
MGL7460
Réalisation & Maintenance logiciels

Projet Individuel

Nasreddine Salem

TABLE DES MATIERES

Introduction.....	3
TP1- Réalisation & maintenance logiciels.....	3
Sommaire d'analyse	3
Dimension - ÉQUIPE de développement	3
Git-Statistica	4
Implication des contributeurs	4
Architecture logicielle.....	9
Dimension - test	10
DIMENSION – DÉPOIEMENT & LIVRAISON CONTINUE	10

INTRODUCTION

TP1- RÉALISATION & MAINTENANCE LOGICIELS

Le cours sur la *réalisation et maintenance de logiciels* se conclut avec ce travail pratique. Lors des dernières cours, les concepts tel que la gestion de version et les tests unitaire et intégrer lors de la réalisation d'un logiciel et leur importance ont été présentés. Ces pratiques et concepts d'architecture qui présentent des solutions aux problèmes les plus communs complètent la formation théorique.

Ce devoir est une application de ces concepts, mon choix est porté sur le logiciel open source Spring Initializer. C'est un outil développer par l'équipe de Spring boot afin de facilité la génération d'une première structure de projet basé sur du Spring boot, un générateur de projet utiliser par des millions de développeurs autours du monde.

Ce rapport présente les différentes dimensions d'analyse suivi, afin de mieux comprendre la phase de réalisation de ce logiciel et sa maintenance. Ces derniers fournissent les détails nécessaires de la contribution des développeurs via l'analyse de Git. Ce faisant, il sera possible de démontrer la bonne forme de l'architecture de logiciel et sa maintenabilité ou pas.

SOMMAIRE D'ANALYSE

Je travaille sur l'analyse et l'étude d'un logiciel en libre-service, est un générateur de code pour les projets de types Spring-Boot.

L'analyse sera dirigée principalement par les différentes dimensions suiavante :

- Dimension - équipe de développement
- Dimension - architecture logicielle
- Dimension – tests
- Dimension – Déploiement et livraison

Dans cette synthèse, je vais répondre minimalement à une question par dimension, afin de couvrir le plus d'aspect de ce produit, car un tel produit qui est à l'origine de plusieurs grandes réalisations dans le monde de logiciel mérite d'être observé et analysé sur plusieurs dimensions.

DIMENSION - ÉQUIPE DE DÉVELOPPEMENT

Dans cette phase d'analyse, nous intéresserons principalement aux contributeurs actives, et leurs contributions, par l'analyse des toutes les actions enregistrer au niveau du gestionnaire de version GIT.

PRÉREQUIS :

Avant d'analyser les données, j'ai mis en place un utilitaire batch, qui permet l'extraction des données à partir du log Git, puis les importer dans une base de données de type SQLITE, pour pouvoir exploiter plus aisément les données.

IL faudra avoir ces outils installés préalablement sur le poste de travail avant de commencer l'exploitation de cet utilitaire; voici les prérequis :

- Sqlite version 3
- Git batch ou équivalent

GIT-STATISTICA

```
*****
MENU - GIT STATISTICA
*****
1. Afficher les Contributeurs
2. Afficher les statistiques des merges
3. Afficher les changement dans repo Git les 30 dernier jours
4. Afficher les commit par mois
5. Afficher le nombre de fichier (par default :*.java)
6. Export detail du logs vers sqlite table
7. Exit
Enter choice [ 1 - 7 ] |
```

- Source : https://github.com/nsalem-it/MGL_7460
- Version : 1.0.0

INSTALLATION:

Une fois le dépôt git du projet et cloner sur le post de développement, à partir de la racine de projet, lancez le fichier batch suivant :

```
253LVL /mnt/c/Uqam/MGL-7460/projet/initializr (master) $ ../MGL_7460/gitStatistica/gitStatistica.sh
```

L'outil propose des fonctions qui permet l'affichage de certaines statistiques de contribution dans le projet, mais pour une meilleure analyse des données, l'option d'exportation vers une base de données SQLITE est proposée pour des besoins de forage de données.

La base de données est composée de quatre tables :

- ComitParAnneeParAuteur : Total de commit par auteur regrouper par année.
- ContributionParFichier : Nom des fichiers avec leur contributeur.
- DetailedGitStatsTbl : Pour chaque contributeur on retrouve le nombre de commit et de ligne insérer et supprimer.
- ImplicationTbl : Pour chaque contributeur on calcule son taux d'implication.

IMPLICATION DES CONTRIBUTEURS

Qui sont les développeurs principaux du projet ? pour répondre à cette question, nous allons utiliser un script qui calcule le taux d'implication des développeurs dans ce projet selon la formule suivante :

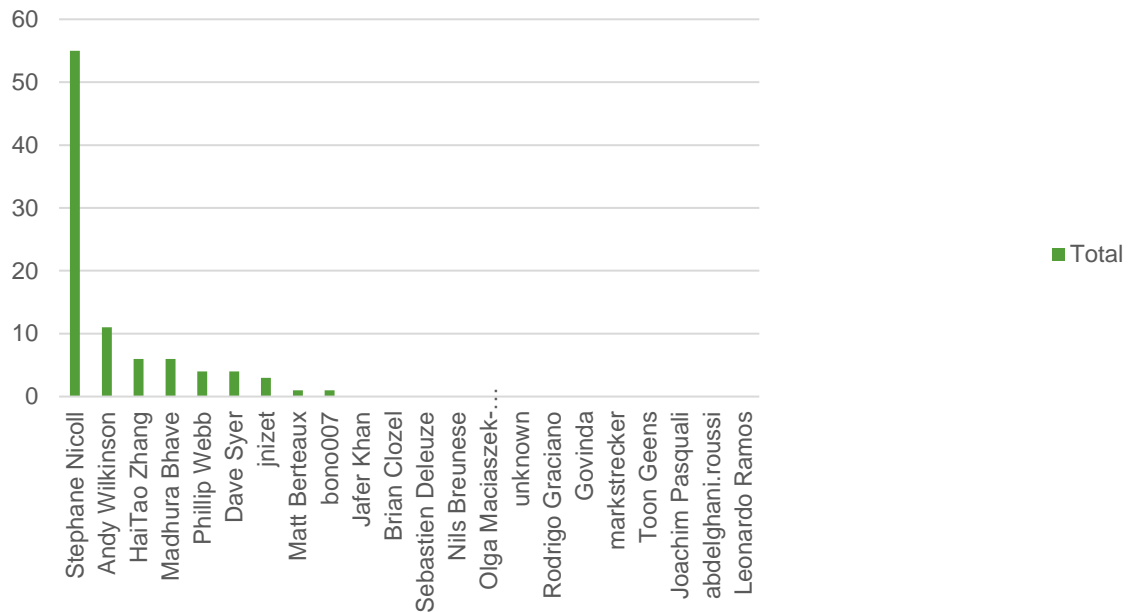
$$\text{Implication (dev)} = \frac{|C|}{|T|} \times 100$$

|T| = Nombre de fichier complet

|C| = Nombre de fichier sur lequel « dev » à contribuer

Après analyse de résultat j'obtiens ce graphe :

Implication par contributeur



Contributeur	Implication	Emplacement	Rôle	Compagnie
Stéphane Nicoll	55%	Liège, Belgium	Committer	Spring Team
Andy Wilkinson	11%	Southampton, UK	Développeur	Spring Team
HaiTao Zhang	6%	s-o	s-o	s-o
Madhura Bhawe	6%	San Francisco, CA	Développeur	Spring Team
Phillip Webb	4%	s-o	s-o	s-o
Dave Syer	4%	London	Senior Consulting Engineer	Spring Team
jnizet	3%	s-o	s-o	s-o

- On distingue clairement un groupe de contributeurs qui sont les plus actifs dans ce projet, puis je constate clairement que **Mr. Stéphane Nicoll** est le contributeur principal de ce projet, d'ailleurs avec une petite recherche sur le web, je retrouve facilement cet auteur comme développeur principal et initiateur de ce projet, dans l'équipe de développement de Spring.

L'autre point intéressant est que les contributeurs les plus importants font partie de la même compagnie, c'est-à-dire Spring-Team, et que sont géographiquement séparés, donc on se demande dans ce genre de situation, comment la collaboration ne soit pas affectée, par la distance ou la langue?



Stéphane a 20 ans d'expérience en génie logiciel avec une forte concentration sur le développement d'API sur la JVM. Après avoir passé plus d'une décennie à développer des applications d'entreprise Java à grande échelle dans les secteurs géospatial, financier ou logistique, il rejoint l'équipe de développement de Spring Framework en 2014. Stéphane travaille maintenant sur Spring Boot et est à la tête de start.spring.io, un service qui aide des millions d'utilisateurs à démarrer leurs prochaines applications

Réf : <https://spring.io/team/snicoll>

Ce contributeur est toujours actif au niveau de ce projet, et continue à le maintenir, mais qui sont-ils les autres contributeurs et quel rôle jouent-ils dans le maintien de ce projet open-source?

ANALYSE DES COMMITS

Je commence par analyser les commits totaux effectués pendant chaque année, et cela depuis la première année à ce jour :

Somme de nombreDeCommit		Étiquettes de colonnes								Total
Étiquettes de lignes	2013	2014	2015	2016	2017	2018	2019	2020		général
Stephane Nicoll		49	131	196	248	217	296	6		1143
Dave Syer	23	49	30	21	25	7				155
Andy Wilkinson			6	1	2	17	28			54
Madhura Bhawe						9	40			49
Brian Clozel		1	9	7	4	1	5			27
Claus Ibsen				6	10	6				22
Sebastien Deleuze		2		5	9	1	1			18
Phillip Webb						11	1			12
Spencer Gibb				3	4	3				10
Ryan Baxter					4	6				10
Spring Buildmaster					3	2	4			9
Craig Walls				2	5	2				9
Josh Long		1		4	2	1				8
HaiTao Zhang							8			8
Roy Clarkson				4		3				7
Stéphane Nicoll					5	1				6
weiping						5				5
Johannes Edmeier						5				5
jnizet							5			5
Total général	23	102	176	249	321	297	388	6		1562

On remarque un changement dans le classement des contributeurs, Mr. Dave Syer prend la deuxième place comme committer dans cette équipe, et sa contribution est assez stable chaque année jusqu'à 2019 où il a arrêté de contribuer, tandis que Mr. Andy Wilkinson et Mr. Madhura Bhawe, leurs courbes de contribution ont subitement connu une hausse progressive depuis 2018, on peut imaginer qu'un événement au sein de l'équipe a eu lieu ou un changement majeur dans l'architecture de projet a évolué, comme les changements de rôle ou d'affectation.

Nous allons creuser un peu plus pour comprendre ce changement. On passe à l'analyse de type de fichier sur lesquels a travaillé ces trois contributeurs : Dave Syer, Andy Wilkinson et Madhura Brave :

[illegible]

Un contrôle de version est un livre d'histoire qui raconte le vécu d'une équipe, et ces interactions, ainsi que les changements dans son vécu, car le but qui les réunissent avant tous c'est la contribution dans le même projet.

Afin de confirmer cette histoire, nous allons continuer notre investigation dans l'analyses de données Git, les plus détailler tel que les nombre de ligne insérer et les dates du premier et dernier comite pour chaque contributeur.

ANALYSE DES INSERTION ET SUPPRESSION DES LIGNES DE CODE

Dans ce graphe la chronologie dans le temps des différentes interventions des insertions et suppression de ligne dans le code, nous démontre que Mr.Dave a participé activement pendant la période entre 2013 et 2018, tandis que Mr Madhura a repris la modification d'une grande partie des fichiers sur les quelles à contribuer Dave, alors que Andy a beaucoup plus ajouter des lignes dans le codes, ce qui le qualifie de nouveau mainteneur dans ce projet.

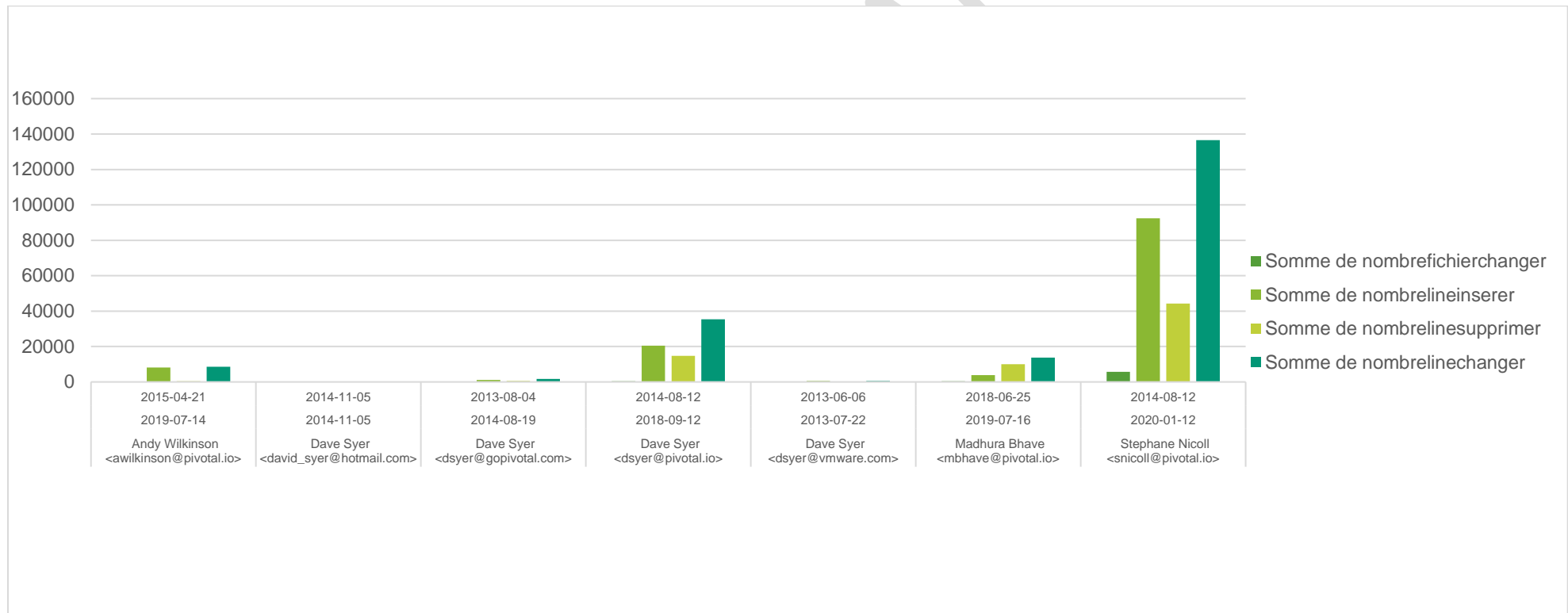


Figure : Implication détailler

Les autres points qui ont attiré mon attention, est le changement de statut de l'entreprise Spring, car on voit clairement que entre 2013 et 2014 c'était goPivotal puis a partir de 2014 elle deviens Pivotal. Aussi la parcours de Mr.Dave qui passe de vmware a pivotal, avec probablement une phase de consulting avant de rejoindre définitivement l'équipe spring Io.

LES MERGE

Dans cette vue on remarque clairement les mainteneurs les plus importants dans ce projet, ce qui confirme notre théorie sur le fait que Andy Wilkinson prend un rôle important dans la continuité de l'évolution du produit, et cela par l'occupation de la deuxième place après Mr. Stéphane.

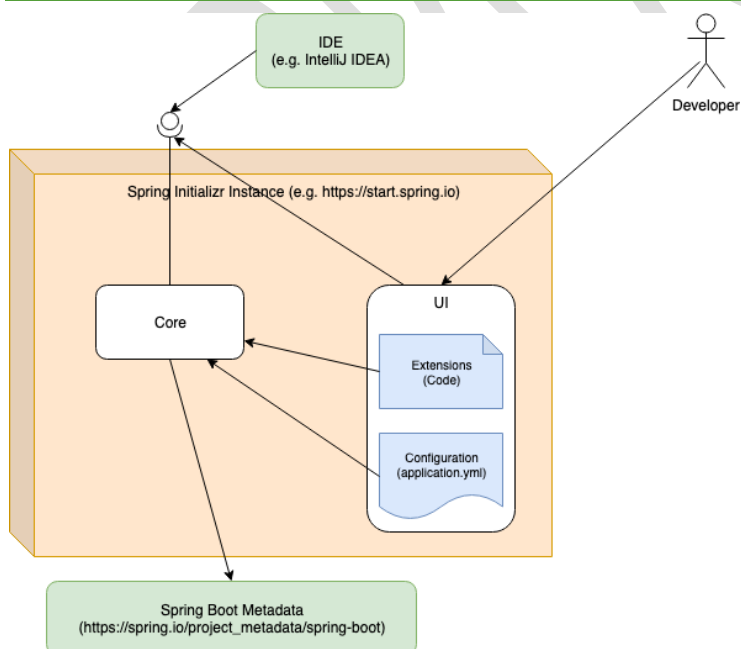
```
Enter choice [ 1 - 7 ] 2
% of Total Merges      Author  # of Merges  % of Commits
      89.52      Stephane Nicoll      188      16.44
       6.19      Andy Wilkinson       13      24.07
       1.90      Stéphane Nicoll        4      66.66
       1.42      Madhura Bhav        3       6.12
       0.95      Dave Syer             2       1.29
Press [Enter] key to continue...|
```

ARCHITECTURE LOGICIELLE

Référence : <https://docs.spring.io/initializr/docs/current-SNAPSHOT/reference/html/>

Dans cette partie du document, je vais explorer la dimension architecturale de ce produit, je vais principalement m'intéresser aux différents modules composant ce produit, et les relations entre ces modules, mon analyse portera aussi sur la contribution des développeurs dans ces modules.

VUE GLOBALE D'ARCHITECTURE



Référence : <https://medium.com/digitalfrontiers/how-to-customize-the-spring-initializr-2439ecabb069>

Nous avons plusieurs modules qui compose cette application :

- `initializr-actuator`: Module optionnel pour fournir des informations et des statistiques supplémentaires sur la génération de projets
- `initializr-bom`: Fournit une nomenclature pour une gestion plus facile des dépendances dans votre projet.
- `initializr-docs`: Documentation.
- `initializr-generator`: Core principal du projet.
- `initializr-generator-spring`: Module optionnel définissant les conventions pour un projet Spring Boot typique. Peut être réutilisé ou remplacé par vos propres conventions.
- `initializr-generator-test`: Tester la composition des modules pour la génération de projets.
- `initializr-metadata`: Infrastructure de métadonnées pour divers aspects du projet.
- `initializr-service-sample`: Présente une instance personnalisée de base.
- `initializr-version-resolver`: Module optionnel pour extraire les numéros de version d'un POM.
- `initializr-web`: Points de terminaison Web pour les clients tiers.

À compléter bientôt :

Prochaine étape c'est d'analyser chaque module pour savoir les principaux contributeurs et la stabilité des ajouts et modification du code et son évolution dans les versions précédentes.

DIMENSION - TEST

Dans cette dimension, mon analyse s'effectuera principalement sur les tests effectuer dans les différents modules, tel que la qualité des tests, pour cela je vais m'appuyer sur trois outils :

- SonarQube : pour l'analyse statique.
- CodeMR : un plugin qui donne des statistiques complémentaires à sonarQube.
- Pitest : pour analyser la qualité des tests.

DIMENSION – DÉPOIEMENT & LIVRAISON CONTINUE

Dans cette dimension, je vais m'intéresser principalement sur le pipeline mis en place pour automatiser la mise en production des nouvelles versions, plus spécifiquement le changement dans le module Web qui est la porte principale de génération de million de projet dans le monde.

Je vais regarder les outils et la mis en place du CI. Regarder plus en profondeur les différents stages existant dans le pipeline et la raison de leur existence.

BROUILLON