**Contexte de la mission**

Il existe une application nommée OpenFramework (OF) qui permet de collecter des flux dans le parc d’application d’une entreprise. Ce grand nombre de données sera stocké, puis analysé pour afficher, dans une interface web, des graphiques et des statistiques.

La version 3 de cette application semble avoir atteint ses limites au vu du nombre de données qui augmente de plus en plus. C’est pourquoi ma mission globale de stage était de participer à l’élaboration de la version 4 d’OF, qui sera doté de nouvelles technologies pour pallier à ce manque de performance.

Cette nouvelle version est découpée en 2 projets, un premier en Java et utilisant le Framework Spring Data, qui va collecter les flux venant d’un ESB[[1]](#footnote-1), et les insérer dans le moteur de recherche, ElasticSearch. Le second projet est une application web utilisant AngularJS, qui va chercher les données dans ElasticSearch, pour les afficher dans l’interface web

**Semaine 1**

28/09 – 02/10

* Correction des dates, et du typeahead (ouverture, et non fermeture des choix)
* Ajout d'une colonne dans le tableau de résultats pour indiquer les statuts (FINISHED, STARTED, …) avec des icones.

**Semaine 2**

12/10 – 16/10

* Filtres des recherches pour renvoyer toutes les instances qui n'ont pas d'instance parente, pour limiter les résultats, et avoir une meilleure hiérarchie.
* Changement des noms Utils -> Util
* OF-core : jeux de tests pour mon appli, envoie de message dans la queue
* Tri des résultats par status (FINISHED, STARTED, ERROR, …)
  + Problème : tri sur la page en court, et pas sur tous les résultats du tableau !
  + Résolution : change gridApi, pour spécifier l'action à faire, lorsque l'on clique sur une colonne pour la trier. Je relance la requête pour trier directement tout les éléments et ça rempli le tableau seulement après.
* Ajout du niveau des priorités pour les statuts, après la collecte des messages, pour l'envoie dans ElasticSearch
* Tri des statuts par leur priorité : petit problème: le ng-grid ne rafraichi pas les directives de statuts. Résolution : watcher sur la cellule de la colonne, et si elle change, on met a jour l'icone, la couleur et le tooltip.
* Commit change
* Début d'association d'of-core et of-front. Je dois lancer ElasticSearch et of core (collecte) depuis of front, des le lancement du serveur tomcat.

**Semaine 3**

26/10 – 30/10

* Association des deux projets, avec leurs dépendances maven. Test pour voir si tout fonctionne bien. OK.
* Lancement d'ElasticSearch au démarrage de tomcat. Un ElasticSearch embarqué. Ajout d'un plugin java, et modification pour qu'il s'adapte a nos besoins….. OK
  + Problème : je ne sais pas ou est instancié ES.
    - Résolution: Enfaite il se trouve sur l'"url-pattern" de la servlet que le lance, je devais ajouter à "/elasticsearch", "/\*" a la fin, pour pouvoir explorer ElasticSearch, et pouvoir faire des requête dessus.
  + Problème : accès interdit, via CrsfFilters, pour les requêtes de type POST,
    - Idée de résolution: création du CrsfFilters perso pour qu'il accepte les requêtes POST. Definition du csrf filter dans spring security avec <security:csrf request-matcher-ref="maclass"/>
    - Fonctionne -> ajout des données déjà créer sur le serveur, pour les intégrer dans le dossier spécifié
* Résolution des erreurs au démarrage de tomcat. Ex : log4j qui n'a pas accès au fichier de log
* Améliorer l'outil de versionning de documents. 2 logiques de versionning proposé, à moi de choisir l'une d'elle.
  + Logique 1 : Comparer le nouveau document (avant insertion), avec le document actif. Puis ajouter dans la base de versionning, que les champs qui diffère entre les 2. Et insérer le nouveau document comme actif.
  + Logique 2 : Insérer dans la base de versionning le document actif au complet, puis mettre le nouveau document comme actif. Et faire tournée une routine qui va comparer les champs et supprimer ceux qui sont identiques.
  + Réflexion -> logique 2, compliqué et surcharge de boucle pour comparé dans la base de versionning ce qui a changer etc.
  + J'ai choisi la logique 1, après avoir trouvé une API java qui compare 2 objets, et renvoie les différences entre les 2, ce qui permet de savoir ce qui a réellement changer, et nous permettra d'ajouter directement dans la base de versionning que les champs qui ont été modifié.
  + Test de l'api java Javers qui compare 2 objet et renvoie les champs qui diffèrent.
  + Implémentation de Javers dans le projet
    - Pas beaucoup de doc, difficile d'accéder aux propriétés et à leur valeur (celle qui ont été modifié)

**Semaine 4**

09/11 – 13/11

* TODO: (BOA) modifier l'interface Kibana, pour ajouter des colonnes au tableau, et donc modifier la requête qui renvoie les résultats.
* DONE: ajouter les colonnes c'est bon, il manque de modifier la requête sur ElasticSearch pour ajouter les données que je veux pour les mettre dans mes nouvelles colonnes.
* IN PROGRESS: créer nouvelle requête pour avoir le compte d'erreur uniquement
* DONE: requête trouver, et insérer.
* IN PROGRESS: insérer les données dans le tableau de résultats
* DONE: insertion des données dans le tableau
* TODO: lister les applications sans que ce soit possible de les filtrer
* Je refais la requête pour ajouter les erreurs et succès, elle s'exécute juste avant d'insérer les données dans le tableau.
* Pas facile, beaucoup d'erreur de promise (asynchronisme), avec les \_.each. J'ai tout repris, la logique c'est : Kibana charge et compte les noms d'applications (nombre max de 10), et seulement après qu'il ait fini, je refais une requête, qui va chercher le nombre d'erreur et le nombre de succès de cette application.
* Kibana calcul une ligne supplémentaire "Other values", avec le nombre restant de données (étant donnée qu'il y en a plus de 10), cependant sur mes colonnes que j'ai ajouter, rien n'était indiqué, donc j'ai créer une nouvelle requête qui s'exécute avant le traitement des données et qui me renvoie le nombre maximum d'erreur et de succès, et puis au moment du traitement des données, des qu'il insert un nombre d'erreur (ou de succès), il incrémente une variable de cette somme, pour qu'a la fin du traitement on ai le total d'erreur que l'on a afficher (et de succès), et ensuite j'insert a coté de "Other values" le nombre total d'erreur – la somme trouvé pour avoir le restant.
* OBJECTIF semaine prochaine : Faire un tableau FIXE avec que les noms d'applications, un bouton d'action qui ajoute des filtres pour les autres graphes, MAIS lui ne change PAS !
* Idée : créer un style (ex : tablefixe), avec une requête spécifique à intégrer dans le code (voir panel/terms/module.js et modifier module.html pour faire un tableau spécifique avec le bouton etc. Cool ☺

**Semaine 5**

23/11 – 27/11

* Je dois créer un "widget" sur Kibana, qui permet d'ajouter des filtres précis sur Kibana 3.
* Ajout d'un module, pour l'instant le même que celui qui gère les filtres, pour analyser les fonctions d'ajout, pour ensuite réutiliser certaines fonctions, ou les modifier un peu pour les adapter.
* Fais. Pour le moment, on a 4 champs (appli source, appli cible, médiation et domaine), on génère la liste brute de toute les appli, etc… puis l'utilisateur choisir parmi la liste pour ajouter au filtres de Kibana.
* Rendre générique, quelques soit le filtre.
* OK
* Maintenant, je dois ajouter une ligne de constante sur un histogramme.
* Je cherche comment ajouter une barre sur l'histogramme, sans aucun critère ni rien. Voir si c'est faisable déjà !
* Idée:
  + Paramétrer le graphe pour ajouter une autre valeur (avoir 2 courbes) problème: pour le moment la variable SLA qui est fixe pour chaque données, ne e trouve pas dans les donnée, dans ElasticSearch, donc peu pas test.
  + Intégrer en dur la valeur du SLA, après que Kibana est fait sa requête sur ElasticSearch, problème: je ne sais pas comment ajouter un point avec cette valeur sur le graphe.
  + Monter l'axe des X, pour qu'en gros les valeurs positivent soit supérieur au SLA, et les valeurs négatives soit inférieur au SLA. J'ai réussi à monter l'axe X, en indiquant que la valeur minimum est de -3 (par exemple), je pense qu'il faut ensuite enlever la valeur du SLA a toute les données. ANNULE...
* Apparemment c'est chaud patate…
* Je suis partie sur l'idée d'intégrer la valeur en dur en créant une deuxième courbe avec des valeurs fixes. D'abord en dur, et après paramétrable en fonction d'une médiation choisi.
* Pour ça, lorsque Kibana traite ses données pour les transformer en graphe, je créer une deuxième courbe qui aura les même paramètres, mais les données seront différentes, pour chaque temps enregistrer (axe X), on fixe une valeur (ici 0.5 par exemple, mais ce sera paramétrable plus tard, valeur du SLA)
  + Problème : entre 2 enregistrements de valeurs, la courbe retourne à 0, et on observe donc des "trous" dans la courbe
  + Résolution: modifier le paramètre fill\_style de "null" à "no". Fill c'est la liaison entre les points, et le no permet de ne pas remettre la courbe à 0 quand il n'a pas de valeur
* Rendre plus générique cette fonctionnalité.
  + checkbox dans les options du graphe, qui servira de test dans le code, puis input text pour entrer la valeur en dur pour le moment pour un graphe précis. OK
  + Lorsque l'on filtre sur le champ NOM\_MEDIATION, on ajoute automatiquement au graphe avec la chexbox coché, la valeur du SLA de cette médiation.
* Reprise d'OFv4, petite correction, pour présentation devant l'équipe.
* Correction de la fonction remove de Kibana, reprise à 0 des ids, décalage des ids.
* Correction d'asynchronisme get business fields dans la liste du côté résultats.

**Semaine 6**

07/12 – 11/12

* Nouveau widget : faire des quartiles sur le temps de réponses.
* Je dois utiliser le graphe existant, histogram, abscisse : temps, ordonnée : nb instances, et modifier le tooltip pour afficher les quartiles.
* Problème avec le http.post, il ne s'exécute pas, ou pas au bon moment…
* Faire un graphe avec abscisse : instance id, ordonnée : liste des statuts.
* Pas facile...
* Test : Kibana 4

**Semaine 7**

21/12 – 24/12

* Aie aie aie …. Les indexes dans ElasticSearch où se trouvent les configurations, les Dashboard de Kibana, ont été malencontreusement supprimé, lors d'un packaging…
* EA -> fixé objectifs pour année prochaine.
* Elaboration d'un tableau de tache avec temps estimé, % etc.
* Problème avec les counts, surement à cause du nouveau modèle de données. Les requêtes faites par Kibana n’obtiennent pas les mêmes résultats lorsque je fais une requête similaire. Je ne sais pas qui de moi ou Kibana a les bon résultats. Je cherche beaucoup sur ce point !

**Semaine 8**

28/12 – 29/12

* Info : cette semaine je travaille uniquement lundi et mardi, j'ai pris 2 jours de congé (30–31)
* Remise en forme de Kibana sur le nouveau type plus quelques petite optimisation pour qu'il soit plus stable (ex: filtre type : value : boaType9, auto grâce au fichier config.js)
* Nouvel tache : les liens des applis, automatiser les liens, temps estimé 2 jours, et il en faudra peu être 3, mais ce n'est pas moi qui vais terminer cette tache parce qu’ils vont livrer la toute première version pendant que je serais en cours.

**Semaine 9**

18/01 – 22/01

* Retour sur OFv4
* Il y a eu une migration de SVN vers Git, j'ai donc importé le "nouveau" projet avec git.
* Proposition d'utiliser IntelliJ, suite à de nombreux problèmes rencontrés sur eclipse (config, etc....)
* Téléchargement d’IntelliJ Idea, avec la licence étudiante pour POC
* Configuration de tomcat, maven …
* Configuration du plugin LiveEdit + extension chrome JetBains IDE Support, qui permet de faire du live coding, ce qui me permet de gagner beaucoup de temps de développement, parce que je n'aurai plus besoin de redémarrer à chaque changement mon serveur tomcat.
* Repackaging du projet toolbox-angular -> séparé le fichier d'import d’ittbui et ittb utils
* Update des versions de
  + Angular (+ angular-aria + angular-animate) : 1.3.15 -> 1.5.0-rc1
  + Angular-Material : 0.10.1 -> 1.0.0
  + Ui-grid : 3.0.1 -> 3.1.0
  + Angular-translate : 2.7.1 - > 2.8.1
  + jQuery : 2.1.4 -> 2.2.0
  + ui-bootstrap : 0.13.0 -> 1.1.0
  + Momentjs : 2.10.6 -> 2.11.1
* Reprendre requireJS, pour faire quelques chose de plus propres pour normaliser, et insérer les fichiers css, pour éviter de surcharger la page html avec tout les imports

**Semaine 10**

01/02 – 05/02

* Réorganisation des imports via RequireJS, dans une configuration Framework (avec projet tooblox-angular) qu'utilise le projet OF pour son interface graphique.
* Séparation du Framework, et de l'application (js, imports, css)
* Réparation du champ autocomplete (problème de nom de variable)
* Env. de test parce que trop chiant de recharger tomcat tout le temps
* Optimisation du datepicker ittb ;)
* Navbar passé de tooblox à of + optimisation

**Semaine 11**

15/02 – 19/02

* Séparation du autocomplete ittb et of, tentation de généralisation…
* problème of démarrage…………

**Semaine 12**

29/02 – 04/03

* modification d’of-core. Ajout du businessFields auto dans ES
* touch touch java☺. Intégration des businessField dans ElasticSearch, ajout d'un businessfieldrepository qui hérite d'une abstractRepository etc...
* Ajout du businessField dans ElasticSearch en java
* Création d'une classe de test, instance descriptionTest. Qui permet d'ajouter dans ElasticSearch une instance description qui va bien, et tester si elle est bien dans ES
* Reprise de l'IHM, sur les businessFields

**Semaine 13**

14/02 – 18/03

* Retour sur IHM, évolution, refresh auto de la requête OK
* Test de performance sur le dépilage de la queue. Pas tip top, cependant l'empilage non plus …
* Corriger le filtre du navbar sur les descriptionType dans l'autocomplete. OK
* MAJ Redmine + Git commit

**Semaine 14**

29/03 – 01/04

* Fil d'Ariane, Redmine 3544, anomalie. Chiffrage 2 j.
* Migration environnement sur nouvelle machine.
* POC intégrer Kibana 4 dans le projet, et qu’il se base sur le serveur Tomcat déjà lancé, plutôt que sur son propre serveur Node. Chiffrage 3 j.
  + J’ai trouvé une version statique de Kibana 4. A tester !
  + Kibana ajouté dans of-front.
  + Configuration ok.
  + Problème avec les url pour passer d’of a Kibana.
  + Quelqu’un d’autre s’en charge
* Aide sur le core, problème au démarrage

**Semaine 15**

04/04 – 08/04

* Reprise sur le front, fil d’Ariane. Simplifié, et plus stable.
* Création de la vue du détail d’une instance.
  + Maquettage, qui a été ensuite validé par le restant de l’équipe, et modifié suite au débat.
  + Développement de cette vue. Estimation 2j (fait en 1.5j)
* Création d’un service angular, qui enregistre la recherche fait précédemment, pour que lorsque l’on revient en arrière après avoir vu des détails, ça ne réinitialise pas les champs. (2j)
  + Problème avec autocomplete. Marche plus avec les regex (texte taper a la main)

**Semaine 16**

11/04 – 15/04

* Correction de l’autocomplete avec les textes taper a la main.
* Problème pour ouvrir les choix au premier clic ! réussi
* Continuer service search, pour le retour précédant sauver la search

**Semaine 17**

02/05 – 04/05

* Mise en place d’un menu avec arborescence, comme dans l’explorer Windows.
* Reprise du tableur Excel, avec les taches, amélioration, imitations tableau SCRUM, avec a faire, en cours et terminé. Des % de taches réalisé. Et je devrais envoyer ce tableur toute mes semaines d’entreprises à mon tuteur, et manager, pour qu’ils voient l’avancement du projet

**Semaine 18**

17/05 – 20/05

* Reprise du menu
* Aide d’un stagiaire, sur des compétences Angular
* Menu terminé, en plus de temps que prevu..
* Creation d’une nouvelle page, chercher les référentiels, et les manipuler.
  + Gestion des vue avec ui router. (State) OK
  + Creation de la vue et du controller.

1. ESB : Entreprise Service Bus. [↑](#footnote-ref-1)