# **Dossier Projet**

**Discord Communautaire Simplon**

# **Rédigé par Benjamin Gamache**



# **Résumé**

I chosed to present the Community Discord Simplon Server

This project consist of centralizing each entities of Simplon on a same Discord Server.

Our team discussed with Simplon teams members and as learner we saw some issues with Simplon communication.

The setup of communications supports of Simplon was extremly messy and so we had to provide a solution to make those process less tedious.

That was the main idea of this project.

I mainly focused on Back End developpement with NestJS & DiscordJS as Bot Developpement Framework

I firstly worked on this project as project for real client with my course mates.

We were 4 from 3 first months, working with Agile method using SCRUM.

When intership period arrived my mates gone in their company to work as intern.

I decided to take this project as internship project because it was an interesting project.

I refactored API side and Bot side from the first month of my internship.

And i'd worked on new features to implement those to Onboarding Bot whose

goal was to simplify the Onboarding process of new learners.

This project was firstly designed to be a set of 1 API, 1 Database, 1 Web Management & Monitoring Dashboard & 7 Bots, but at this moment it's 1 finalized bot instead of 7.

## Présentation personnelle

J'ai commencé à coder à l'âge de 8 ans, même s'il s'agissait uniquement de modifications mineures du code des pages Web. J'appréciais le fait que du code que je ne comprenais pas puisse donner forme à des pages Web.

Au fil du temps, ma compréhension de l'informatique a évolué, me permettant de me tourner vers des langages plus avancés. Ma passion pour le développement s'est intensifiée.

De plus, j'ai acquis mes compétences de manière autodidacte, sans suivre d'études dans le domaine de l'informatique. Ce qui était autrefois une passion est devenu un désir professionnel.

J'ai arrêté l'école à l'âge de 16 ans car le système scolaire ne me convenait pas.

C'est pourquoi, il y a quelques mois, j'ai décidé d'intégrer une formation de Concepteur Développeur d'Applications afin de renforcer mes compétences et de passer du simple développement à des travaux de conception et de réflexion préalables à la création de projets.

Grâce au projet que j'ai réalisé pendant ma formation, j'ai pris conscience des opportunités offertes par Discord dans le monde professionnel. On peut dire la même chose de Wordpress. Qui aurait pu imaginer que Wordpress deviendrait si lucratif ?

C'est pourquoi, à l'issue de ce projet, j'ai décidé de créer ma start-up spécialisée dans le développement de bots Discord.

# **Présentation du projet**

### Le client : Simplon

Dans ce projet, notre client était l'organisme de formation **Simplon Hauts-de-France** (Simplon HDF). Simplon compte 21 000 apprenants et dispose de 7 fabriques numériques dans la région Hauts-De-France.

**La communication est un enjeu majeur** pour Simplon HDF, et le principal support de communication utilisé est Discord, qui permet les échanges entre les équipes et les apprenants.

### Contextualisation du projet

Dans le cadre de ce projet, nous nous sommes concentrés sur la manière dont Simplon HDF communique.

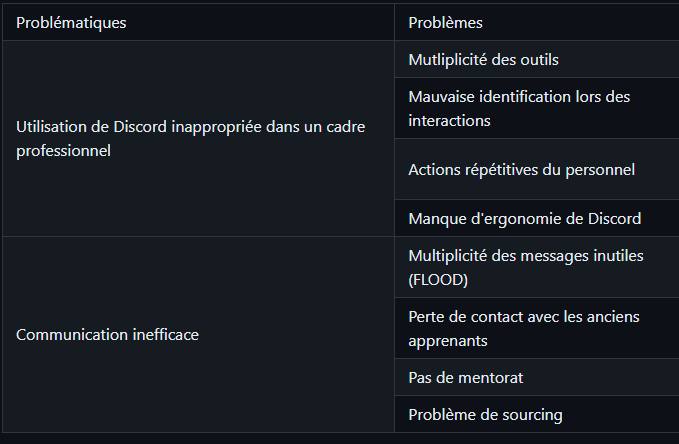
En tant qu'apprenants, nous avons remarqué que les équipes de Simplon HDF effectuent des **actions répétitives** qui pourraient être **automatisées facilement**.

De plus, la **communication**, notamment avec les apprenants, est **fastidieuse**.

En tant qu'utilisateurs réguliers de Discord, notre équipe s'est automatiquement tournée vers Discord en guise de support de communication principal pour ce projet.

### Recueil du besoin Client

Pour recueillir les besoins du client, notre équipe a d'abord discuté avec les différents membres des équipes Simplon. Suite à ces discussions, nous avons dressé une liste des problèmes les plus fréquemment rencontrés par les membres des équipes Simplon.



1. Multiplicité des supports de communication**: Concernant ce problème, notre équipe a constaté qu'il existait une multitude de serveurs Discord utilisés par les membres de Simplon pour communiquer avec différentes entités. Par exemple, un formateur peut avoir accès à 12 serveurs Discord (sur une période d'un an et demi) réservés aux promotions auxquelles il participe, sans compter les autres serveurs nécessaires pour communiquer avec les équipes Simplon.**
2. Mauvaise identification lors des interactions**: Discord utilise nativement un système de pseudonyme, ce qui n'est pas adapté à une utilisation professionnelle où l'identification claire des utilisateurs est nécessaire. Cela peut entraîner des complications pour identifier les personnes avec lesquelles nous souhaitons communiquer via Discord.**
3. Actions répétitives du personnel**: Il a également été observé que les membres des équipes Simplon, en particulier les formateurs, effectuent de nombreuses actions répétitives. Par exemple, lors du démarrage d'une nouvelle promotion chez Simplon HDF, un formateur doit effectuer les étapes suivantes pour permettre aux apprenants de communiquer entre eux (Annexe 1) ces étapes doivent être répétées à chaque nouvelle promotion.**
4. Manque d’ergonomie de Discord**: Les membres des équipes ont également exprimé leur malaise quant à l'utilisation de Discord, notamment dans un contexte professionnel. Ils ont relevé des lacunes ergonomiques qui rendent l'utilisation de Discord moins confortable.**
5. Multiplicité des messages inutiles**: Il a également été remarqué que les apprenants ont tendance à envoyer des messages de rappel aux membres des équipes dans des situations dans lesquelles ils n'ont pas reçu de réponse à leurs sollicitations. Bien que cela puisse sembler anodin, cela génère des messages inutiles et constitue une source de distraction pour les membres des équipes.**
6. Perte de contact avec les anciens apprenants**: Certains membres des équipes Simplon nous ont fait part de la difficulté à maintenir le contact avec les anciens apprenants. En effet, dans la configuration actuelle de l'organisme de formation, il n'existe pas de solution concrète permettant de rester en contact avec les anciens apprenants. Une fois leur formation terminée, ils quittent le serveur Discord dédié à leur promotion, ce qui entraîne une perte de contact.**
7. Problème de sourcing**: Les membres ont également signalé des problèmes de sourcing. Selon eux, il est relativement difficile de trouver des profils correspondant aux critères des promotions et de les intégrer.**

### L’équipe en charge du projet

**BOURREZ Bastien** alias le pro Discord

**PHILIPPE** **Nelson** alias l'aventurier

**LEROY Cédric** alias l'encyclopédie / rédacteur en chef

**GAMACHE Benjamin** votre rédacteur

### Proposition de solution :

### Notre équipe s'est interrogée sur la meilleure solution entre l'utilisation de **Discord** et le développement d'une application spécifique pour **Simplon**.

### Utiliser **Discord** présente les avantages suivants :

### Il est déjà utilisé par la majorité des membres des équipes et des apprenants.

### Il est gratuit.

### Il constitue déjà un support de communication stable

### Il est flexible et adapté à tout type d'équipe de développement.

### En revanche, développer une application spécifique représente les contraintes suivantes :

### Ajouter un nouveau support de communication pour tous les utilisateurs.

### **Engendre des coûts** en termes de graphisme.

### Nécessite un hébergement de serveur plus important.

### Requiert des **connaissances avancées** dans certains langages de programmation.

### C'est pourquoi nous avons choisi d'utiliser **Discord** plutôt que de développer une application à partir de zéro.

# **Gestion du projet**

Avant de démarrer un projet, il est important de se poser les bonnes questions, telles que la façon dont le projet devra être géré, quels objectifs le projet se fixe, et bien d'autres questions auxquelles mon équipe a tenté d'apporter les réponses les plus adaptées possibles.

## **Méthodologie de travail Agile**

**La méthodologie Agile** est une méthodologie **itérative** permettant le découpage des différents cycles de développement d'un projet en "**sprint**", chaque sprint à une durée moyenne d'1 à 4 semaines, à l'issue de ce sprint, une version fonctionnelle peut être présentée au client.

De plus, une équipe Agile interagira de façon **auto-gérée**, au sein d'une équipe Agile, la **collaboration** se fait entre tous les membres de façon étroite, le **partage des responsabilités** est aussi un des aspects clés de cette méthodologie, cette proximité au sein de l'équipe Agile permet une **communication continue**.

Agile apporte une livraison continue / itérative du produit au client, à chaque fin de sprint, le client peut récupérer une version du produit fonctionnelle et en tirer parti.

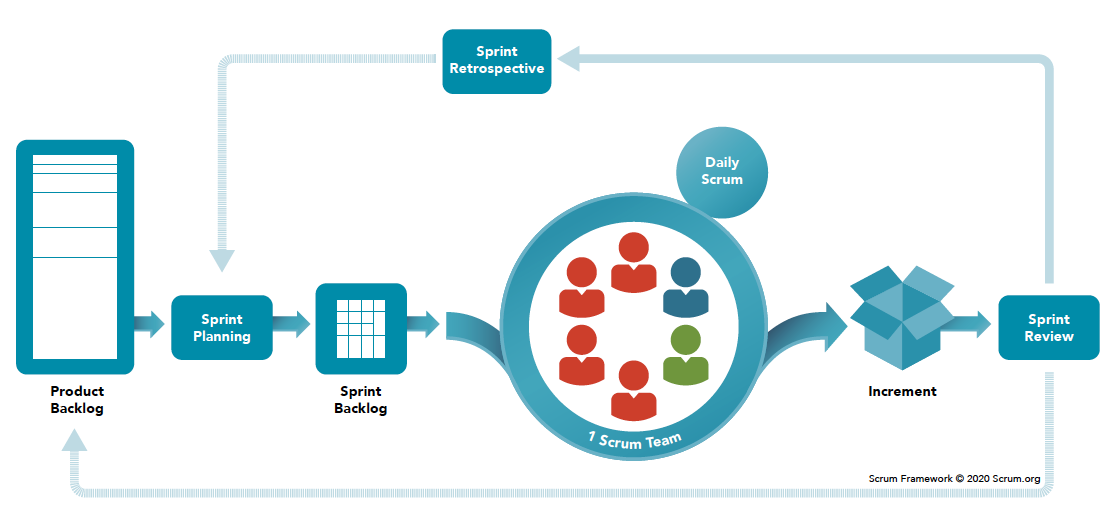
Les tâches d'un projet sont priorisées au sein d'un **Product Backlog** selon la valeur ajoutée pour le client, il est nécessaire d'établir un Product Backlog en priorisant les tâches à effectuer dès le début afin de permettre au client de tirer avantage au plus vite du produit.

Aussi, Agile considère que les besoins et les exigences du projet évoluent tout au long de la vie de ce dernier, avec Agile, **l'équipe peut s'adapter** à ces changements et effectuer des ajustements en fonction des nouvelles informations fournies par le client après qu'il ait reçu la dernière version fonctionnelle du produit.

Dû à ces délimitations des cycles de développement flexible, Agile permet à l'équipe de développement de s'améliorer de façon continue en mettant en place des **rituels** tels que les **Sprints Reviews** et les **Sprints Retrospectives** qui ont lieu à la fin de chaque sprint, permettant aux différents membres de l'équipe de discuter des succès et des difficultés rencontrées lors du dernier sprint.

De plus, Agile inclut une **métrique** : **La vélocité** qui est recalculée à chaque fin de sprints, de cette façon les membres de l'équipe sont capables d'évaluer la quantité de travail achevée lors du sprint.

### **Le framework SCRUM**



SCRUM ou littéralement **Mêlée** est un framework permettant aux équipes qui le mettent en place d'adopter une approche Agile et d'apporter de la flexibilité au projet.

Le framework SCRUM permet à chaque membre d'une équipe d'avoir un rôle à jouer dans la prise de décision concernant un projet.

La taille d'une équipe SCRUM peut varier, mais il est important de savoir que SCRUM est plus adaptée à des équipes à effectif réduit, en effet, à partir d'un certain nombre de membre, peu importe la méthodologie utilisée au sein d'une équipe, **la communication devient de plus en plus complexe**.

#### Les quatre grandes valeur de SCRUM

1. **Individus & Interactions > Processus & Outils** : Une équipe engagée crée un produit de valeur estimable
2. **Un produit fonctionnel > Une documentation exhaustive** : La documentation d'un produit est essentielle, cependant une équipe doit se concentrer sur les fonctionnalités d'un produit plutôt que sur la rédaction d'une documentation.
3. ***Collaboration avec le client > Négociation contractuelle*** : Il est essentiel de placer le client au sein du processus de développement du produit, les retours du client sur le produit sont une source d'information précieuse, une bonne collaboration entre le client et l'équipe SCRUM permet l'élaboration d'un produit fini le plus fidèle aux exigences du client.
4. **Adaptation au changement > Suivi d'un plan** : Dans SCRUM l'apport de modifications aux projets n'est pas un obstacle, étant donné que le produit se construit de façon itérative, en plaçant le client au centre du processus de développement, il est plus facile pour les équipes d'effectuer des ajustements concernant le projet tout au long de ces cycles de développement, plutôt que d'avoir à revoir complètement certains aspects du produit à la fin du développement de ce dernier.

#### Les rôles dans SCRUM

##### Le Product Owner

##### Le Product Owner est le responsable de la **liaison entre le client et l'équipe de développement**, son rôle est de comprendre le client et de retranscrire sa demande aux équipes de développement, **les tâches** du Product Owner sont les suivantes :

##### De créer et gérer un **Product Backlog** en prenant en compte les **tâches à prioriser**.

##### **Collaborer avec les équipes** afin de s'assurer que tous les membres comprennent le Product Backlog du projet.

##### Fournir à l'équipe les prochaines fonctionnalités à **livrer.**

##### Décider de la **durée des sprints**.

##### Le SCRUM Master

##### Le SCRUM Master est le membre **responsable de l'application de SCRUM** au sein de son équipe, il analyse perpétuellement les **optimisations possible**s à apporter en termes d'application SCRUM au sein de son équipe, **il s'assure de comprendre ce que l'équipe doit réaliser** et aide celle-ci à **optimiser la transparence du projet** et le **flux de livraison** de ce dernier, il gère les **ressources** humaines et logistiques d'une équipe pour planifier les sprints.

##### Il organise les cérémonies SCRUM.

##### L’équipe de développement

L'équipe de développement est constituée de **5 à 7 développeurs** la plupart du temps, ces développeurs travaillent en **étroite collaboration**, ce qui permet d'aider chaque membre de l'équipe de développement en difficulté afin de ne pas se retrouver avec un **retard dans la livraison** du produit à la fin du sprint.

#### Les rituels

##### Sprint planning

Le Sprint Planning est organisé par le **SCRUM Master** afin d'établir la liste des **User Stories** à mettre dans le prochain sprint, l'équipe s'accorde à dire que toutes les User Stories sont en mesures **d'être complétées** à l'issue du sprint (notamment en se basant sur la **vélocité** calculée à la fin du dernier sprint)

##### Dailys SCRUM

Le Dailys Scrum a lieu **tous les jours**, il s'agit d'une réunion d'une durée moyenne de **15 minutes** qui permet à l'équipe de se remettre en phase avec les objectifs du Sprint en cours.

Les q**uestions indicatives à se poser** pour les membres de l'équipe afin d'établir s'ils sont en phase avec les objectifs sont :

* Qu'est-ce que j'ai fait hier ?
* Qu'est-ce que je dois faire aujourd’hui ?
* Quels obstacles j'ai et je pourrai rencontrer ?

##### Sprint Review

La Sprint Review permet à l'équipe de présenter **le travail réalisé** lors du Sprint aux différentes parties prenantes, lors de cette réunion, le Product Owner peut décider de livrer ou non le travail réalisé.

Pour un Sprint d'une durée d'**1 semaine envisagez 1 heure** de Sprint Review

##### Sprint Retrospective

Cette réunion permet aux membres de l'équipe de **revenir sur le déroulement du Sprint**, ce qui a fonctionné et ce qui n'a pas fonctionné, l'idée de cette réunion est de **trouver des pistes d'améliorations** pour les prochains Sprints de l'équipe, et non **pas de mettre en lumière les échecs**.

### **Définition des rôles SCRUM au sein de l’équipe**

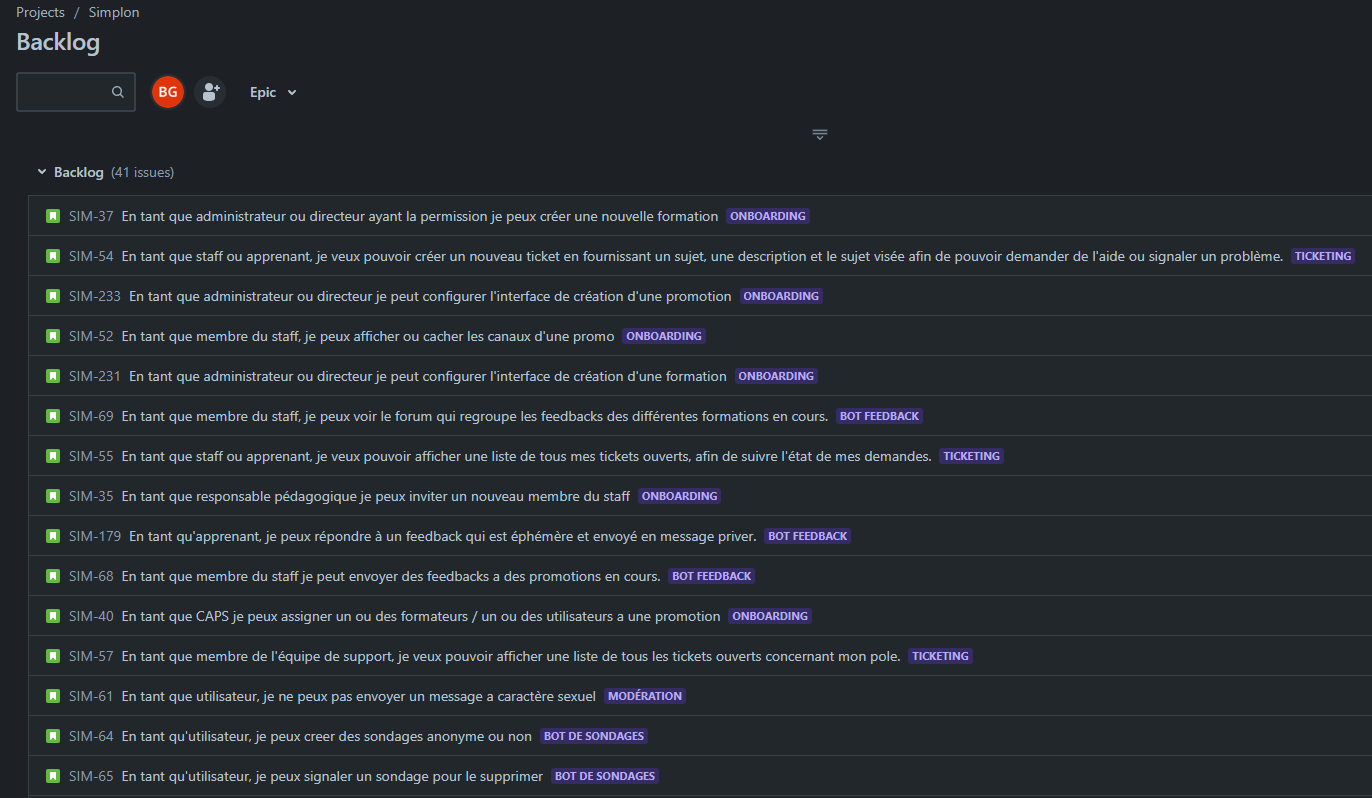
**GAMACHE Benjamin** occupe le rôle de **Product Owner**, qui consiste notamment à prendre en charge les interactions entre l'équipe en charge du projet et le client.

**BOUREZ Bastien** occupe le rôle de **Scrum Master**, qui consiste à diriger les membres de l'équipe en charge du projet et à assigner des tâches à chacun.

**PHILIPPE Nelson** et **LEROY Cédric** occupent tous deux le rôle de **Développeur**, qui consiste à développer les différentes solutions élaborées par tous les membres de l'équipe en charge du projet et validées par le client.

### **Jira**

Pour nous **organiser** sur ce projet, mon équipe et moi-même avons choisi **Jira**, qui est une plateforme permettant d'organiser les différentes tâches d'un projet en les découpant en **Epic**, puis en **User Story**, et enfin en tâches.



*Le Product Backlog du projet sous Jira*

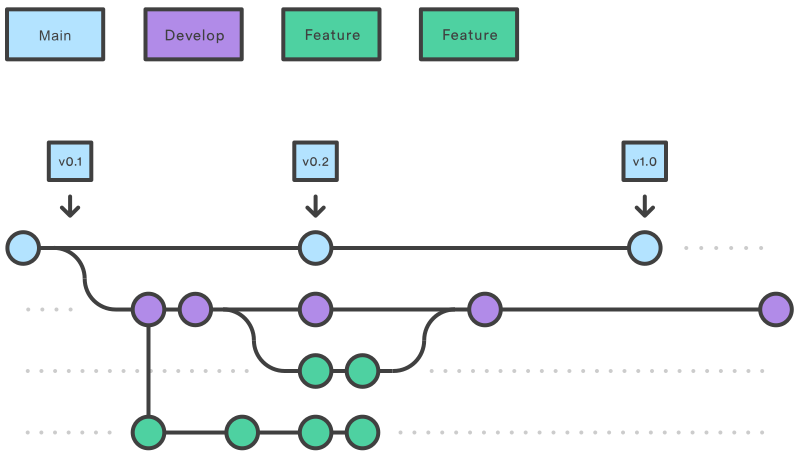
### **Git**

Git est un outil permettant de **versionner son code**. Avec Git, il est possible d'écrire du code et de le séparer sur des **branches**, de revenir à des versions antérieures du code. Git permet aussi d'envoyer son code sur des **plateformes en ligne** comme **BitBucket**, **GitLab**, **GitHub**, ou de l'envoyer sur des plateformes de **self-hosting** pour héberger du code en ligne.

Git est **essentie**l à chaque équipe de développement afin de **collaborer** sur un projet, chaque membre d'une équipe de développement utilise Git afin d'apporter des modifications au code d'un projet.

### **GitFlow**

GitFlow est un **workflow standardisé**, il permet à tous les membres d'une équipe d'adopter le même workflow.



Avec GitFlow, il est facile et rapide de définir un Workflow pour une équipe de développement, c'est pourquoi notre équipe s'est tournée vers GitFlow pour **collaborer**

### **GitHub**

GitHub est une plateforme en ligne qui sert d'**outil collaboratif de versionning de code**. Elle permet d'**héberger du code** en ligne et de retrouver différentes versions du code.

GitHub permet donc de **partager** son code avec une **équipe restreinte** (comme c'est le cas ici) ou de le partager de **façon publique**. Nous avons donc utilisé GitHub afin de travailler en équipe sur ce projet.

#### Architecture du projet

#### Mono-Repo VS Multi Repo

Pour choisir la meilleure façon d'**organiser le projet** sur GitHub, nous nous sommes posé des questions sur la **solution optimale** entre le **Mono Repository** et le **Multi Repository.**

Le **Mono Repository** :

* **Centralise** toute la **conception** de l'application.
* **Centralise** tout le code de l'application.
* Mise en place de pipelines de déploiement (**CI/CD**) **plus complexes**.
* **Centralisation** de toutes les **dépendances** de l'application, qu'elles soient utiles à un module ou non.

Le **Multi Repository** :

* **Décentralise** la conception de l'application.
* **Décentralise** les sources de l'application.
* Mise en place de pipelines de déploiement (CI/CD) **plus légères**.
* **Répartition des dépendances** de chaque partie de l'application dans des dépôts séparés.

Nous avons adopté une **approche mixte** en hébergeant notre application avec **3 dépôts** :

* **1 dépôt pour l'API et la base de données**.
* **1 dépôt** pour l'application Web (**front-end**).
* **1 dépôt pour tous les bots** constituant l'application.

#### GitFlow Vs Fork

Bien qu'il était initialement prévu de travailler en forkant le dépôt du projet, notre équipe est revenue sur sa décision quant à la **méthodologie à adopter** pour travailler avec GitHub, car elle ne convenait pas à tous les membres de l'équipe.

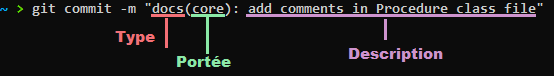
De plus, avec une **petite équipe de 4 membres**, le **fork n'est pas une obligation**. Nous pouvons tout à fait utiliser GitFlow en raison de la taille de l'équipe, créer des branches et les envoyer sur le dépôt distant.

**Le fork est plus apprécié dans les projets Open-Source** où la taille des équipes de développement n'est pas fixe, et dans lesquels les contributions sont émises par des collaborateurs qui ne sont pas à l'initiative du projet. Le fork permet une meilleure organisation dans les équipes de grande taille.

#### Convention de formatage des commits

Pour réaliser nos commits, notre équipe a adopté la convention "**Angular Style Commits**" qui permet de définir :

* Un type.
* Une portée (optionnelle).
* Une description.
* Un corps de commit (optionnel).
* Un footer de commit (optionnel).



# **Annexe**

[1] : Algorigramme concernant la création de promo

