaque entretien. De façon à être intégré, mon code devait pouvoir s'exécuter à base de commandes, lancées à partir du terminal.

Cette mission a duré 3 jours. Elle était très difficile, du fait de la création de commandes.

Tri des entretiens Enfin, j'ai dû trouver un moyen de sauter certaines lignes, lors de l'analyse d'entretiens. En effet, tous les détails sont écrits. Par exemple, si un stylo tombe pendant l'interaction, le bruit sera modélisé par une balise, telle que [b]. Cela va sans dire que ces interactions ne sont pas forcément pertinentes, c'est pourquoi il fallait faire en sorte que celles-ci ne soient pas prises en compte dans l'analyse.

Cette mission a duré 2 jours et marque la fin de mon stage.

Étapes	Tâches	Durée (en jours)
1) Affichage de la version	A. Récupération du numéro de version dans le fichier + Création d'un objet ayant pour attribut ce numéro	2 (27 et 28 mai)
	B. Création d'un lien à propos	2 (29 mai et 2 juin)
	C. Réalisation des tests d'intégration	2 (3 et 4 juin)
2) Complétion du formulaire du patient	A. Mise à jour de la méthode update	3 (5, 8 et 9 juin)
	B. Mise à jour du formulaire + Écriture des règles que doivent respecter les champs	6 (10, 11, 12, 15, 16 et 17 juin)
	C. Réalisation des tests d'intégration	3 (18, 19 et 22 juin)
3) Affichage des informations du patient	A. Affichage côté client	6 (23, 24, 25, 26, 29 et 30 juin)
4) Correction de code	A. Généralisation de la conversion des dates	1 (1 ^{er} juillet)
	B. Suppression des boutons inutiles	1 (2 juillet)
5) Analyse des entretiens	A. Écriture des expressions régulières	1 (2 et 3 juillet)
	B. Complétion du code permettant le chargement de la transcription	3 (3, 6 et 7 juillet)
	C. Réalisation des tests unitaires pour les segments	1 (8 juillet)
	D. Réalisation des tests unitaires pour le parser	3 (9, 10 et 11 juillet)
6) Adaptation pour plus de 2 personnes	A. Création de la première commande	2 (15 et 16 juillet)
	B. Création de la seconde commande	1 et 1/2 (17 et 17 juillet)
7) Tri des entretiens	A. Écriture du code permettant le tri	2 (27 et 28 juillet)

FIGURE 7 – Calendrier des missions réalisées

3.2 Explication des missions

Toutes les missions m'étaient propres et ont été réalisées par moi-même avec l'aide, voire en collaboration avec mon encadrant, Pierre Lefebvre.

Technologies utilisées

- **Node.js** : Node.js est une plateforme en JavaScript, permettant notamment de créer des applications événementielles. Node.js m'a été imposée mais selon moi, ce choix était optimal du fait de sa particularité. En effet, il permet d'utiliser JavaScript côté client, comme c'était initialement le cas, mais aussi côté serveur. De plus, son efficacité repose sur son modèle non-bloquant. En effet, Node.js permet de lancer plusieurs fonctionnalités en même temps (dans la mesure du possible) et donc d'adapter au mieux le temps d'exécution.
- **Express** : Express est un framework⁵ connu de JavaScript, spécialisé dans le développement de serveur en Node.js.
- **Vuetify** : Vuetify est une implémentation de Material Design, c'est-à-dire un ensemble de règles de design proposées par Google. **Vue.js** est donc le framework de JavaScript associé, permettant la gestion de données, le découpage en composants et la communication serveur. Vuetify a permis une facilité de création du formulaire, de manière esthétique et complète.
- **Python** : Python est un langage de programmation interprété. Un langage est interprété si les instructions qu'on lui envoie sont transcrites en langage machine au fur et à mesure de leur lecture, d'après la définition de *OpenClassrooms*.
- **Test d'intégration** : Les tests d'intégration reposent sur le test de différents modules, qui sont au fur et à mesure imbriqués pour tester le fonctionnement global d'une fonctionnalité.
- **Test unitaire** : À l'inverse des tests d'intégration, les tests unitaires consistent à tester un module indépendamment des autres, dans une situation précise. En général, un test unitaire est propre à une méthode. On supposera que si le test fonctionne dans une situation choisie aléatoirement alors, le test est réussi et le module fonctionnel.
- **Mutateur** : Un mutateur, aussi appelé *setter* est une méthode permettant de modifier la valeur des attributs privés d'une classe.
- **Accesseur** : Un accesseur, aussi appelé *getter*, renvoie un attribut privé d'une classe.

Selon moi, ces technologies sont optimales car je suis fier du résultat qu'elles ont permis d'apporter et, même si la prise en main pouvait parfois être difficile, elles étaient, pour la plupart, intuitives et surtout très adaptées à la mission demandée. En effet, leurs choix se sont faits suite à des études. Ces technologies

5. "Désigne un ensemble cohérent de composants logiciels structurels, qui sert à créer les fondations ainsi que les grandes lignes de tout ou d'une partie d'un logiciel". Source : Wikipédia

et donc l'utilisation du client/serveur sur la même machine ont finalement été retenues car :

- Elles offrent une certaine ergonomie,
- Elles permettent de faire tourner l'application sur un appareil mobile,
- Elles offrent des possibilités d'envoi/réception de média,
- Elles permettent la conservation des données patients chez le professionnel,
- Certains autres traitements et bibliothèques utilisées étaient trop lourds,
- D'autres technologies induisaient un manque de visibilité sur les indices linguistiques qui seront proposés et nécessitaient donc un besoin de flexibilité.

3.2.1 Affichage du numéro de version

Le but de cette mission était de récupérer le numéro de version auprès du serveur et de l'afficher dans le client. Il s'agit d'une mission assez importante et évidente puisque la plupart des applications et logiciels permettent d'avoir accès facilement à ce numéro. De plus, il peut être utile de connaître le numéro de version pour savoir si le logiciel est à jour ou non, notamment en cas de problème.

Cette mission a été intégralement codée à l'aide de *Node.js*, en JavaScript.

Côté serveur J'ai dû créer un module se chargeant de récupérer le numéro de version. Ce dernier se trouve dans un fichier créé préalablement par moi-même, à la racine du serveur, et qui sera, à terme, rempli dans le processus de déploiement. Plusieurs cas doivent être créés :

- Erreur code 404 (**Not Found**) : Le fichier n'est pas trouvé.
- Erreur code 500 (**Internal Error**) : Pour toute autre erreur. L'erreur interne est, en effet, une erreur assez vaste et regroupe différents types d'erreurs.
- Code 200 (**Ok**) : Création et envoi d'un objet JSON (format de données) contenant un champ *'version'*.

Ces résultats correspondent à des codes HTTP. Le code HTTP est le résultat des requêtes, réalisées par le serveur. Celles-ci sont envoyées, grâce à *Express*. Concrètement, la méthode `existsSync` était d'abord utilisée pour vérifier l'existence du fichier. Puis, celui-ci était lu grâce à `readFileSync`.

Côté client J'ai créé un lien *À propos* dans le menu principal. Ce lien ouvre une page où est affichée la version, préalablement récupérée par le biais d'une méthode du serveur, renvoyant l'objet JSON.

Tests Des tests d'intégration ont été réalisés par le serveur, permettant de vérifier notamment que :

1. Un utilisateur est bien connecté,

2. Le fichier contenant le numéro de version existe,
3. Le numéro de version a bien été récupéré.

Chacun des cas de tests vérifie que la réponse ou l'erreur renvoyée est bien celle à laquelle on s'attend. Par exemple, si le fichier contenant le numéro de version n'existe pas, l'erreur levée doit être 404 Not Found.



FIGURE 8 – Affichage de la version

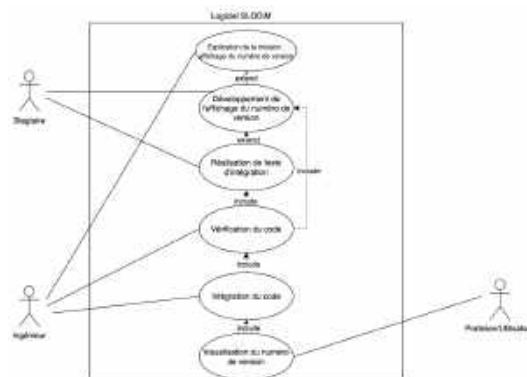


FIGURE 9 – Diagramme de cas d'utilisation du développement de l'affichage du numéro de version

3.2.2 Complétion du formulaire du patient

Cette mission avait pour but la complétion du formulaire du patient en ajoutant de nouveaux champs, afin de le rendre plus riche. Mon travail a donc été de compléter du code déjà existant.

Cette fois, *Node.js* a encore été utilisé côté serveur mais, *Vue.js* l'a aussi été côté client.

Côté serveur Il s'agissait de compléter une méthode nommée `update`. Celle-ci permettait la mise à jour, dans une base de données, des informations entrées. Bien que le formulaire n'est pas censé pouvoir être mis à jour, si les champs ne

sont pas corrects, il est quand même important de traiter cette possibilité. Ainsi, avant de mettre à jour les informations, tous les critères que doivent remplir chaque champ sont vérifiés. En cas de champ non-conforme, une erreur HTTP : 400 **Bad Request** est alors envoyée, de la même manière que la première mission, grâce à *Express*. Par ailleurs, le modèle de la base de données est renseigné et contient une définition des différents champs.

Côté client La première étape consistait à mettre à jour le formulaire en ajoutant les différents champs : les initiales, le numéro de sujet, le sexe, la date de naissance, la ville de naissance, la date d'inclusion, le mode de vie, le parcours professionnel et la profession, grâce à *Vue.js* ; ainsi que les valeurs que ces champs pouvaient prendre, dans le cas où celles-ci étaient contraintes. Par exemple, le choix du parcours professionnel doit se faire parmi une liste de propositions. De manière similaire au côté serveur, les champs étaient vérifiés. La mise à jour du formulaire n'était alors possible que si tous les champs étaient validés. Dans le cas où un champ n'est pas conforme, un message rouge est affiché et il n'est pas possible de cliquer sur *Mettre à jour*.

Tests Des tests d'intégration ont été réalisés. Ceux-ci vont tester plusieurs cas possibles lors de la mise à jour des données :

1. Aucun utilisateur n'est connecté,
2. La connexion se fait proprement,
3. Certains champs obligatoires ne sont pas entrés,
4. Les champs ne remplissent pas tous les critères,
5. Certains champs uniques sont utilisés à plusieurs reprises,
6. L'utilisateur connecté n'a pas les droits sur le patient à modifier,
7. Les données mises à jour sont inchangées,
8. Les données ont bien été mises à jour.

Chacun des cas de tests vérifie que la réponse ou l'erreur renvoyée est bien celle à laquelle on s'attend. Par exemple, si un champ ne remplit pas tous les critères, la réponse est une erreur 400 **Bad Request**.

Mise à jour des informations du patient

Prénoms et nom
 Angeline 0 / 44

Email
 @angeline@mail.fr

Initiales du sujet 0 / 3
 123 123

Numéro du sujet 3 / 3

Date de naissance
 1999-02-11

Date d'inscription
 2020-06-09

Commune de naissance 0 / 64
 Sexe
 Féminin

Niveau scolaire/parcours
 Bac+3 - Licence, licence professionnelle

Mode de vie
 Célibataire

Profession/Source de revenus 0 / 44

ANNULER METTRE À JOUR

FIGURE 10 – Formulaire de mise à jour



FIGURE 11 – Diagramme de cas d'utilisation du développement de la complétion du formulaire du patient

3.2.3 Affichage des informations du formulaire

Cette partie consiste à afficher les informations entrées du patient sur la page d'accueil. En effet, cela permet d'avoir une vision globale et efficace des informations. Si l'utilisateur veut modifier un champ, cela est directement possible sur cette page.

Côté client Cette mission se situe uniquement côté client. En effet, il s'agit seulement d'afficher des données récupérées directement de ce côté. Pour cela, *Vue.js* a été utilisé. Bien que cette mission puisse sembler simple, il y avait deux spécificités à prendre en compte. Tout d'abord, la conversion de date et son affichage ont posé problème à plusieurs reprises. De plus, les champs comme *Parcours* (qui correspondent à des choix multiples) sont, en réalité, identifiés par des clés numériques. Il fallait donc penser à utiliser la fonction `find`, pour avoir accès à la valeur associée, pour l'affichage. Sans cela, des nombres auraient été visibles, au lieu d'intitulés.

Cette partie ne fait pas l'objet de tests, puisqu'elle ne concerne que le client.



FIGURE 12 – Affichage des informations du formulaire



FIGURE 13 – Diagramme de cas d'utilisation du développement de l'affichage des informations du patient

3.2.4 Correction de code

Généralisation de la conversion des dates Les dates sont souvent utilisées dans l'application mais ne peuvent jamais être affichées, sans travail préalable. En effet, elles ne sont pas écrites dans le bon format. Après avoir rencontré des difficultés dans l'affichage de dates dans la mission précédente, mon encadrant a jugé nécessaire de généraliser cette conversion. Pour cela, une méthode a été créée en JavaScript, à l'aide de *Vue.js*.

```
// Return date in the right format
convertDate(date) {
  if (date) {
    return new Date(date).toLocaleString('fr-FR').split(' ')[0];
  }
  return '';
},
```

FIGURE 14 – Méthode de conversion de dates

Dans cette méthode, nous vérifions tout d'abord que le champ date n'est pas nul. S'il est nul, une chaîne de caractères vide est renvoyée. Sinon, la date est convertie en date française et tout ce qui apparaît après un espace disparaît. En effet, la date du 11 février 1999 apparaissait sous la forme : 1999-02-11T00:00:00:00Z. Ainsi, cette méthode a été utilisée plusieurs fois, de différentes façons, pour éviter des problèmes et de la duplication de code, d'où sa nécessité.

Suppression des boutons inaccessibles Toujours à l'aide de *Vue.js*, certains boutons ont dû être rendus inaccessibles, le logiciel n'étant pas terminé. Cela était relativement simple, puisqu'il suffisait de rajouter un attribut `disabled` à la balise des boutons correspondant.

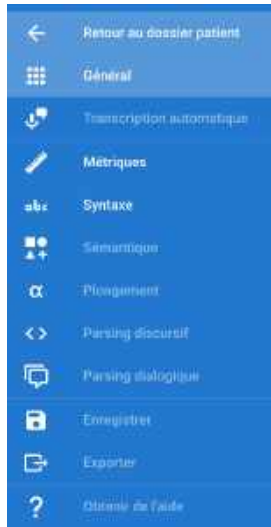


FIGURE 15 – Boutons inaccessibles

Ces modifications n'ont pas fait l'objet de tests.

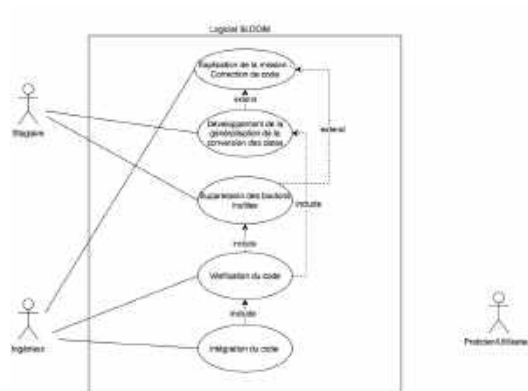


FIGURE 16 – Diagramme de cas d'utilisation de la correction de code

3.2.5 Analyse d'entretiens

Cette mission a été codée intégralement en *Python*.

Développement Tout d'abord, comme explicité précédemment, il m'était demandé de charger les transcriptions à l'aide de l'écriture d'expressions régulières.

```

# Tag used to parse plain file
K_SPEECH_LINE_REGEX = '([0-9]{4})\s{1}([^\s]+)\s{1}(.+)'
K_TIMECODE_LINE_REGEX = '\s+([0-9]{2};){2}([0-9]{2})\s{1}([0-9]{3})\s{1}([0-9]{2};){2}([0-9]{2})\s{1}([0-9]{3})'
K_SILENT_LINE_REGEX = '\s+([0-9]+\.[0-9]+)\s{1}'

```

FIGURE 17 – Rédaction des expressions régulières

Chaque variable correspond à la transcription d'une ligne, représentant une partie de la prise de parole. Plus explicitement, `K_SPEECH_LINE_REGEX` correspond à la ligne comportant l'identifiant, l'orateur et le discours ; `K_TIMECODE_LINE_REGEX` équivaut à la ligne comportant le temps du début et de la fin de la prise de parole ; `K_SILENT_LINE_REGEX` correspond à la durée du silence, s'il y en a eu un. De plus, chaque caractère a une symbolique bien précise. À titre d'exemple, nous allons étudier la première expression régulière :

- `[0-9]` signifie que l'expression débute par un chiffre compris entre 0 et 9. Cependant, celui-ci est suivi de `{4}` qui implique la répétition de ce caractère 4 fois. L'expression débutera donc par 4 chiffres compris entre 0 et 9.
- Le tout est suivi de `\s` qui symbolise un espace. L'expression comporte donc pour le moment 4 chiffres compris entre 0 et 9 (l'identifiant), suivi d'un espace.
- `^` signifie "Tout sauf ..." donc la combinaison de `^` et `\s` signifie "Tout sauf un espace". Le `+` signale que ce caractère peut apparaître une ou plusieurs fois. Le tout correspond donc à une chaîne de caractères ne comportant pas d'espaces, il s'agit du prénom de l'orateur. Ensuite, il y a de nouveau un ou plusieurs espaces, représentés par `\s`, suivi de `+`
- Enfin, le `.` dénote un caractère, peu importe lequel. Puisqu'il est suivi d'un `+`, cela correspond à sa répétition et indique donc le discours.

Une fois ces expressions écrites, j'ai pu rédiger des méthodes permettant le chargement de la transcription des entretiens. Cela revenait à parcourir un fichier texte contenant l'entretien, identifier de quelle ligne il s'agissait pour finalement, créer un objet, qu'on appellera *segment*, contenant les attributs d'une prise de parole : identifiant, orateur, discours, temps du début, temps de la fin et silence. Tous les segments sont finalement stockés dans une liste, contenant donc toutes les données de l'entretien.

Tests Des tests unitaires sur les segments ont d'abord été réalisés. En tout, 3 tests ont été écrits, chacun correspondant à une méthode :

- Le premier test permet de tester le constructeur de `Segment`. En effet, cette méthode permet de créer des segments à partir du discours. Le test vérifie que le discours a bien la même valeur que celui entré et, que les autres attributs sont nuls (puisque'ils ne sont pas encore initialisés).
- Le deuxième test permet de contrôler que les *mutateurs* fonctionnent. Ainsi, le test inspecte donc que toutes les valeurs des attributs sont bien mises à jour.

- La troisième test teste, cette fois, l'*accesseur* du discours. Le but du test était donc de comparer le résultat de l'accesseur et le résultat attendu.

Puis, des tests unitaires pour l'ensemble des segments et donc la transcription informatique de l'entretien ont été réalisés. De nouveau, j'ai dû écrire 3 tests :

- Le premier permet de vérifier que l'objet a bien été créé et que ses attributs sont initialisés.
- Le deuxième teste la méthode `_get_speech`. Cette méthode retourne une liste contenant tous les tours de parole d'un entretien. Ainsi, le test consiste à comparer les tours de parole attendus à ceux provenant de `_get_speech`.
- Le dernier test permet de vérifier si la transcription informatique a bien été réalisée, en comparant tous les attributs d'un entretien à ce qui est attendu.

Cette mission était urgente à réaliser, afin de pouvoir la faire tester à des personnes de l'équipe. Le code écrit sert actuellement.

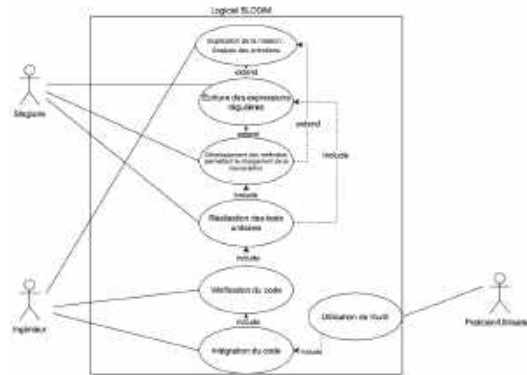


FIGURE 18 – Diagramme de cas d'utilisation du développement de l'analyse des entretiens

3.2.6 Adaptation pour plus de 2 personnes

Cette mission n'était pas la plus impérative, il s'agit plutôt d'un projet.

Toujours en *Python*, j'ai dû créer des scripts qui avaient diverses fonctionnalités. Pour cela, j'ai utilisé le module `argparse`, permettant l'analyse des arguments de lignes de commandes. Le premier script permet d'afficher tous les orateurs d'un entretien. Le second permettait de réduire le nombre d'orateurs à 2, en modifiant leur nom. Pour cela, l'utilisateur choisit les 2 noms qu'il souhaite attribuer et ceux-ci le sont, à tour de rôle.

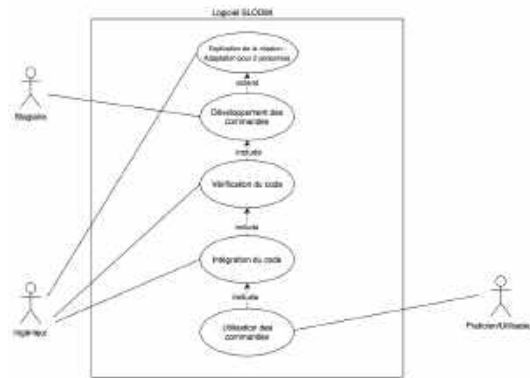


FIGURE 19 – Diagramme de cas d'utilisation du développement de l'adaptation pour 2 personnes

3.2.7 Tri des entretiens

En *Python*, j'ai dû faire en sorte que les lignes d'entretiens qui transcrivaient un bruit soient évitées. Dans ce cas, l'orateur avait pour nom **default**. Ainsi, une liste comportant tous les noms d'orateurs à éviter a été créée :

`K_SPEAKERS_AVOIDED`. Pour le moment, celle-ci ne contient que **default**, mais elle pourra être complétée selon les besoins de l'équipe. Pour réaliser le tri, la fonction `filter` a été utilisée, à la fin du chargement de la transcription, sur le tableau contenant tous les segments de l'entretien.

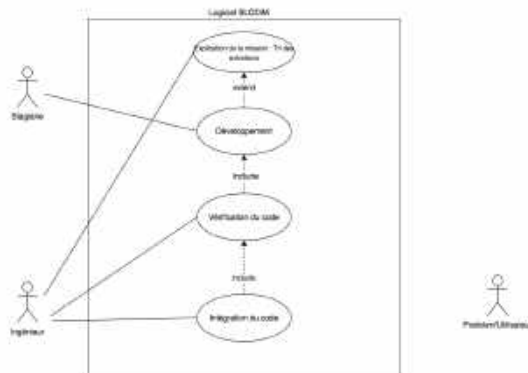


FIGURE 20 – Diagramme de cas d'utilisation du développement du tri des entretiens

3.3 Procédure d'intégration du code avec Git

Une fois le code écrit de mon côté, il fallait pouvoir l'intégrer à la branche stable du projet. Pour cela, une succession de commandes *Git* était utilisée :

1. `git add` est utilisé pour suivre un fichier.
2. `git commit` est utilisé pour enregistrer les modifications faites sur les fichiers suivis (grâce à `git add`), dans le dépôt.
3. `git fetch` permet de ramener les branches distantes sur mon dépôt local.
4. `git checkout develop` sert à changer de branche et se placer sur *develop*, branche stable du projet.
5. `git pull` permet de mettre à jour la branche par rapport à sa version dans le dépôt distant.
6. `git checkout maBranche` me déplace sur ma branche.
7. `git merge develop` fusionne *develop* dans ma branche.
8. `git push` permet de mettre le tout à jour, avec les objets distants.

Enfin, une *merge request* était faite pour que mon encadrant vérifie le code écrit. Si le code est validé, il pourra alors être intégré.

3.4 Conclusion : Le développement d'une application à travers divers outils

Au cours de ces semaines de stage, je suis plutôt satisfaite de mes réalisations. En effet, j'ai pu découvrir des technologies basiques qui me serviront par la suite en tant qu'informaticienne, comme Git. J'ai aussi pu découvrir, de façon plus poussée et plus aboutie qu'auparavant, le Web, à l'aide de *Node.js* et *Vue.js*. En effet, la récupération du numéro de version, la mise à jour du formulaire, ainsi que son affichage m'ont fait découvrir de nombreuses facettes de ces technologies. Et, j'ai pu m'améliorer en Python, en découvrant notamment la création d'objets, en permettant l'analyse d'entretiens. Ce stage a donc été très enrichissant, notamment grâce à la diversité des missions proposées et des technologies utilisées.

Malheureusement, du fait de la crise sanitaire que nous vivons actuellement, j'ai dû travailler sur toute la durée du stage, en télétravail. Bien que mes responsables fût très présents et disponibles, cela pouvait entraver mon travail personnel. En effet, il était parfois difficile de communiquer et de trouver des solutions à distance, ainsi que de travailler dans un milieu non-professionnel.

En conclusion, je suis fière de mon travail puisqu'il est, à ce jour, intégré à la version stable du projet et servira donc aux praticiens.

4 Conclusion générale : Le développement d'un logiciel au sein d'une organisation innovatrice

J'ai effectué mon stage de 8 semaines au sein du Laboratoire Lorrain de Recherches en Informatique et ses Applications, en tant que développeuse d'un logiciel informatique à l'attention de praticiens, permettant l'analyse de leurs entretiens avec leurs patients. Le stage s'est intégralement déroulé en télétravail. À mon arrivée, il existait déjà une première version de celui-ci. Ma principale mission était donc d'aider à l'améliorer, en y ajoutant des fonctionnalités.

Le travail d'analyse résultant de l'écriture de ce rapport m'a permis, dans un premier temps, de pouvoir caractériser le LORIA. En effet, il s'agit d'un grand laboratoire de recherche en Lorraine. Celui-ci prône l'innovation, les interactions entre différents membres de la hiérarchie - permettant une facilité de communiquer de manière ascendante, une décentralisation de la prise de décision et l'importance des formations notamment. Bien que ce type d'organisation puisse paraître trop ouvert et libre, les employés semblent s'y sentir bien. En effet, je n'ai remarqué ni problèmes de communication, ni mésententes par rapport au travail. De plus, malgré le télétravail généralisé au début de mon stage, l'avancée des projets ne semblaient pas être retardée et, l'ambiance semblait détendue et amicale.

Dans un second temps, les différentes missions informatiques que j'ai réalisées m'ont permis de compléter drastiquement mes connaissances. En effet, j'ai eu la chance de découvrir de nouvelles technologies comme *Node.js* et *Vue.js*. J'ai aussi pu améliorer mon niveau en *Python* et, j'ai découvert le logiciel de gestion *Git*, qui me sera indéniablement utile dans mes prochains projets. Par ailleurs, j'ai pu découvrir l'avancée d'un projet informatique au sein d'une organisation. En effet, la procédure est très rigoureuse. Celle-ci débutait par le recueil du besoin, fait par l'ingénieur de l'équipe. Puis, débutait le développement. Bien souvent, il était suivi d'une phase de tests. Ensuite, le travail était validé par mes supérieurs pour finalement, être intégré à l'application.

L'annonce du télétravail fut une déception puisque, bien que ce stage avait pour but de renforcer mes connaissances informatiques, il devait aussi me permettre de découvrir la vie professionnelle, que ce soit à travers le travail collaboratif, les réunions d'équipes ou encore, les interactions sociales entre collègues. Néanmoins, avec du recul, le LORIA s'est parfaitement adapté à cette situation. En effet, je n'ai jamais eu de soucis pour communiquer avec des membres de l'équipe. De plus, des salons informatiques ont été mis en place pour ne pas perdre les échanges non-formels qui avaient lieu en présentiel. Enfin, malgré le fait que je sois physiquement absente, je ne me suis pas sentie oubliée. De fait, je communiquais quotidiennement avec mon encadrant, participais aux conversations de groupe et aux réunions hebdomadaires. Aussi, je possédais une adresse mail, où je recevais beaucoup d'informations concernant l'organisation, me faisant me sentir totalement intégrée à l'équipe.

Outre le fait d'avoir pu découvrir partiellement la vie en entreprise, les missions informatiques ont été très bénéfiques. En effet, elles m'ont permis de découvrir de nouvelles technologies, mais surtout, en travaillant avec mon encadrant, j'ai pu découvrir de nouvelles façons de travailler et de réagir en cas de problèmes. Je pense personnellement que ce stage m'a fait mûrir dans mon travail et m'a appris comment procéder. Par exemple, avant, je ne comprenais pas forcément l'intérêt des tests, ni l'apprentissage de Git. Or, dans une organisation telle que le LORIA, ces derniers demeurent indispensables.

Je ne pense pas que j'agirais différemment si je devais refaire ce stage. Certes, mon travail ne fut pas parfait dès le début, mais, ce stage fut fondateur et m'a appris beaucoup de choses, que ce soit professionnellement ou personnellement, et, c'est en ayant fait des erreurs que j'ai pu, d'autant plus, les assimiler.

En conclusion, ce stage fut très bénéfique sur tous les points. Il m'a permis de découvrir le travail au sein d'un laboratoire de recherche, qui pourrait m'intéresser dans ma vie professionnelle future.

5 Webographie

1. Loria.fr
2. Univ-lorraine.fr
3. Fr-hermite.univ-lorraine.fr
4. Team.inria.fr/semagramme
5. Git-scm.com/docs
6. Regardssociologiques.free.fr/l3suo2019.htm
7. *Évaluation de l'unité : Laboratoire lorrain de recherche en informatique et ses applications* - Campagne d'évaluation 2016-2017, réalisée par le Haut Conseil de l'Évaluation de la Recherche et de l'Enseignement Supérieur
8. Document interne au LORIA, 2020 : *Présentation de l'unité de recherche*
9. Document interne au LORIA, 2019 : *Présentation LORIA*
10. Document interne au LORIA, 2019 : *Le LORIA en bref*
11. Marion CRÉHANGE et Marie-Christine HATON, 2007, *L'informatique universitaire à Nancy : un demi-siècle de développement dans L'Histoire de la Lorraine, Le Pays Lorrain*