

# Cahier des Charges

Sonny Klotz - Jean-Didier Pailleux - Malek Zemni

*Interface de chargement, de contrôle  
et d'analyse statistique des données  
pour la constitution d'un graphe de flux*

14/03/2017

# Table des matières

<b>1</b>	<b>Motivations du projet</b>	<b>1</b>
1.1	But du projet . . . . .	1
1.1.1	Contexte du projet . . . . .	1
1.1.2	Objectif du projet . . . . .	1
1.2	Parties prenantes . . . . .	1
1.2.1	Maître d'ouvrage . . . . .	1
1.2.2	Client . . . . .	1
1.2.3	Autre partie prenante . . . . .	1
1.3	Utilisateurs du produit . . . . .	2
<b>2</b>	<b>Contraintes sur le Projet</b>	<b>2</b>
2.1	Contraintes imposées . . . . .	2
2.1.1	Contraintes sur la conception : . . . . .	2
2.1.2	Environnement de fonctionnement : . . . . .	3
2.1.3	Applications partenaires : . . . . .	4
2.1.4	Temps dont disposent les développeurs du projet : . . . . .	4
2.1.5	Budget du projet : . . . . .	4
2.2	Glossaire et conventions de dénomination . . . . .	4
2.3	Faits et hypothèses déterminants . . . . .	4
2.3.1	Facteurs influençant le produit, mais qui ne sont pas des contraintes imposées sur les exigences : . . . . .	4
2.3.2	Hypothèses que l'équipe fait sur le projet : . . . . .	5
<b>3</b>	<b>Exigences fonctionnelles</b>	<b>5</b>
3.1	Périmètre de l'ouvrage . . . . .	5
3.1.1	Diagramme de contexte : flux élémentaires . . . . .	5
3.1.2	Contexte du travail . . . . .	5
3.2	Périmètre de l'œuvre . . . . .	5
3.2.1	Diagramme de cas d'utilisation . . . . .	5
3.2.2	Description sommaire des cas d'utilisation . . . . .	5
3.3	Exigences fonctionnelles et exigences sur les données . . . . .	6
3.3.1	Exigences fonctionnelles . . . . .	6
3.3.2	Exigences sur les données . . . . .	6

<b>4</b>	<b>Exigences non fonctionnelles</b>	<b>6</b>
4.1	Interface utilisateur du produit . . . . .	6
4.1.1	Exigences d'apparence . . . . .	6
4.1.2	Exigences de style . . . . .	6
4.2	Utilisabilité . . . . .	6
4.3	Exigences de performance . . . . .	7
4.4	Fonctionnement du produit . . . . .	7
4.4.1	Principales fonctionnalités . . . . .	7
4.4.2	Précision et exactitude . . . . .	8
4.5	Maintenabilité du projet . . . . .	8
4.6	Sécurité . . . . .	8
<b>5</b>	<b>Autres aspects du projet</b>	<b>9</b>
5.1	Question ouvertes . . . . .	9
5.2	Solution étagère . . . . .	9
5.3	Problèmes nouveaux . . . . .	9
5.4	Tâche à faire . . . . .	9
5.5	Contrôle de la finalisation . . . . .	9
5.6	Risques liés au projet . . . . .	9
5.7	Estimation des coûts . . . . .	9
5.8	Documentation utilisateur et formation . . . . .	9
5.9	Question mise en attente . . . . .	9
5.10	Idées de solutions . . . . .	9

# 1 Motivations du projet

## 1.1 But du projet

### 1.1.1 Contexte du projet

De nos jours, les masses de données collectées sont de plus en plus importantes. L'objectif principal de cette collecte de données est d'en extraire une valeur ajoutée. Or, ces données à l'état brut sont difficilement exploitables dû à leur volume et à leur complexité.

Notre produit correspond au travail indispensable d'analyse de ces données, afin de faciliter leur exploitation.

### 1.1.2 Objectif du projet

Ce projet a pour but de fournir aux utilisateurs une application qui se chargera d'une part de structurer les données, les analyser et les visualiser, et d'autre part de préparer ces données pour un chargement via des API.

## 1.2 Parties prenantes

### 1.2.1 Maître d'ouvrage

Notre interface de contrôle, de chargement, et d'analyse de données est développée pour l'entreprise **DCbrain**.

Le projet a été lancé en collaboration avec l'**UVSQ**.

### 1.2.2 Client

DCbrain est également l'entreprise qui va bénéficier des paquets finaux après leur développement.

### 1.2.3 Autre partie prenante

Les industriels clients de DCbrain sont des parties prenantes indirectes. Notre travail doit pouvoir être utilisé par DCbrain pour renforcer leur application.

## 1.3 Utilisateurs du produit

En premier lieu, l'application va servir aux membres de DCbrain étant donné que leurs clients industriels (Total, ERDF,...) leur fournissent les fichiers CSV, afin qu'ils puissent appliquer des analyses descriptives sur les données dans le but de repérer des anomalies sur les réseaux de ces derniers. Puis en second lieu, il se pourrait que DCbrain veuille déployer l'application pour ses clients, dans ce cas les membres de ces organismes deviendront des utilisateurs.

On peut supposer l'absence de restrictions d'utilisations, étant donné l'absence d'exigence de la part du maître d'œuvre sur ce sujet.

# 2 Contraintes sur le Projet

## 2.1 Contraintes imposées

### 2.1.1 Contraintes sur la conception :

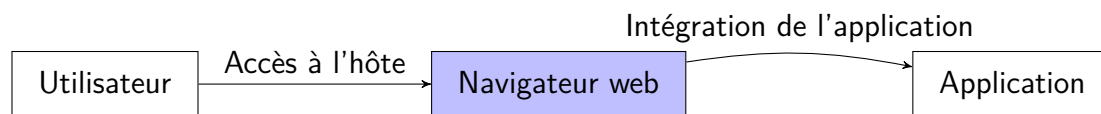
Contrainte	Fiche
1. Le produit doit fournir une application web	<p><b>Description</b> : notre produit sera une application fonctionnant sur un navigateur web, appelée <i><b>applet</b></i>.</p> <p><b>Justification</b> : assure une très grande portabilité et fournit à l'utilisateur une interface interactive.</p> <p><b>Critère de satisfaction</b> : on peut lancer l'application sur un navigateur web.</p>

<p>2. Le produit doit être développé avec un langage de programmation compatible avec l'analyse de données</p>	<p><b>Description :</b> le langage de programmation choisi doit inclure une bibliothèque qui permet d'analyser les données (tâches de data mining et de machine Learning).</p> <p><b>Justification :</b> permettre une analyse de données la plus efficace possible.</p> <p><b>Critère de satisfaction :</b> on peut analyser les données de manière efficace.</p>
<p>3. Le produit doit fournir une API d'analyse de données en sortie</p>	<p><b>Description :</b> l'application doit intégrer des API d'analyse descriptive de données qui pourront être livrées en sortie au client.</p> <p><b>Justification :</b> permettre une réutilisabilité des fonctionnalités majeures du produit.</p> <p><b>Critère de satisfaction :</b> on peut exporter une API d'analyse de données en sortie.</p>

**Choix des langages de programmation :** les contraintes 1 et 2 nous permettent de nous fixer sur le choix du langage : **Java EE**. D'une part, ce langage est orienté pour le développement d'applications web robustes et distribuées, déployées et exécutées sur un serveur d'applications. D'autre part, ce langage qui est basé sur Java est doté de plusieurs modules d'analyse de données, dont l'outil **Weka** par exemple. De plus, l'applet Java doit être intégrée dans une page web pour être exécutée : on va donc avoir recours à des langages de balisage comme **HTML** pour la présentation et **CSS** pour la mise en forme.

### 2.1.2 Environnement de fonctionnement :

Le produit va fournir une application web ou **applet**. L'environnement technologique de ce genre d'application sont les navigateurs web. Ces navigateurs web jouent le rôle d'interface entre l'utilisateur et l'application.



Le produit doit donc être compatible avec tous les navigateurs web bureau (pas de version mobile exigée) prenant en charge les fonctionnalités des dernières versions des langages (**HTML5** et **CSS3**), par exemple **Google Chrome** et **Mozilla Firefox**.

### 2.1.3 Applications partenaires :

Le produit va fournir une API en sortie. Il doit donc prendre en compte de l'environnement d'intégration de cette API, c'est à dire que cette API doit être compatible avec les outils du client, l'entreprise DCbrain.

### 2.1.4 Temps dont disposent les développeurs du projet :

Le produit doit être rendu avant le 26/05/2017.

### 2.1.5 Budget du projet :

La réalisation du produit n'exige pas de ressources financières. Aucun budget n'est donc nécessaire.

## 2.2 Glossaire et conventions de dénomination

**API** : *Application Programming Interface*, constituent les paquets utilisables par les développeurs (intégrés), qu'on va livrer au client en plus de l'application elle-même.

**ADD** : *Analyse Descriptive de Données*, fonctionnalité d'analyse de description statistique des données.

## 2.3 Faits et hypothèses déterminants

### 2.3.1 Facteurs influençant le produit, mais qui ne sont pas des contraintes imposées sur les exigences :

Rien à mettre à priori, mettez des trucs si vous en trouvez

### 2.3.2 Hypothèses que l'équipe fait sur le projet :

rien pour l'instant

## 3 Exigences fonctionnelles

### 3.1 Périmètre de l'ouvrage

#### 3.1.1 Diagramme de contexte : flux élémentaires

Le diagramme ci-dessous définit le contexte du travail : **Diag à insérer ici**

#### 3.1.2 Contexte du travail

- développeurs : **développeur souhaitant apporter de la VA à son produit / rééviter de coder lui-meme blabla ...**
- industriels : **utilisateurs sans connaissances en informatique ni en ADD présumposées => simple blabla...**

### 3.2 Périmètre de l'œuvre

#### 3.2.1 Diagramme de cas d'utilisation

Le diagramme ci-dessous définit le périmètre d'utilisation de notre travail : **Diag à insérer ici**

#### 3.2.2 Description sommaire des cas d'utilisation

L'utilisation de notre travail va se limiter à deux cas, un pour chaque type d'utilisateur :

- développeurs : **il prend les fct, exporte les vals, intègre notre travail dans l'appli qu'il développe blabla ...**
- industriels : **lance l'applet cherche un fichier blabla ...**



## **3.3 Exigences fonctionnelles et exigences sur les données**

### **3.3.1 Exigences fonctionnelles**

Voir organigramme - on liste tout (numero aux exigences) justification et critere de satisfaction

### **3.3.2 Exigences sur les données**

Début de spécification : csv formaté, format d'exportation, vals tmp stockées pour les calculs + cardinalités

## **4 Exigences non fonctionnelles**

### **4.1 Interface utilisateur du produit**

#### **4.1.1 Exigences d'apparence**

Le produit devra adopter une apparence simple, agréable et devra être facile à comprendre dès sa première prise en main. Le maitre d'œuvre a émis le souhait d'avoir de l'Anglais comme langue utilisée pour l'affichage textuel.

#### **4.1.2 Exigences de style**

L'application s'adresse à un public professionnel, alors l'une des exigences concernant le style de l'interface est qu'il devra opter pour un style classique, au goût du jour et professionnel. L'affichage des résultats et autre devra susciter la confiance auprès des utilisateurs.

### **4.2 Utilisabilité**

Le produit devra être simple d'utilisation et facile à comprendre dès sa première prise en main, afin d'éviter une formation concernant la manipulation du produit pour les

futurs utilisateurs. Il aidera également l'utilisateur à analyser une quantité importante de données.

## 4.3 Exigences de performance

Dans le template y a pas cette partie

mais y a "Adéquation du produit avec son environnement" qui découle de la partie "Contraintes/Environnement"

Les exigences de performance du futur produit concerne la latence acceptable que devra avoir l'application. Ici la durée de réponse de l'affichage d'un CSV, ou d'une requête concernant l'analyse de données venant de l'utilisateur ne devra pas être trop longue, le client à suggérer que cette attente devra être de l'ordre de la minute maximum. En ce qui concerne le temps de passage d'une fenêtre à l'autre (Choix de(s) colonne(s) pour l'exécution d'une analyse statistique, affichage de la fenêtre d'erreurs, lors du choix du méthode d'affichage des résultats,...), l'application devra répondre de manière fluide.

## 4.4 Fonctionnement du produit

### 4.4.1 Principales fonctionnalités

Tout d'abord, l'application devra avoir la fonction majeure de posséder une interface afin de permettre le chargement d'un fichier au format CSV. Après avoir sélectionné un fichier, l'application pourra afficher le contenu de ce fichier sous forme de tableau et les erreurs dans le contenu d'une ligne dans une nouvelle fenêtre. Si le fichier contient une quantité importante de données (Supérieur à 1000), l'affichage ne se fera que sur les 1000 premières données. Il n'y aura pas d'importance à déterminer le type de colonne car il y aura un format prédéfini pour le CSV.

Ensuite, il y aura la possibilité de faire une analyse descriptive sur les colonnes du fichier. Avec une analyse qualitative pour obtenir les informations synthétiques tel que la Médiane et les Quantiles mais également les anomalies présente sur certaines valeurs. Puis une analyse quantitative pour obtenir par exemple la distribution des valeurs.

Enfin, l'application aura la fonctionnalité de récupérer les résultats de ces analyses sous forme d'API afin de permet à DCbrain de connecter ces nouvelles données à leur base de données pour insérer de nouvelles informations dans leurs graphe de flux.

#### 4.4.2 Précision et exactitude

Lors de l'analyse des données, les résultats sur la Variance et la Moyenne doivent être calculée de façon très précise car elles vont servir de base dans d'autres calculs statistiques. Les autres valeurs calculées peuvent être les plus précises possible mais ces valeurs vont servir pour l'interprétation afin mieux comprendre ce qui se passe sur le réseau.

### 4.5 Maintenabilité du projet

Le produit doit pouvoir être maintenu par ses utilisateurs finaux ou par des développeurs qui ne sont pas les développeurs d'origines, dans le cas où DCbrain souhaiterait ajouter de nouvelles méthodes d'analyses descriptives ou de nouveaux procédés pour afficher les résultats de manière graphique afin de satisfaire les exigences de leurs clients.

Le produit devra permettre l'insertion d'éventuels API supplémentaires, tel que l'analyse Multidimensionnel qui utilisera l'API de l'analyse Unidimensionnel dans le but de fournir des descriptions statiques plus poussé pour obtenir une meilleur vue d'ensemble sur les données collectées. Mais encore l'insertion de technique d'analyse de Graphe afin d'obtenir des informations sur l'état d'un graphe.

### 4.6 Sécurité

c pas bon

L'application ne sera pas connectée à un réseau, la question de la sécurité du système contre des attaques extérieures ne devrait sûrement pas se poser. Mais le produit doit être programmé de telle sorte qu'il n'endommage pas le système hébergeant l'application.

## **5 Autres aspects du projet**

**5.1 Question ouvertes**

**5.2 Solution étagère**

**5.3 Problèmes nouveaux**

**5.4 Tâche à faire**

**5.5 Contrôle de la finalisation**

hello

**5.6 Risques liés au projet**

**5.7 Estimation des coûts**

**5.8 Documentation utilisateur et formation**

**5.9 Question mise en attente**

**5.10 Idées de solutions**