

THESIS / THÈSE

MASTER EN INGÉNIEUR DE GESTION À FINALITÉ SPÉCIALISÉE EN DATA SCIENCE

Vers de la Business Intelligence réellement Self-service grâce à la mise en œuvre d'un dashboard paramétrable

GUISE, Nicolas

Award date:
2023

Awarding institution:
Universite de Namur

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.



Vers de la Business Intelligence réellement Self-service grâce à la mise en œuvre d'un dashboard paramétrable

Guise Nicolas

Directeur : Prof. C. Burnay

Mémoire présenté
en vue de l'obtention du titre de
Master 120 en ingénieur de gestion, à finalité spécialisée
en data science

ANNEE ACADEMIQUE 2022-2023

Remerciements

Je tiens à exprimer ma profonde gratitude envers toutes les personnes qui ont contribué de manière significative à mon parcours académique et personnel. Le soutien que j'ai reçu a été inestimable, et je ne saurais jamais assez les remercier pour leur impact positif sur ma vie.

Tout d'abord, je voudrais adresser mes remerciements les plus chaleureux à mes parents. Leur soutien indéfectible, tant sur le plan psychologique que financier, m'a permis de poursuivre mes études avec détermination. Leur amour et leurs encouragements constants ont été ma source d'inspiration, et je suis profondément reconnaissant pour l'environnement qu'ils ont créé pour moi, me permettant ainsi de donner le meilleur de moi-même.

À ma compagne, je veux exprimer ma reconnaissance pour son soutien moral inébranlable. Sa présence bienveillante dans les moments difficiles a été une véritable source de motivation et son aide précieuse m'a permis de surmonter les obstacles avec confiance. Sa patience, son écoute et son amour ont été d'une importance capitale pour moi et ma réussite.

À mes amis, qui ont été à mes côtés tout au long de mon parcours universitaire et même bien au-delà, je dis un immense merci. Ces cinq années à l'université ont été enrichies par leurs sourires, leurs rires et leurs encouragements. Leur amitié m'a permis de vivre des moments inoubliables et m'a soutenu au travers des projets et des défis auxquels j'ai dû faire face. Je tiens à remercier tout particulièrement Jordan Abras, un ami cher, à qui je dois ma réussite académique. Les 12 années à ses côtés m'ont permis d'atteindre ce master. Sans lui, je peux confirmer que je n'en serais pas là. Au-delà des rires, des blagues et de la camaraderie, je le remercie pour son aide précieuse, son temps et son dévouement lorsque j'étais perdu. Je souhaite à Jordan Abras une réussite exemplaire pour tous ses futurs projets. Son engagement, son soutien et sa détermination méritent d'être reconnus et récompensés. C'est une fierté de finir avec lui.

Un remerciement spécial à mon directeur de mémoire, le Professeur Corentin Burnay. Son implication dévouée envers ses étudiants, son attention à nos besoins et sa gentillesse ont fait de cette expérience académique un réel plaisir. Son expertise et ses conseils éclairés ont été essentiels pour mener à bien ce mémoire.

Je tiens également à remercier chaleureusement Benito Giunta pour son aide et sa participation. De même, je suis reconnaissant envers Stéphane Ryckebusch et Pascal Walczak, employés chez Delaware, où j'ai effectué mon stage. Leur soutien et leurs conseils m'ont guidé à travers cette aventure professionnelle et académique avec succès.

En somme, ma réussite et ma croissance personnelle sont le résultat de l'investissement et du dévouement de chacune de ces personnes formidables. Je suis heureux d'avoir eu l'opportunité de côtoyer ces personnes tout au long de mon parcours. Mes plus sincères remerciements vous sont adressés pour avoir fait de mon cheminement une réussite.

Résumé/Summary

La Business Intelligence (BI) s'est imposée comme un pilier stratégique, permettant aux organisations de prendre des décisions éclairées en exploitant des informations pertinentes. Parallèlement, le concept de libre-service (self-service) s'est développé, offrant aux utilisateurs la possibilité de manipuler et d'explorer les données de manière autonome, réduisant ainsi la dépendance vis-à-vis des équipes techniques. Malheureusement, elle n'arrive pas à correctement trouver sa cible, car elle oscille entre une solution parfois trop complexe pour les utilisateurs non experts et trop simplistes/contraignantes pour les data scientists (experts). Afin de répondre à cette problématique, nous tâcherons, dans ce mémoire, de proposer une solution permettant à l'utilisateur d'être réellement indépendant quant à ses besoins en Business Intelligence. Il pourra la réutiliser dans n'importe quelle autre situation, autant de fois qu'il le souhaite, peu importe sa base de données, quel que soit son objectif.

Business Intelligence (BI) has established itself as a strategic pillar, enabling organizations to make informed decisions by exploiting relevant information. At the same time, the concept of self-service has developed, offering users the ability to manipulate and explore data autonomously, reducing dependency on technical teams. Unfortunately, however, it is not finding its target audience, as it oscillates between a solution that is sometimes too complex for non-expert users and too simplistic/constraining for data scientists (experts). In order to address this issue, we're going to propose a solution that gives users real independence when it comes to their Business Intelligence needs. He'll be able to reuse it in any other situation, as many times as he likes, whatever his database, whatever his objective.

Contents

1	Introduction	2
2	Problématique	4
3	Proposition de Solution	7
4	Méthodologie : choix et descriptions des méthodes de recherche	9
4.1	DSR : Présentation et explication de la méthodologie	9
4.2	la DSR dans le cadre académique?	11
4.3	Mise en application de la DSR à notre sujet	12
4.3.1	The Relevance Cycle - Le cycle de pertinence	12
4.3.2	The Rigor Cycle - Le cycle de la rigueur	13
4.3.3	The Design Cycle - Création d'artefact	13
5	Introduction à la business intelligence pour structurer notre revue de la littérature/Rigor Cycle	14
6	Phase 1 : Comment charger les données de manière user friendly?	18
6.1	Rigor Cycle	18
6.2	Relevance Cycle	19
6.2.1	Elicitations	19
6.2.2	Documentations	21
6.2.3	Validation/négociation	21
6.3	Design Cycle	22
7	Phase 2 – Comment établir le contenu du dashboard?	27
7.1	Rigor Cycle	27
7.2	Relevance Cycle	29
7.2.1	Elicitations	29
7.2.2	Documentations	29
7.2.3	Validation/négociation	29
7.3	Design Cycle	29
8	Phase 3 – Comment établir les visuels du dashboard?	33
8.1	Rigor Cycle	33
8.2	Relevance Cycle	36
8.2.1	Elicitations	36
8.2.2	Documentations	36
8.2.3	Validation/négociation	36
8.3	Design Cycle	36
9	Résultas	38
10	Discussion - Limitations - Future Works	39
11	Conclusion	41
A	Annexe : BPMN 2.0 - Business Process Model and Notation	45
B	Annexe : Documentation des exigences pour les dashboards	46
C	Annexe : KPI list definition and formula	48

1 Introduction

La Business Intelligence (BI) a considérablement évolué au cours des dernières décennies et est devenue un élément crucial dans le paysage des entreprises modernes, offrant un avantage compétitif significatif en termes de prise de décision stratégique. Dans un environnement en constante évolution, la disponibilité d'informations précises et pertinentes devient un atout majeur pour prendre des décisions éclairées et maintenir une longueur d'avance. Ainsi, les solutions de BI ont été développées pour répondre à cette demande croissante en matière d'analyse et de gestion des données.

Parmi les évolutions les plus significatives de la BI, l'émergence du concept de Self-service a révolutionné la façon dont les utilisateurs interagissent avec les données et les analyses. Cette dernière leur permettant d'accéder et d'exploiter les informations de manière autonome, sans dépendre de l'intervention d'une équipe informatique dédiée. Le Self-service BI permet aux utilisateurs non techniques, tels que les responsables opérationnels, les gestionnaires et les analystes métier, d'accéder et d'exploiter les informations pertinentes de manière autonome et rapide.

La BI en Self-service est définie par [Imhoff and White \(2011\)](#) dans leur ouvrage " *Self-service Business Intelligence*" comme étant " *les équipements de l'environnement de BI qui permettent aux utilisateurs de BI de devenir plus autonomes et moins dépendants de l'organisation informatique*". Selon eux, le SSBI (Self-service Business intelligence) se concentre sur 4 points (Figure : 1) :

- Un accès facile aux données sources pour les rapports et les analyses.
- Des outils de BI faciles à utiliser et un support amélioré pour l'analyse des données.
- Des options d'entrepôt de données rapides à déployer et faciles à gérer.
- Des interfaces utilisateur plus simples et **personnalisables**.

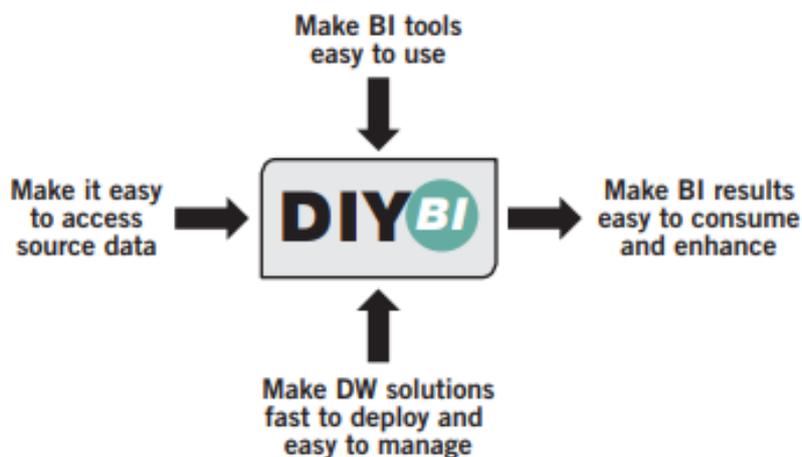


Figure 1: 4 SSBI features : [Imhoff and White \(2011\)](#)

En supprimant les obstacles traditionnels liés à la dépendance vis-à-vis de l'informatique, le Self-service BI permet aux utilisateurs de prendre des décisions informées en temps réel, améliorant ainsi leur réactivité et leur agilité dans un monde des affaires en constante évolution.

Cependant, malgré les avantages du Self-service BI, il existe toujours des défis à relever pour permettre une véritable autonomie et une exploitation optimale des données. Bien que les solutions BI se prétendent de plus en plus Self-service et user friendly, elles restent trop souvent techniques et difficiles à prendre en main pour des utilisateurs non experts. DAX, Power Query ou encore le langage M, sont des exemples d'outils trop rugueux dans Power BI pour des utilisateurs business. Nous discuterons plus précisément des problématiques dans la **Section 2 Problématique**.

En outre, c'est à cette problématique que mon mémoire et ma solution conceptuelle interviennent au travers du concept de dashboard paramétrable, une composante clé de la Business Intelligence ([Vaisman and Zimányi \(2014\)](#)). Un dashboard, ou tableau de bord ([Eckerson \(2010\)](#)), est une interface visuelle qui présente de manière synthétique les données essentielles et les indicateurs de performance d'une entreprise ([Sarikaya et al. \(2019\)](#)). C'est l'élément en première ligne d'un ensemble plus vaste de notions mais dont l'ajout de la fonctionnalité paramétrable offrira aux utilisateurs la possibilité de personnaliser et d'adapter les tableaux de bord en fonction de leurs besoins spécifiques, améliorant ainsi l'expérience utilisateur tout en appuyant l'approche Self-service grâce à des technologies que le business connaît vraiment.

2 Problématique

Lennerholt et al. (2018) dans leur revue de la littérature appelée "Implementation Challenges of Self-service Business Intelligence" présentent les défis associés à la mise en œuvre de la SSBI, ou encore dans leurs utilisations. L'ensemble des articles repris dans cette revue de la littérature nous ont permis de confirmer et justifier notre problématique ici présente (*étape 1 : Awareness of the problem*; Peffers et al. (2012), p.3-4).

L'outil Power BI est une solution d'analyse de données à travers la création de visualisation personnalisable. Dans un point de vue **interne** à l'entreprise, ce dernier a pour objectif de répondre à la tendance Self-service fort demandée par les entreprises actuellement. **95%** des organisations informatiques prévoient d'investir dans la SSBI, au cours des deux prochaines années, contre **84%** l'année dernière (2014) (Analytics (2015)).

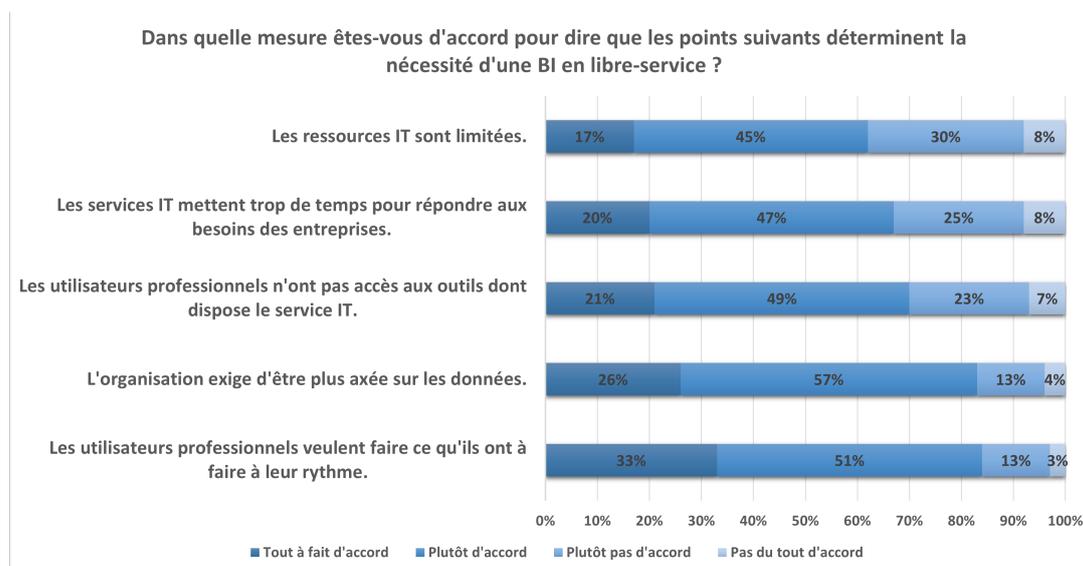


Figure 2: Analytics (2015) : "State of Self-service BI Report"

En effet sans cet outil, le manager doit envoyer sa requête à l'analyste, celui-ci traduit cette dernière en besoin technique et le responsable technique exécute cette réclamation (Alpar and Schulz (2016)). Une fois cela terminé, le chemin inverse doit alors être parcouru jusqu'au manager qui peut proposer au client les divers changements. Malheureusement, ce dernier s'est probablement mal exprimé et tout le processus doit être alors recommencé. Cela peut durer plusieurs itérations et durant plusieurs semaines, entraînant des coûts considérable en temps et en argent. (Lennerholt et al. (2018), p.5055-5056).

Dans leur études Logi Analytics confirme que près de 70% des personnes interrogées estiment que les "Power users" IT sont trop lents à répondre aux besoins du Business (Figure : 2).

Pour répondre à cela, la perspective Self-service proposée par Power BI apporte au manager la possibilité de choisir les données qui l'intéressent et de produire sur base de celles-ci son propre rapport de manière indépendante (c'est-à-dire sans avoir le besoin de passer par ses collègues analystes et techniques pour exécuter sa requête). Selon Analytics (2015) dans leur étude "State of Self-service BI Report" en 2015, pour 91% des personnes interrogées, il est nécessaire d'accéder facilement aux données, sans l'aide des services informatiques et des "Power user".

Malheureusement, Power BI reste un outil **complexe** et **chronophage** qui nécessite des connaissances approfondies dans cette plateforme et des connaissances métier en data science (Imhoff and White (2011)). En effet, nous pouvons constater que près de 20% des personnes interrogées (Figure : 3) ne trouvent pas que la SSBI améliore leur efficacité au travail et leur capacité à réagir rapidement au changement.

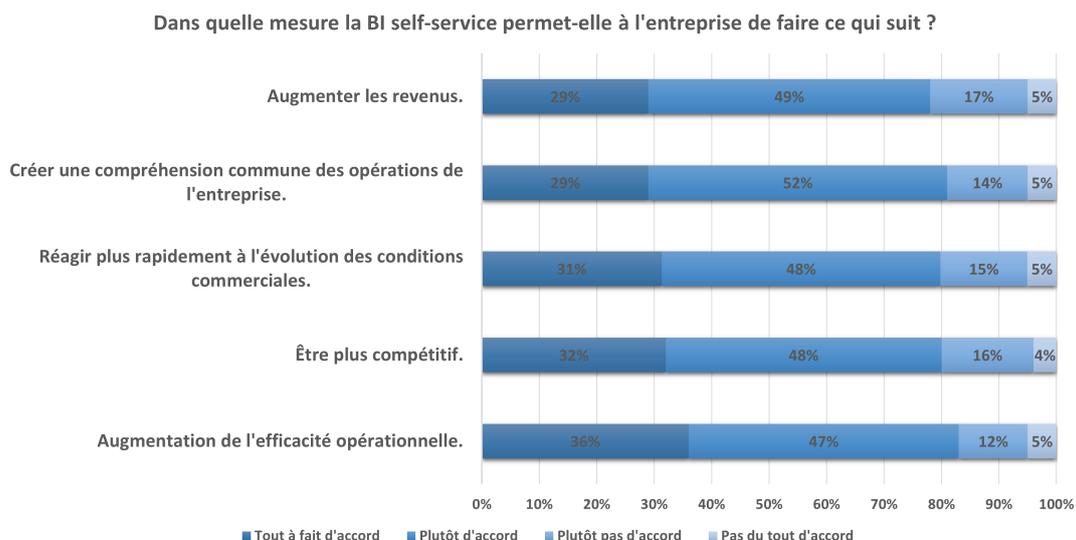


Figure 3: Analytics (2015) : "State of Self-service BI Report"

De plus, selon le rapport de Analytics (2015) sur l'état de la Self-service Business Intelligence, il est indiqué que le taux de réduction des requêtes effectuées par les Business ou les "casual users" grâce à la SSBI est de 47% (Figure : 4). Bien que cela puisse sembler significatif, il est important de noter que c'est 53% des requêtes qui continuent d'être adressées aux "Power users" de l'IT impactant plus de la majorité des projets qui ne sont pas totalement exécutés en Self-service. Cette constatation confirme que malgré la tendance vers le Self-service, les utilisateurs occasionnels restent dans la majorité des cas dépendants des compétences IT des "Power users".

Imhoff and White (2011) dans leur ouvrage "Self-service Business Intelligence" confirme ce paradoxe entre la nécessité pour l'IT de contrôler la création et la distribution des ressources BI et la demande des business de disposer de liberté et de flexibilité sans avoir besoin de l'aide des IT.

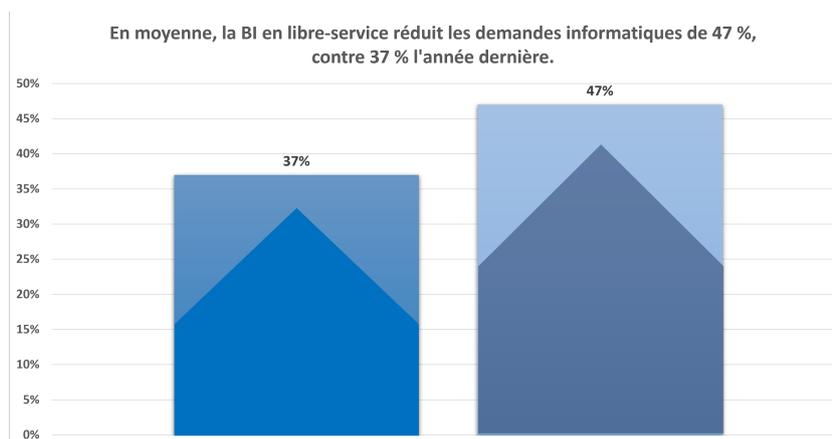


Figure 4: Analytics (2015) : "State of Self-service BI Report"

Enfin, nous pouvons conclure que la statistique la plus intéressante est celle qui confirme notre problème. La difficulté d'utilisation des outils SSBI est confirmée par plus d'un tiers des personnes interrogées (Figure : 5).

Une autre problématique concerne l'aspect externe de l'entreprise. Lorsque l'on arrive en tant que consultant en data science dans une nouvelle société, il est nécessaire d'exécuter une phase de requirements engineering (Pohl (2016)) pour comprendre les besoins du client en termes de reporting ainsi que leur base de données. Deux éléments importants mais obligatoires pour chaque nouveau client. Une

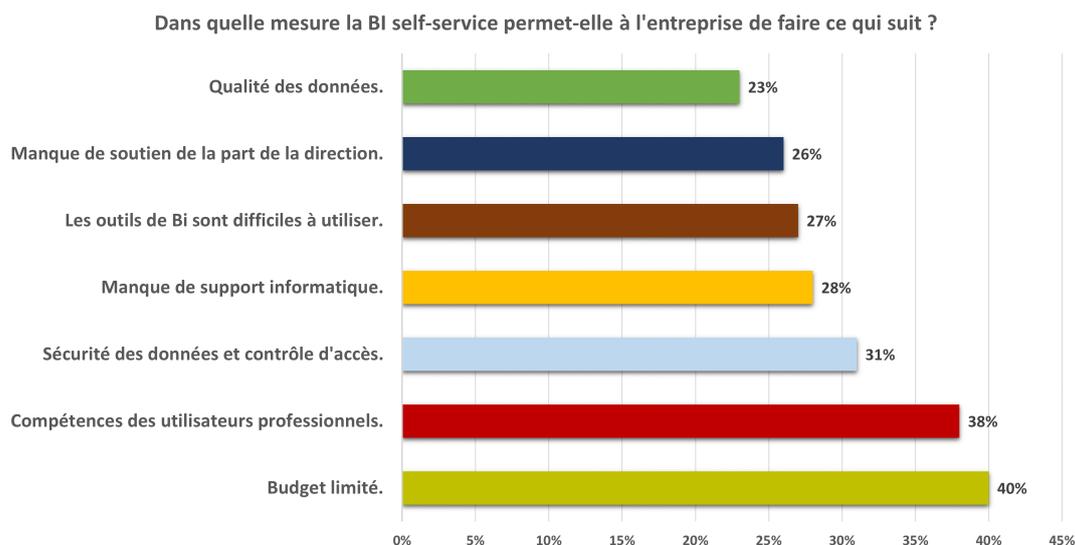


Figure 5: [Analytics \(2015\)](#) : "State of Self-service BI Report"

fois, cela fait, le data scientist met en place des visuels cohérents, des filtres ergonomiques et des dashboards design pour un rendu structuré et customisé selon la demande du client. Malheureusement, c'est une méthodologie ([Vaisman and Zimányi \(2014\)](#)) est chronophage car il faut généralement toujours la recommencer "from scratch" pour chaque nouveau projet.

Mais le sujet de cette seconde problématique est tel que nous pouvons remarquer que la **demande (basique)** en termes de reporting, dashboarding est globalement la même quelle que soit la compagnie dans laquelle un consultant travail. Un diagramme en bâtonnet, un line chart, un filtre temporel sur les données, sont des éléments redondants et régulièrement demandés par les entreprises. Les seuls changements significatifs sont les faits, les dimensions et les mesures qui sont derrière ces outils de visualisation. Cependant, pour chaque graphe, chaque dashboard et chaque compagnie, il est nécessaire de répéter l'entièreté de l'exercice "from scratch", entraînant un temps de développement inutilement et systématiquement long avec un coût financier correspondant.

Je peux personnellement témoigner de cette problématique, car j'ai eu l'occasion de l'expérimenter à plusieurs reprises pendant une courte période dans le cadre de mon expérience en entreprise. En effet, sur une durée de seulement 4 mois, j'ai été chargé de construire deux solutions Power BI à partir de zéro pour répondre à des besoins d'analyse et de navigation de données basiques. Cependant, c'est à moi que cette tâche a été confiée, car le "end user" ou encore mon responsable n'avaient tout simplement pas les compétences nécessaires pour construire cette solution sur Power BI. J'ai ainsi pu constater concrètement que le caractère Self-service de Power BI ne répondait pas à ses objectifs.

En résumé, le taux de mise en œuvre de l'approche SSBI est faible. Selon une enquête menée par Eckerson ([Eckerson \(2012\)](#)) auprès de 234 personnes, les deux tiers des 70% de professionnels qualifiés de la BI estiment que leur succès avec l'approche SSBI est moyen ou inférieur, tandis que plus de la moitié des "casual users" estiment que leurs tentatives de SSBI sont passables ou inférieures.

Nous pouvons donc affirmer sans hésitation qu'il existe un réel besoin d'améliorer le SSBI.

"En raison des coûts actuels et du temps nécessaire à la conception des tableaux de bord, il n'est pas possible de fournir à tous les employés un tableau de bord spécifiquement adapté à leurs besoins d'information." [Stoop \(2009\)](#).

3 Proposition de Solution

La réponse à ces deux problématiques, tant interne qu'externe oriente donc le sujet de mon mémoire comme étant : **Comment rendre une solution BI plus Self-service et par extension réutilisable?**

Cette solution (*étape 2 : Suggestion; Peffers et al. (2012) p.3-4*) sera composée de deux fichiers. Le premier sera un fichier **Power BI** générique, c'est-à-dire un template de dashboarding conçu une seule fois et dont chaque élément qui a été prédéfini est attribué à une variable. Le second fichier est un fichier **Excel**, externe à Power BI dans lequel chaque variable provenant du fichier Power BI pourra être assignée à une valeur.

En résumé, nous aurons donc un fichier de customisation sur Excel afin d'ajuster chaque dashboard sur Power BI. Le fichier Excel contiendra la **logique métier simplifiée** et extraite de Power BI. Tandis que ce dernier sera utilisé comme un orchestrateur alliant ETL et support de visuel et de rendu.

Par exemple, lorsqu'un consultant en data science arrivera chez un nouveau client, il ne devra plus travailler directement avec la plateforme Power BI et perdre du temps à la mise en page de chaque forme, graphe, couleur. Il pourra uniquement utiliser le fichier Excel pour définir quelques règles propres au client et ainsi avoir des résultats rapides sur le dashboard prédéfini sur Power BI. Comme la partie complexe et chronophage aura déjà été faite avant et sera la même pour n'importe quel client, n'importe qui avec n'importe quelle compétence pourra utiliser cette plateforme. Un outil similaire, proposé par une société Suisse dans un point de vue commercial est déjà sur le marché et est appelé "**Lintao**" mais ce dernier n'est pas suffisamment user friendly et donc nécessite encore des compétences IT provenant des "**Power users**".

Nous avons décidé d'opter pour l'outil Power BI de Microsoft plutôt que ses nombreux substituts. Pourquoi? Pour plusieurs raisons :

Tout simplement parce que Power BI est l'un des outils de Business Intelligence, surfant sur la vague Self-service, les plus utilisés sur le marché actuellement. Selon une étude réalisée par The Forrester en 2019 ([The Forrester \(2019\)](#)), Power BI de Microsoft se distingue par sa combinaison d'une offre solide, d'une stratégie forte et d'une part de marché importante, le propulsant directement comme étant le leader sur le marché (Figure : 6).

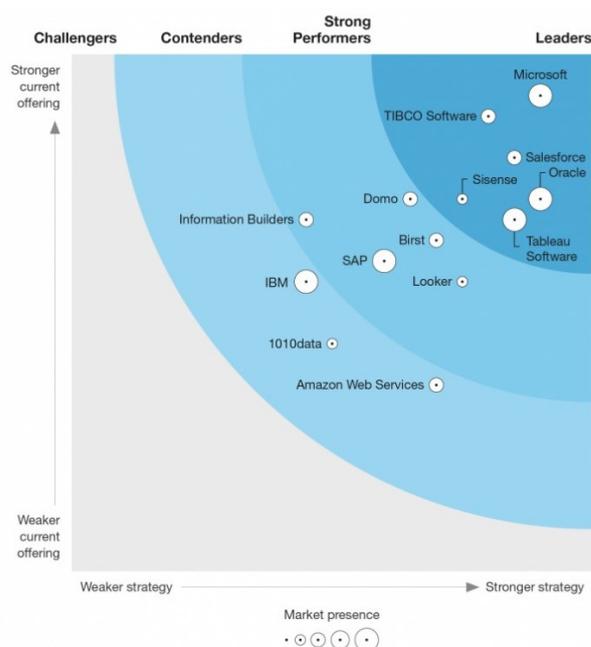


Figure 6: [The Forrester \(2019\)](#) : "Enterprise BI Platforms (Vendor-Managed)"



Figure 7: [Gartner \(2020\)](#) : "2020 Gartner Magic Quadrant"

Gartner a ensuite confirmé cette position en 2020 en désignant Microsoft et Power BI comme leaders avec une vision complète (Gartner (2020)) (Figure : 7).

En outre, suite à mon expérience professionnelle, j'ai pu constater, à travers mes échanges avec mes différents collègues, que Excel était encore largement utilisé dans le monde des affaires. Il est considéré comme un outil indispensable pour tous les métiers business en plus d'être un outil simple et facile à prendre en main.

Finalement, comme vous avez pu le constater, cette solution est fortement axée sur une partie du processus de modélisation de solution de Business Intelligence. Ce n'est pas un hasard si nous nous concentrons sur la simplification, à la fois sur le plan temporel et technique, de la partie visualisation et reporting. En effet, les trois points les plus importants concernant la SSBI sont les suivants (Figure : 8):

- Consulter un rapport ou un tableau de bord.
- Interagir avec un rapport ou un tableau de bord.
- Analyser des données et créer une visualisation.

Notre solution s'inscrit clairement dans ces trois points et accorde une importance significative aux utilisateurs de la SSBI.

En conclusion, la suite Microsoft est donc un choix pertinent pour développer cette solution, car ce sont des outils connus et largement utilisés dans de nombreux domaines par un grand nombre de personnes.

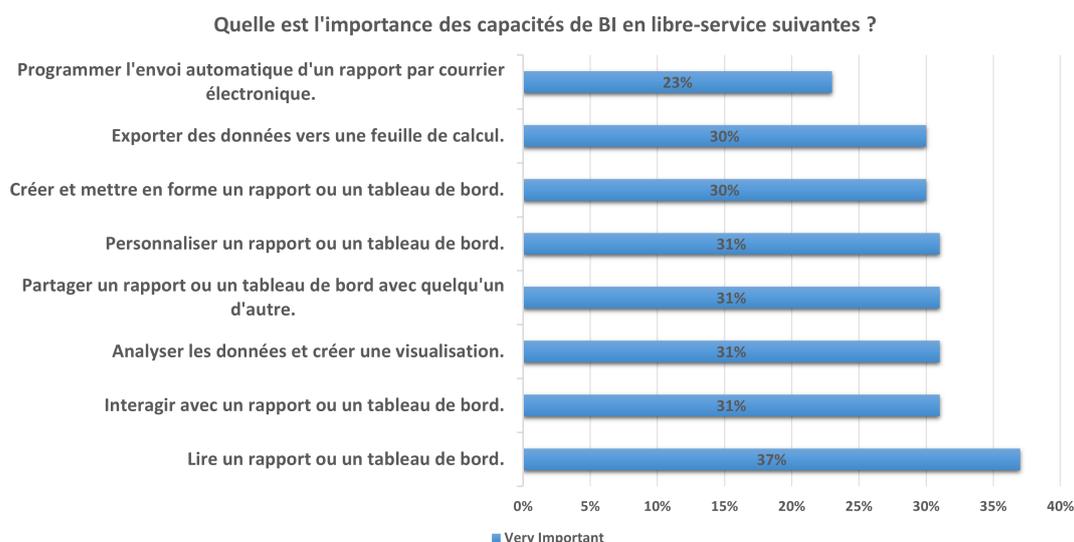


Figure 8: Analytics (2015), "State of Self-service BI Report"

L'objectif central de ce mémoire réside dans la communication d'une idée qui puisse être appliquée de manière étendue et évolutive (scalable) à des logiciels plus complexes, solides et cohérents. Dans le cadre de notre solution, le choix de Microsoft s'est imposé en raison des nombreux avantages qu'il offre (des outils simples à utiliser et favorisant une interconnexion aisée entre eux). Ces avantages ont été mis à profit pour élaborer un POC qui soutiendra ma vision conceptuelle. D'un autre côté, les outils Microsoft ont aussi leurs limites **Section 10 Discussion - Limitations - Future Works**, apportant leurs propres contraintes. Nous avons donc dû construire notre solution en tenant compte de ces aspects. Une section sur le résultat général (l'idée conceptuelle globale) de notre rapport est abordée à la **Section 9 Résultats**.

4 Méthodologie : choix et descriptions des méthodes de recherche

La méthodologie utilisée pour répondre à cette question est partagée entre un point de vue **académique** et un **pratique**. Ce que nous voulons dire, c'est que nous ne nous contentons pas seulement d'introduire un nouveau concept dans la littérature, mais nous l'accompagnons également de la construction d'un artefact pour étayer nos propos.

4.1 DSR : Présentation et explication de la méthodologie

La méthodologie appliquée tout au long de ce mémoire est tirée du domaine de la **Design science research (DSR)** et plus précisément de la démarche proposée par [Hevner et al. \(2010\)](#) dans leur ouvrage "*Design Research in Information Systems*". Ces derniers proposent une démarche en plusieurs **cycles et guidelines** pour développer et/ou améliorer un nouveau système d'information à partir d'artefact itératif avec l'objectif de les tester, évaluer dans un environnement concret et les justifier dans un cadre théorique.

Nous allons donc au cours de cette partie expliquer le choix d'une telle méthodologie pour ce mémoire. Nous allons pour cela présenter les aspects **théoriques et structurels** de la DSR dans le but de transposer ces éléments à notre projet afin de justifier la légitimité de ce dernier à suivre une telle approche.

Premièrement, nous pouvons définir la notion de design science research et ses domaines d'application. Cette dernière est présentée par [Hevner et al. \(2010\)](#), comme étant une approche de recherche visant à développer et évaluer de nouveaux artefacts dans le but de résoudre des "real-world" problèmes dans le cadre de système d'information IT ([Peffer et al. \(2012\)](#), p.197).

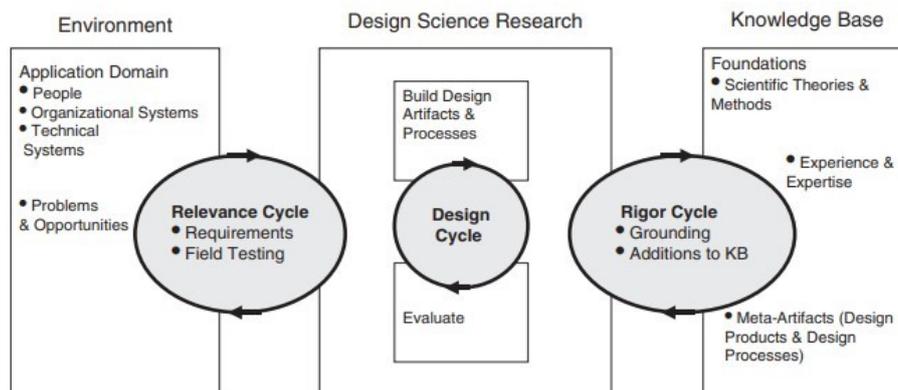
"La recherche en science de la conception est motivée par le désir d'améliorer l'environnement par l'introduction d'artefacts nouveaux et innovants et des processus de construction de ces artefacts ([Simon \(1996\)](#))" [Hevner et al. \(2010\)](#).

La DSR stimule la créativité humaine sur la construction et la conception d'artefacts ([Peffer et al. \(2012\)](#), p.401-404) qui sont d'abord **justifiés** sur base de **fondements scientifiques** puis seront ensuite testés dans des **environnements d'application "professionnels"**. Ces artefacts seront par la suite incrémentés par des "feedbacks", nouvelles idées provenant de l'environnement test jusqu'à l'obtention d'une solution IT complète répondant aux besoins business.

Nous arrivons facilement à dégager une structure très claire derrière la DSR. Comme énoncé précédemment, nous remarquons l'utilisation de cycles itératifs de conception, c'est-à-dire de mise en œuvre, d'évaluation, et de théorie.

[Hevner et al. \(2010\)](#) définissent ces trois cycles comme étant (Figure : 9) :

- "Le cycle de pertinence relie l'environnement contextuel du projet de recherche aux activités de conception scientifique."
- "Le cycle de rigueur relie les activités de la science de la conception à la base de connaissances des fondements scientifiques, de l'expérience et de l'expertise qui informe le projet de recherche."
- "Le cycle central de conception itère entre les activités principales de construction et d'évaluation des artefacts et des processus de conception de la recherche."

Figure 9: Design science research cycles : [Hevner et al. \(2010\)](#)Table 1: Design science research guidelines : [Hevner et al. \(2010\)](#).

Guideline	Description
Guideline 1 : Design as an Artifact	Design science research must produce a viable artifact in the form of a construct, model, a method, or an instantiation.
Guideline 2 : Problem relevance	The objective of design science research is to develop technology-based solutions to important and relevant business problems.
Guideline 3 : Design evaluation	The utility, quality, and efficacy of a design artifact must be rigorously demonstrated via well-executed evaluation methods.
Guideline 4 : Research contributions	Effective design science research must provide clear and verifiable contributions in the areas of the design artifact, design foundations, and/or design methodologies.
Guideline 5 : Research rigor	Design science research relies upon the application of rigorous methods in both the construction and evaluation of the design artifact.
Guideline 6 : Design as a search process	The search for an effective artifact requires utilizing available means to reach desired ends while satisfying laws in the problem environment.
Guideline 7 : Communication of research	Design science research must be presented effectively to both technology-oriented and management-oriented audiences.

De plus, un cadre conceptuel et un ensemble clair de Guidelines (Table : 1) principalement à destination des systèmes d'information, sont proposés par [Hevner et al. \(2010\)](#) afin de mener une DSR correctement.

La DSR est composée par deux catégories de recherches distinctes :

- **Le design en tant que recherche** : Nous nous concentrons dans ce cas-ci sur la conception d'éléments, d'artefacts innovants dans un secteur précis afin d'obtenir une compréhension plus grande des méthodes, théories et phénomènes au sein de l'environnement de l'artefact attendu. **Nous effectuons une recherche par le design.**
- **La recherche sur le design** : La recherche sur la conception se concentre sur le design, c'est-à-dire sur l'étude de la conception et de ses processus. La conception peut être définie comme un processus itératif de génération d'alternatives et de sélection d'une conception satisfaisante. **Nous effectuons une recherche sur le design.**

Une approche très similaire à celle proposée par [Walls et al. \(1992\)](#) dans leur article "*Building an Information System Design Theory for Vigilant EIS*" et repris par [Dresch et al. \(2015\)](#), p.78.

4.2 la DSR dans le cadre académique?

Maintenant que nous avons présenté ce qu'est la DSR, son domaine d'application et sa structure selon Hevner et al. (2010), nous pouvons appliquer et **justifier** le choix de cette approche à notre sujet de réutilisabilité de solution BI et d'amélioration du Self-service .

La méthodologie traditionnelle n'est pas adaptée pour ce mémoire "pratique", en effet nous devons nous concentrer sur la conception et l'élaboration de solutions innovantes et non sur la découverte de nouvelles connaissances ou la vérification d'hypothèses. Dans le cadre de ce projet et tout comme la DSR, **l'importance est mise sur la création d'une solution pratique dans un environnement réel**, plutôt que sur la production de connaissances théoriques qui peuvent ou non être directement applicables. Ceci répond à notre cas car nous devons développer une nouvelle solution répondant à une problématique métier dans le domaine IT.

Les connaissances existantes sont utilisées pour appuyer les concepts et notions déjà existantes pour développer nos artefacts. Cependant, les connaissances requises sont souvent inexistantes car nous sommes dans le cas de la création d'une innovation, quelque chose de nouveau.

Hevner et al. (2010) proposent aux **doctorants** qui souhaitent mener des études dans le domaine des SIs, un parcours et une méthodologie structurée en **légitimant leur recherche** itérative et/ou par essais erreurs.

Dans leur publication intitulée "*Design science research evaluation*", Peffers et al. (2012) détaillent comment ils ont employé la méthodologie globale du cadre de la Recherche en Science du Design dans le domaine des SIs. Ils illustrent ainsi comment leur recherche se révèle à la fois pertinente et rigoureuse. Leur travail contribue à enrichir le corpus de connaissances dans le domaine des systèmes d'information en abordant de façon résolue un problème croissant au sein de l'enseignement supérieur. Ces derniers expriment clairement dans cet article l'importance d'utiliser la DSR dans un cadre académique strict, afin de fournir une méthode d'évaluation simple et structurée.

"Les artefacts de la DLMA IS abordent le problème complexe (Hevner et al. (2004); Rittel and Webber (1973)) auquel sont confrontés les professeurs, où il existe une dépendance critique à l'égard des capacités sociales et de la créativité des étudiants pour générer un retour d'information à la fois sommatif et formatif pour les pairs." (Peffers et al. (2012), p.2)

Nous venons de prouver que l'utilisation de la DSR était légitime dans le cadre de recherche universitaire tels que des mémoires, comme le notre, à connotations pratiques qui ont pour objectif de fournir des résultats concrets au lieu de nouvelles théories.

4.3 Mise en application de la DSR à notre sujet

Dans ce paragraphe, nous appliquerons les différentes guidelines et cycles à notre projet afin de le structurer selon les normes DSR proposées par [Hevner et al. \(2010\)](#) dans leur ouvrage "*Design Research in Information Systems*".

Les **trois cycles** doivent être présents et clairement identifiables dans un projet de recherche utilisant cette méthodologie. Les sections suivantes expliquent brièvement chaque cycle avec leur mise en application dans notre sujet.

4.3.1 The Relevance Cycle - Le cycle de pertinence

Confrontation de notre artefact à la réalité ou domaine d'application (Section 6.2, 7.2 et 8.2).

L'environnement d'application est constitué de tous les éléments qui interagissent pour atteindre un objectif tels que des personnes, des systèmes organisationnels et techniques. [Hevner et al. \(2010\)](#) définissent la **première étape** de la DSR comment étant l'identification et la représentation des opportunités et des problèmes dans un environnement d'application réel. [Peffers et al. \(2012\)](#), indentifie cela à la première étape du processus en 5 étapes proposée par [Vaishnavi and Kuechler \(2015\)](#) : "*Étape 1 : Awareness of the problem*", (**Section 2 Problématique**).

En plus de définir les exigences de recherche (opportunité, problème), le contexte d'application caractérise également les critères intermédiaires d'évaluation d'artefacts afin de répondre à la question de savoir si l'artefact créé améliore l'environnement.

Mais comment l'analyser, le mesurer, l'évaluer? **Seconde étape**

Comme nous l'avons dit précédemment, à chaque itération d'artefact, celui-ci doit être renvoyé dans l'environnement pour être étudié, évalué (*étape 4 : Evaluation*; [Peffers et al. \(2012\)](#), p.3-4) par le domaine d'application. Une méthode d'analyse appelée **action research** est proposée par [Cole et al. \(2005\)](#) et [Järvinen \(2007\)](#). Mais dans notre cas, nous opterons pour une méthodologie de **requirements engineering** ([Pohl \(2016\)](#) "*Requirements Engineering Fundamentals*").

Sur base de cela, une autre itération (**troisième étape**) du cycle de pertinence commencera avec le retour à l'environnement d'un nouvel artefact construit et une reformulation des exigences au travers de cette nouvelle introduction.

"En d'autres termes, la pertinence pratique du résultat de la recherche doit être évaluée au même titre que la rigueur de la recherche effectuée pour obtenir ce résultat." ([Hevner et al. \(2010\)](#), p.12).

4.3.2 The Rigor Cycle - Le cycle de la rigueur

Introduction des notions et éléments de notre artefact dans la littérature scientifique (Section 6.1, 7.1 et 8.1).

La DSR se base sur un ensemble de connaissances qui peuvent être des méthodes d'ingénierie ou des théories scientifiques qui constituent les fondements d'une recherche rigoureuse. En plus des méthodes et théories, cet ensemble de connaissances utilisées dans le cadre d'une DSR comprend **deux types de connaissances additionnelles** :

- Les compétences pratiques et l'expertise qui définissent le niveau actuel de développement dans le domaine d'application de la recherche.
- Les artefacts et processus existants trouvés dans le domaine d'application.

Dans les parties précédentes, nous avons beaucoup parlé de la légitimité de notre artefact face au cadre scientifique mais son application se discute ici. Le cycle de rigueur permet d'utiliser les connaissances préexistantes au projet de recherche afin de stimuler l'innovation. Cela permet de s'assurer que les conceptions créées représentent une contribution significative à la recherche.

Il est donc impératif d'exiger que toutes les décisions et tous les processus de conception soient étayés par des théories ou méthodes solides appropriées pour construire et évaluer l'artefact. Dans notre cas, il sera donc primordial que nous suivions ce cycle de rigueur car nous travaillons dans la production d'un rapport académique scientifique qui ne doit pas être perçu comme un cas de conception de routine (c'est-à-dire fondé sur l'application de processus de conception connus et l'appropriation d'artefacts de conception connus). Pour notre projet, avant chaque nouveau cycle de création, nous étayerons les méthodes et théories que nous avons utilisées pour chaque nouvel artefact.

"Les contributions définitives de la recherche à la base de connaissances sont essentielles pour vendre la recherche à un public universitaire, tout comme les contributions utiles à l'environnement sont les principaux arguments de vente à un public de praticiens." (Hevner et al. (2004),p.18).

4.3.3 The Design Cycle - Création d'artefact

Le cycle de conception interne est le cœur de tout projet de recherche en design science (Section 6.3, 7.3 et 8.3).

La construction et l'amélioration d'artefact doivent être fondées de manière convaincante selon les informations récoltées dans le cycle de pertinence et de rigueur (étape 3 : Développement; Peffers et al. (2012), p.3-4). Nous pouvons parler d'un **paradoxe** de dépendance au sein du cycle de création et de pertinence car **avant** d'être mis à **l'essai dans l'environnement**, les artefacts doivent être **testés** de manière rigoureuse et approfondie en interne, en laboratoire.

Cette phase de test nécessite de multiples itérations internes et inhérentes au cycle de conception avant que les nouvelles fonctionnalités ne soient introduites dans le cycle de pertinence et par le cycle de rigueur. Dans notre cas, lors de la création d'un artefact, celui-ci sera premièrement testé en interne, comme nous le recommandons Hevner et al. (2010). Puis en parallèle, introduit dans son environnement pour éliciter des améliorations et fonctionnalités supplémentaires (**Section 4.3.1 Relevance Cycle**) et appuyé nos théories et méthodes sur une base scientifique rigoureuse. L'artefact en question sera considéré comme un prototype car comme nous rapporte Peffers et al. (2012), p.405, dans leur ouvrage *"Design science research evaluation"*, l'utilisation d'une instantiation de prototype pour démontrer l'efficacité d'une conception peut fournir des preuves solides lorsqu'elle est utilisée pour montrer qu'une conception fonctionne comme prévu, qu'elle est utile pour l'objectif visé ou qu'elle a le potentiel d'atteindre un niveau de performance.

"Simon (1996) décrit la nature de ce cycle comme la génération d'alternatives de conception et l'évaluation de ces alternatives par rapport aux exigences jusqu'à l'obtention d'une conception satisfaisante."(Hevner et al. (2010), p.18).

5 Introduction à la business intelligence pour structurer notre revue de la littérature/Rigor Cycle

Dans cette section, nous aborderons les concepts clés et principaux de la business intelligence tels qu'expliqués par [Vaisman and Zimányi \(2014\)](#) dans leur article "Data Warehouse System". Ces derniers proposent une introduction complète à la conception de data warehouse moderne. Les auteurs présentent les principes fondamentaux, les techniques et les outils nécessaires pour concevoir et mettre en œuvre une solution business intelligence efficace. Il était essentiel pour nous de faire un bref rappel de cette architecture, car cela nous permettra ensuite d'expliquer comment nous avons décidé d'articuler nos cycles de création (DSR) d'artefacts. Un entrepôt de données est une base de données particulière destinée à l'aide à la décision. Il récupère des données provenant de diverses bases de données opérationnelles et d'autres sources de données et les transforme en de nouvelles structures mieux adaptées au traitement analytique en ligne (OLAP, [Conn \(2005\)](#)). Les entrepôts de données sont basés sur un modèle multidimensionnel, où les données sont représentées sous forme d'hypercubes, avec des dimensions correspondant aux différents points de vue commerciaux et où les cellules contiennent les mesures à analyser.

Les bases de données classiques ne sont pas adaptées pour l'analyse de données. En effet, elles sont conçues pour gérer les activités quotidiennes d'une entreprise et sont appelées "bases de données opérationnelles ou OLTP" ([Hainaut \(2009\)](#)). Ces systèmes ne sont pas performants pour traiter des requêtes complexes qui nécessitent de nombreuses tables avec des jointures gourmandes (OLTP VS OLAP; [Conn \(2005\)](#), [Vangenot \(2003\)](#), p.19-22, Table : 2).

Table 2: Comparison between operational data base (OLTP) and data warehouse (OLAP) : [Vaisman and Zimányi \(2014\)](#).

Aspect	Operational Databases	Data Warehouses
User type	Operators, office employees	Managers, executives
Usage	Predictable, repetitive	Ad hoc, nonstructured
Data content	Current, detailed data	Historical, summarized data
Data organization	According to operational needs	According to analysis needs
Data structures	Optimized for small transactions	Optimized for complex queries
Access frequency	High	From medium to low
Access type	Read, insert, update, delete	read, append only
Number of records per access	Few	Many
Response time	Short	Can be long
Concurrency level	High	Low
Lock utilization	Needed	Not needed
Update frequency	High	None
Data redundancy	Low (normalized tables)	High (denormalized tables)
Data modeling	UML, ER model	Multidimensional model

Les caractéristiques d'un entrepôt de données sont ([Inmon \(1995\)](#)) :

- Orienté sujet : orienté selon les besoins analytiques des différentes zones de l'organisation.
- Intègre : intégrées à partir de plusieurs systèmes opérationnels et externes.
- Non volatile : ne peuvent pas être modifiées ou supprimées.
- Variable dans le temps (historisation).

Nous allons maintenant présenter une architecture et les différents niveaux d'un data warehouse ([Anand \(2014\)](#), Figure : 10) :

- **Le back end** de la base de données comprend les outils ETL ([Kimball and Ross \(2011\)](#)) et une zone de stockage de données. Les outils ETL extraient, transforment et chargent les données en provenance de différentes **sources de données**, internes ou externes à l'organisation, avant de les intégrer dans la base de données (**Section 6 Comment charger les données de manière user friendly?**).

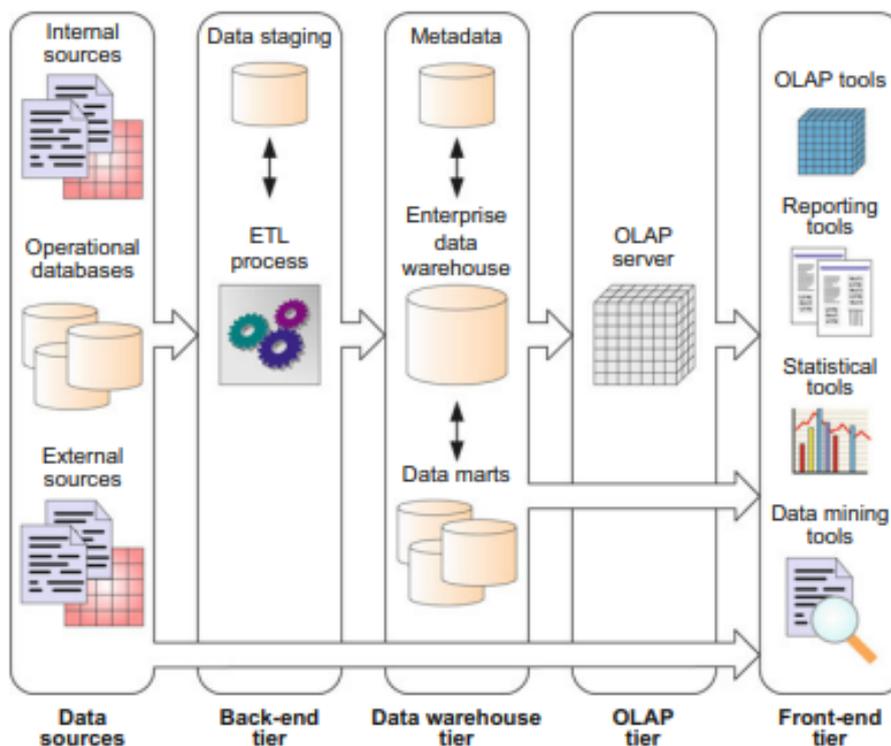


Figure 10: Typical data warehouse architecture : [Vaisman and Zimányi \(2014\)](#)

- **Le data warehouse tier** est constitué d'un entrepôt de données (EDW) et/ou de plusieurs data marts. L'EDW est centralisé et couvre l'ensemble de l'organisation, tandis que les data marts sont des entrepôts de données spécialisés destinés à une fonction ou à un département particulier dans une organisation.
- **Le niveau OLAP** est constitué d'un serveur OLAP, qui permet la construction et la requête de cubes de données, ainsi que la navigation, l'analyse et le reporting.
- **Le niveau Front-End** de l'architecture d'un entrepôt de données est utilisé pour permettre l'exploration interactive, la manipulation, l'analyse et la visualisation des données par l'utilisateur au moyen de différents outils :
 - Les outils de reporting (**Section 8 Comment établir les visuels du dashboard?**) sont utilisés pour la production, la livraison et la gestion de rapports ([Sarikaya et al. \(2019\)](#)).
 - Les outils statistiques pour l'analyse et la visualisation des données de cube.
 - Les outils de data mining (**Section 7 Comment établir le contenu du dashboard?**) pour l'analyse des données afin de découvrir des connaissances précieuses telles que des motifs et des tendances.

En plus de cela nous allons suivre la méthodologie de construction proposée par Vaisman, A. et Zimányi, E. (2014, Figure : 11).



Figure 11: Conception phases in data warehouse design : [Vaisman and Zimányi \(2014\)](#)

La première étape est la **spécification des exigences** et il existe également deux/trois approches possibles pour cette phase (Kimball and Ross (2011), p.340-345) :

- **L’approche d’analyse** est basée sur les besoins des utilisateurs clés de différents niveaux organisationnels.
- **L’approche de la source** est basée sur l’analyse des systèmes sources de données.
- **Analyse/source** est une combinaison des deux autres approches, visant à répondre aux besoins d’analyse des utilisateurs en utilisant les informations fournies par les systèmes sources.

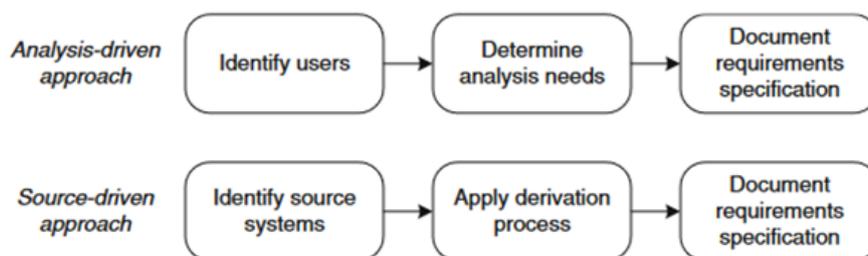


Figure 12: Steps for requirements specification in the analysis/source-driven : Vaisman and Zimányi (2014)

Dans le cadre de notre projet, nous utiliserons cette dernière (Figure : 12) car nous l’avons expliqué antérieurement (**Section 4.3 Mise en application**); nous devons prendre en compte les besoins utilisateur mais d’un autre côté le système nouvellement implémenté dépendra aussi des plateformes source et de leurs caractéristiques.

La deuxième étape de conception sont les **modèles conceptuels**. Il est largement reconnu que leur utilisation pour la conception de bases de données présente de nombreux avantages. Un schéma conceptuel est une description des exigences des utilisateurs en matière de données, sans tenir compte des détails de mise en œuvre. Le modèle conceptuel présenté par Vaisman and Zimányi (2014) est le model MultiDim. Ce dernier est utilisé pour représenter tous les éléments requis dans les entrepôts de données. Il est composé d’un ensemble de **dimensions (hiérarchie, niveau, granularité, agrégation) et de faits (mesure)**.

Une fois notre modèle conceptuel dessiné nous pouvons le transformer en **modèle logique** au travers de règles de mappage MULTIDIM – RELATIONNEL (Vaisman and Zimányi (2014), p126). Les auteurs se concentrent sur quatre implémentations typiques de la représentation relationnelle (**schémas en étoile, en flocon de neige, en flocon d’étoiles et en constellation** (Vaisman and Zimányi (2014), p123-126) et Inmon (1995))).

Pour la **conception physique**, Vaisman and Zimányi (2014) proposent une méthode en quatre étapes (Partitionnement, Allocation, Organisation, Répartition) afin de garantir des temps de réponse de requête adéquats (Kimball and Ross (2011), p.355-357).

Comme nous pouvons le constater, il existe un paradoxe entre la création BI (Figure : 11) qui suit une modélisation waterfall ou pas-à-pas, tandis que notre méthodologie DSR, suit un flux agile (Figure : 9).

Pour répondre à cela, nous n’allons pas itérer (de manière DSR) sur chaque étape de notre processus de conception BI traditionnelle, mais sur l’artefact final. Nous pouvons donc résumer cela en disant que nous introduisons le processus BI (Figure : 11) au sein du cycle de design (**Section 4.3.3 Design Cycle**). Ce cycle sera introduit par les exigences récupérées grâce au cycle de pertinence (**Section 4.3.1 Relevance Cycle**) et débouchera finalement sur le cycle de rigueur (**Section 4.3.2 Rigor Cycle**) pour définir les notions mobilisées. Cela nous donnerait notre première itération et notre premier artefact (**artefact 1**).

Une fois cela fait, nous recommençons tout depuis le début. Les exigences nouvellement perçues au moyen de notre artefact 1, introduiront notre nouveau cycle de design (**Section 4.3.3 Design Cycle**), avec la première étape du process BI qui est justement la "spécification des exigences" (Figure : 11).

Cependant pour ne pas reprendre l'intégralité de chacun de nos prototypes dans ce mémoire, nous nous concentrerons sur le produit final (ce qui est le plus important) en le présentant sous forme de 3 phases.

Notre **première phase** sera axé sur l'ETL (Extract, Transform, Load) (**Section 6 Comment charger les données de manière user**). En effet, nous avons un défi majeur consistant à rendre notre ETL dynamique (paramétrable), c'est-à-dire qu'il doit être capable de se connecter à n'importe quelle source de données et de charger les données dans notre modèle de données sans aucune manipulation manuelle de la part de l'utilisateur.

Une fois cette étape accomplie, nous aborderons la question de ce qui doit être affiché pour l'utilisateur, c'est-à-dire le "quoi?" [Stoop \(2009\)](#), p.49. Pour cela, nous nous orienterons vers la notion de **KPI (Key Performance Indicators)** (**Section 7 Comment établir le contenu du dashboard?**).

Enfin, dans notre dernier cycle, nous répondrons à la question du "comment?" [Stoop \(2009\)](#), p.49. Cela signifie que nous explorerons les différentes façons d'afficher nos informations sur un **dashboard**, les types de graphiques à utiliser, leur importance sur un tableau de bord, leur interactivité, etc (**Section 8 Comment établir les visuels du dashboard?**).

6 Phase 1 : Comment charger les données de manière user friendly?

6.1 Rigor Cycle

”Le processus d’extraction, de transformation et de chargement (**ETL**) est une composante importante de la BI. Il décrit la collecte de données à partir de diverses sources (extraction), leur modification pour correspondre à un état souhaité (transformation) et leur importation dans une base de données ou un entrepôt de données (chargement).” (Anand (2014)).

Une implémentation performante du processus ETL est une tâche cruciale et très rigoureuse dans la cadre de l’implémentation d’une solution BI. Cela peut représenter jusqu’à 80% des efforts déployés dans les projets de BI (Inmon (2002)). Cela renforce notre idée de créer une solution unique qui réduirait le temps de développement et les coûts monétaires (Vaisman and Zimányi (2014), p.285).

Le processus ETL se compose de trois éléments de base :

- **Extraction** : ”Est le processus d’identification et de collecte des données pertinentes à partir de différentes sources.” Anand (2014). Dans la première phase, les données sont extraites de systèmes d’exploitation hétérogènes tant internes que externes (Source de données : Figure : 10). Le but étant de sélectionner les données qui sont significatives pour soutenir la prise de décision au sein de l’organisation. Cette étape ne doit avoir aucun impact négatif sur les performances du système.
- **Transformation** : Lors de cette étape, nous appliquerons un ensemble de modifications nécessaires pour transformer notre donnée en informations pouvant être interprétées en termes commerciaux. Nous pouvons aussi parler de data preprocessing (analyse de qualité, data cleaning, ...) (Alasadi and Bhaya (2017) et Mhon and Kham (2020)) vers des formats cohérents pour le reporting et l’analyse. Le processus comprend aussi la définition de la logique d’entreprise, le mappage des données et la normalisation.
- **Chargement** : Finalement, les données sont poussées dans l’entrepôt de données.

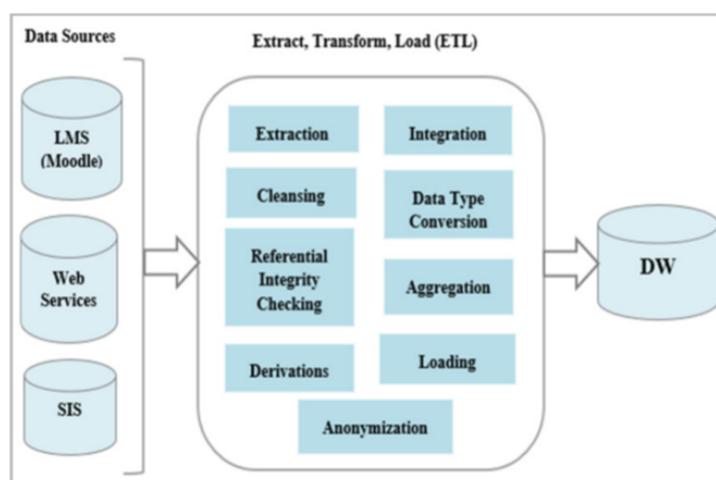


Figure 13: Steps within ETL Process : Mhon and Kham (2020)

L’ETL est un terme bien établi et le marché est déjà occupé par plusieurs acteurs majeurs. Cependant les outils de BI open source ont récemment gagné en popularité (Gartner (2020), Figure : 7) et commencent à rivaliser avec les solutions commerciales (Anand (2012)).

La complexité de la mise à jour des données stockées n’est pas à négliger. Un ETL peut être exécuté soit en mode batch, soit en temps réel. Le mode batch est généralement planifié et exécuté périodiquement. Si les intervalles entre les exécutions sont de quelques heures, voire quelques minutes, on parle de processus en temps quasi réel. En tenant comptes des phases mentionnées précédemment,

leurs exécutions en temps quasi réel sont extrêmement complexes. Une performance élevée est donc essentielle pour pouvoir traiter de grandes quantités de données et disposer d'une base de données actualisée (Anand (2012)).

Dans le but de modéliser un ETL, Vaisman and Zimányi (2014) suggèrent d'utiliser la norme de facto BPMN pour spécifier les processus d'entreprise. Cette approche est largement connue et peut être facilement comprise par le grand public, qu'il ait ou non des compétences métier, et peut être traduite rapidement en spécifications exécutable pour les outils ETL.

Les éléments utilisés dans notre modélisation de l'ETL (Section 6.3 et 7.3 Design Cycle) sont définis par Dumas et al. (2013) dans leur livre "Fundamentals of Business Process Management". Le BPMN présente une dualité, car il permet de modéliser simplement les processus d'entreprise tout en pouvant gérer leur complexité inhérente. Vous pouvez trouver tous les éléments, objets, etc., dans l'annexe A et des définitions des éléments cruciaux à un ETL dans Vaisman and Zimányi (2014), p.286-291.

6.2 Relevance Cycle

Dans cette section, nous regroupons toutes les exigences perçues en termes de modélisation de l'ETL. Pour ce faire, nous avons suivi les conseils de Pohl (2016), qui nous permettront de capturer de manière exhaustive et qualitativement les besoins des clients, tout en étant conscient des problèmes possibles. Pour garantir le succès de cette phase, il est impératif d'intégrer les trois activités clés de l'ingénierie des exigences (élicitation, documentation, validation et négociation). Une remarque importante à souligner est que le processus d'élicitation (Section 6.2.1 Elicitations) dans nos trois phases de développement (Section 6, 7, 8) est le même. En d'autres termes, nous n'avons pas appliqué des phases d'élicitation uniquement tournées autour de l'ETL, des KPI et des tableaux de bord. Nous avons effectué trois cycles d'élicitations (avec les trois méthodes d'élicitations expliquées plus loin) orientés sur la solution globale. Puis, dans un second temps, nous avons nous-même, a posteriori, reparti les exigences en termes de phases (Section 6, 7, 8.) dans les parties correspondantes (Section 6.2.2, 7.2.2, 8.2.2 Documentations) .

6.2.1 Elicitations

Avant de procéder à la phase d'élicitation des exigences, il est important de définir le **contexte du système**. Le contexte englobe toutes les parties de la réalité qui peuvent influencer les exigences finales du système. Cela inclut les parties prenantes, le système lui-même, les processus, les événements et les documents pertinents.

Il est nécessaire de distinguer clairement le contexte du système à développer et les éléments de la réalité qui ne sont pas pertinents pour le système en question. Dans **cette phase**, nous nous concentrons sur le **sous-système ETL** de notre solution finale. Dans la Section 7.2.1 Elicitations, le système sera le **sous-système KPI**, tandis que dans la Section 8.2.1 Elicitations, le système sera le **sous-système dashboard** de notre solution globale (Figure : 14).

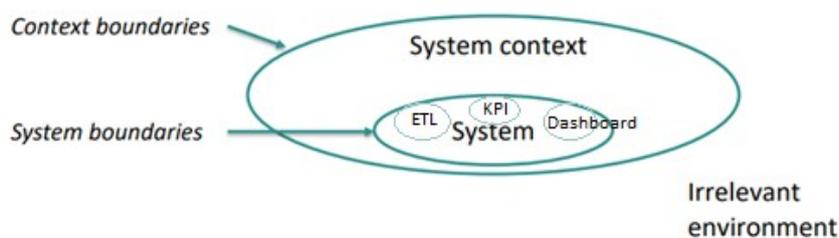


Figure 14: System and context boundary of a system : Pohl (2016)

Pour obtenir une vue d'ensemble des fonctionnalités souhaitées par les parties prenantes, nous utiliserons diverses techniques d'élicitation des besoins. Tout au long de ces techniques, nous garderons un esprit critique, car il est possible de supposer que toutes les parties prenantes ne fournissent pas des informations totalement objectives, même si elles ne sont pas mal intentionnées. Leurs déclarations

peuvent refléter leur perception du problème plutôt que la réalité objective. De plus, les suppositions peuvent être basées sur leurs expériences et/ou leurs connaissances passées.

Si les connaissances de la partie prenante dans le domaine sont très élevées, il ne lui sera pas difficile de fournir ce que l'on appelle des déclarations expérientielles, car elle sera en mesure de fournir des informations plus complètes et peut-être basées sur son expérience passée. Cependant, d'une part, il faut toujours faire attention au biais de mise en œuvre, et d'autre part, ces déclarations manqueront cruellement d'inventivité et de créativité pour trouver la bonne solution.

À l'autre extrême, si la partie prenante est totalement novice dans le domaine, elle fournira des déclarations hypothétiques. Celles-ci seront beaucoup plus créatives, mais contribueront-elles vraiment à résoudre le problème?

Pour compenser cela, nous avons décidé de nous baser sur deux types de personnes différentes :

- Un **praticien** provenant du monde professionnel ("Pro") : Consultant BI intervenant sur la construction d'une solution data (système opérationnel, ETL, restitution et reporting). Spécialisé sur les outils SAP tels que ERP-S/4HANA, ETL-BW, restitution-STACK.
- Un **théoricien** provenant du monde académique ("Aca") : Un Doctorant en ingénieur de gestion dont la thèse est au sujet de la modélisation de KPI complexe par la méthodologie du Model Driven Engineering.

De plus, il ne s'agit pas seulement d'essayer d'explorer les demandes explicites formulées par une partie prenante, mais aussi les demandes inconscientes ou subconscientes qu'elle formule en développant son point de vue sur le problème.

Afin d'obtenir une phase d'élicitation riche, nous avons décidé d'utiliser différentes techniques que nous avons adapté au fur et à mesure des cycles de notre DSR.

Nous avons commencé par des **entretiens guidés** (Burnay et al. (2014)) et libres pour favoriser le **brainstorming** (Pohl (2016)), parfois le **sketching** (Abad et al. (2018)) et stimuler la créativité. Ensuite, nous avons utilisé le **prototypage** et finalement le "**Wizard of oz**" (Abad et al. (2017)).

Pour notre **premier cycle**, nous avons décidé de commencer par des **entretiens classiques**, en combinaison avec des sessions de brainstorming et parfois du sketching. Nous avons posé des questions prédéterminées à chacune de nos parties prenantes, individuellement. Tous les échanges ont été documentés dans un document à part. Cependant, nous avons également laissé place à une conversation ouverte, afin de laisser la créativité de nos interlocuteurs s'exprimer (méthodes synchrones).

Dans notre **deuxième cycle**, nous utilisons le **prototypage** comme support pour obtenir des retours, des contraintes, de nouveaux besoins et mettre en question les exigences établies d'un point de vue plus pratique qu'imaginaire (Pohl (2016), p.30).

Pour notre **dernier cycle**, nous utiliserons la méthode du "**Wizard of oz**" afin d'analyser la facilité de l'utilisabilité de la solution, sa convivialité et autres caractéristiques similaires en effectuant une série de tâches avec la solution presque aboutie. Abad et al. (2017) définit cela au travers d'un paper élicitations mais nous pouvons très bien le postposer sur un système d'information (Wik (2015)). En général, on peut dire que le "Wizard of oz" s'est révélé très efficace pour capturer des exigences qui avaient été manquées lors des étapes précédentes de collecte ou pour les valider (Alasadi and Bhaya (2017)).

6.2.2 Documentations

L'ensemble des déclarations émises par les deux interviewés ont été reprises dans ce tableau (Table : 3) et rédigées en langage naturel en utilisant le modèle **THE SYSTEM SHALL/SHOULD/WILL/MAY** proposé par Pohl (2016) afin de réduire les ambiguïtés et la subjectivité lors de la transformation en langage naturel. Cette phase est très importante et délicate, car elle nous permet de transformer les souhaits des clients en exigences pour le système. L'ensemble des interviews sont présentées dans un "repository". Toute interprétation erronée des déclarations peut conduire à l'insatisfaction et à de nombreuses modifications après la phase de validation.

Chaque ligne de ce tableau représente une exigence, cette dernière est définie selon un ID, le NL requirements, le cycle lors de laquelle elle a été identifiée, la technique d'élicitations utilisée et sa source. La colonne "priority" nous indique l'ordre de priorité des implémentations des exigences dans une future itération de notre solution. Nous nous concentrerons uniquement sur les exigences ayant une priorité de 1 à 3, tandis que les exigences 4 et 5 seront abordées dans notre **Section 10 Discussion - Limitations - Future Works**.

Dans cette section, nous mettrons en évidence les principaux types d'exigences non fonctionnelles (ENF) que nous supposons nécessaires pour définir l'attribut de qualité d'un système et la manière dont le système doit répondre aux exigences fonctionnelles (Pohl (2016), p.8) :

- **Performance** : Chaque requête doit être suffisamment rapide pour aboutir à une utilisation de la solution agréable et tenante (connexion à l'ensemble des données, exportation vers l'outil de visuel ou encore modification de nos choix de création).
- **Scalability** : Garantir la même qualité de service même lorsque le nombre de données ou la complexité des inputs de l'utilisateur augmente.
- **Serviceability** : Fournir un service de qualité avec une utilisation facile.
- **Security** : La plateforme doit maintenir une garantie de conservation de tous les types de données et garantir la sécurité contre tout type de cyber-attaque.
- **Data Integrity** : Le programme doit garantir l'exhaustivité, l'exactitude, l'authenticité, la validité, l'absence d'altération des données et un format utilisable.
- **Interoperability** : Le système doit pouvoir répondre facilement aux connexions et aux échanges d'informations avec d'autres systèmes.

Notre objectif ici n'est pas d'élaborer un rapport d'ingénierie des exigences, c'est pourquoi nous avons énuméré les principaux besoins et que cette liste n'est en aucun cas exhaustive.

6.2.3 Validation/négociation

La technique principale de validation est le prototypage (ici : WOZ), car c'est la méthode la plus efficace pour identifier les erreurs dans les exigences. Les parties prenantes peuvent essayer le prototype et comparer leur propre idée de la manière dont le système devrait être implémenté avec le prototype disponible. Cela leur permet de détecter les divergences entre leurs idées et la solution actuelle.

Pour ceci, nous avons décidé d'utiliser notre dernière et troisième technique d'élicitation, le "Wizard of oz" (Wik (2015) et Abad et al. (2017)). Cette technique combine à la fois le prototypage (Abad et al. (2018)) et la validation car elle permet à l'utilisateur de manipuler directement la solution et y comparer la réalité de ces attentes : "La technique du magicien d'Oz ne doit pas seulement être utilisée comme technique de validation ou de test." (Wik (2015)).

Comme vous pouvez le constater, le nombre de besoins identifiés diminue en fonction de chaque technique. Cette tendance peut s'expliquer par une meilleure compréhension de notre solution et de ses besoins (que ce soit d'un point de vue académique ou professionnel) au fur et à mesure des cycles de création de notre DSR. En effet, nous observons un nombre significatif de besoins issus de la technique de brainstorming, tandis que la technique du WOZ en a généré moins. En réalité, cette dernière méthode ne vise pas à ajouter un grand nombre de besoins, mais plutôt à les confirmer ou les négocier (et à

incorporer des fonctionnalités plus précises basées sur des retours ”sur le terrain”). Ainsi, nous pouvons affirmer que ce cycle de pertinence a été conduit avec rigueur, en respectant les directives proposées par Hevner et al. (2010), et s’est avéré fructueux, répondant aux exigences en matière d’ingénierie des besoins telles que définies par Pohl (2016).

6.3 Design Cycle

Lors de cette section, nous aborderons la solution que nous avons implémentée pour illustrer nos propos et soutenir notre idée conceptuelle. Cette représentation reflète notre vision précise et actuelle, mais dans la **Section 9 Résultats**, vous trouverez une idée générale et évolutive (scalable). Cette introduction pourra également être appliquée aux **Section 7.3** et **Section 8.3 Design Cycle** concernant les KPI et les tableaux de bord. Afin de clarifier notre processus ETL, il est crucial de présenter les acteurs clés de notre solution (Figure : 15) :

- **[Excel]** : Nous disposons ainsi d’un **fichier Excel de personnalisation**, c’est-à-dire un template que l’utilisateur remplira pour créer son tableau de bord sur Power BI. Ce fichier représente ainsi l’acteur majeur de l’interaction entre l’utilisateur et la solution.
- **[Data Base]** : La **base de données** que l’utilisateur souhaite analyser (Dans notre cas un fichier plat Excel).
- **[Power BI]** : Un **fichier Power BI**, il représentera en façade une visualisation de sa base de données conformément aux exigences et aux demandes formulées au travers du formulaire. Toutefois, du point de vue backend, c’est à dire la partie dont l’utilisateur n’aura pas et ne doit pas avoir accès (pour répondre à notre problème initial), Power BI Query jouera le rôle d’orchestrateur entre la **base de données** et le **fichier Excel de personnalisation**.
- **[User]** : Finalement, notre utilisateur.

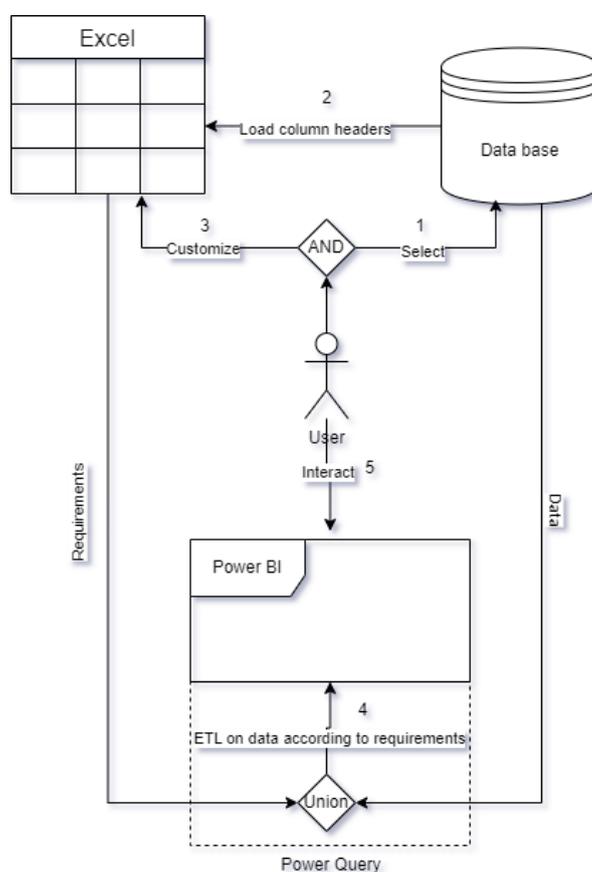


Figure 15: Acteurs clés de notre solution IT

Maintenant que notre schéma "Overview" sur l'orchestration de l'ensemble de la solution a été présenté, nous pouvons entrer plus en détail sur le fonctionnement technique de cette solution. Tout d'abord, il est important de souligner l'importance de la paramétrisation dans notre solution, c'est-à-dire "Dans le langage informatique, la paramétrisation permet d'optimiser un contenu numérique à l'aide de nouvelles données afin d'augmenter son efficacité." [linternaute \(2023\)](#).

En d'autres termes "paramétrer" est utilisé dans divers contextes pour désigner l'action de définir et de configurer les paramètres ou les réglages d'un système, d'une application, d'un logiciel ou d'un processus, afin de les adapter à des besoins spécifiques ou des conditions particulières. Paramétrer implique donc de spécifier les valeurs et les options qui influencent le comportement ou la performance d'un élément donné (dans notre cas notre solution IT).

Les objectifs de la mise en place de paramètres sont variés mais permettent dans notre cas :

- **La personnalisation** : Paramétrer permet d'adapter un système ou un logiciel aux préférences et aux besoins spécifiques de l'utilisateur. Cela peut inclure des préférences d'interface utilisateur, des options de configuration ou des fonctionnalités activées/désactivées en fonction des préférences individuelles. Dans notre cas c'est exactement ce qui est recherché.
- **L'adaptation au contexte** : Dans le domaine de l'IT, le paramétrage permet d'ajuster le comportement des systèmes en fonction des données et des conditions du moment. Par exemple, un algorithme peut être paramétré pour s'adapter à de nouveaux besoins.

1. La relation [Select] entre le **User** et la **base de données [Data base]** :

Le **User** dépose la base de données souhaitée au format Excel dans un dossier.

2. La relation [Load column headers] entre la **base de données [Data base]** et le **fichier Excel [Excel]** :

Dans cette relation, nous avons établi une connexion entre notre fichier de personnalisation Excel [Excel] et un dossier de stockage. L'objectif était de permettre à l'utilisateur de connecter facilement n'importe quelle base de données [Data base] à notre solution, en lui offrant un moyen simple d'effectuer cette action. Pour ce faire, l'utilisateur doit simplement inclure sa base de données [Data base] au format Excel dans ce dossier spécifique. Une fois cette étape accomplie et une fois que le fichier de personnalisation Excel [Excel] a été mis à jour, la connexion entre ce fichier de personnalisation [Excel] et le dossier en question va automatiquement repérer la base de données [Data base] présente, ouvrir la première feuille de ce fichier [Data base] et charger l'intégralité des en-têtes et des colonnes.

3. La relation [Customize] entre le **User** et le **fichier Excel** :

Une fois les étapes 1 et 2 effectuées, l'intégralité du fichier de personnalisation [Excel] fonctionne sur le principe d'une liste déroulante des en-têtes de colonnes récupérées de la base de données [Data base]. L'utilisateur peut ainsi remplir le fichier Excel de personnalisation [Excel] avec ses besoins (Figure : 16).



Figure 16: Exemple de personnalisation d'un graphique avec le fichier de personnalisation [Excel].

L'intégralité de chaque "input" de l'utilisateur sera enregistrée dans des tableaux Excel et regroupée selon les caractéristiques, par exemple l'ensemble des données liées au titre (police, taille, texte) ou l'ensemble des "inputs" liés au premier graphique (titre, colonnes choisies pour l'axe des x et y, fonction d'agrégation, ...) seront des données regroupées dans des tableaux respectifs à une ligne (Figure : 16).

4. La relation [ETL on data according to requirements] entre Power Query, le fichier Excel [Customize] et la base de données [Data base] :

- **Extract** : Power Query va charger la table présente sur la feuille 1 du fichier Excel [Data base] et en même temps charger tous les tableaux présents dans le fichier Excel de personnalisation [Excel].
- **Transform** : Ensuite, nous allons détailler le processus de transformation pour deux types d'éléments; **les graphiques (identique pour les filtres) et le titre, par exemple (identique pour le logo)**.

Pour le **titre**, nous allons simplement extraire les informations données par l'utilisateur et les insérer dans un visuel HTML soumis à une requête DAX. La taille, le texte et la couleur sont des **paramètres**, c'est-à-dire qu'ils prendront n'importe quelle valeur qu'on leur attribuera.

Pour les **graphiques**, c'est plus compliqué. Mais de manière simple, nous chargeons d'un côté les données [Data base] et de l'autre, les tables remplies par l'utilisateur dont celles concernant le premier graphique [Excel], par exemple. Ensuite, toujours via Power Query, nous créons une table qui met en en-tête de colonne la seule et unique ligne du tableau contenant les informations de personnalisation du graphique 1.

Finalement, nous fusionnons la base de données et cette dernière table artificiellement créée sur la base des noms de colonnes. À l'aide d'un **script R et de paramètres**, nous récupérons uniquement les colonnes voulues et les nommons "X axis" et "Y axis".

Ce seront ces deux champs qui seront insérés dans le visuel qui sera associé au graphique 1. Nous avons donc deux champs fixes attribués au graphique, mais dont les données sous-jacentes peuvent changer selon les besoins de l'utilisateur à travers le fichier de personnalisation [Excel].

- **Load** : Lors du chargement de nos données dans le modèle Power Query, et en se basant sur un index artificiellement créé au sein de nos tables (pendant l'étape de transformation des données), des relations entre les différentes tables se forment automatiquement.

5. La relation [Interact] entre le User et Power BI :

L'utilisateur utilise et interagit avec les visuels, filtres et la solution de visualisation offerte par Power BI.

Après la construction de cette solution technique, nous pouvons discuter de ses limitations en termes de paramétrisation, mais cela sera abordé dans la section dédiée (**Section 10 Discussion - Limitations - Future Works**). Tout comme nous venons d'aborder notre solution précise, l'idée générale et la solution scalable seront également discutées dans la section dédiée aux résultats (**Section 9 Résultats**).

Ici, nous pouvons visualiser une "vue d'ensemble" par un diagramme de flux de séquence de notre solution selon les règles BPMN (Figure : 17).

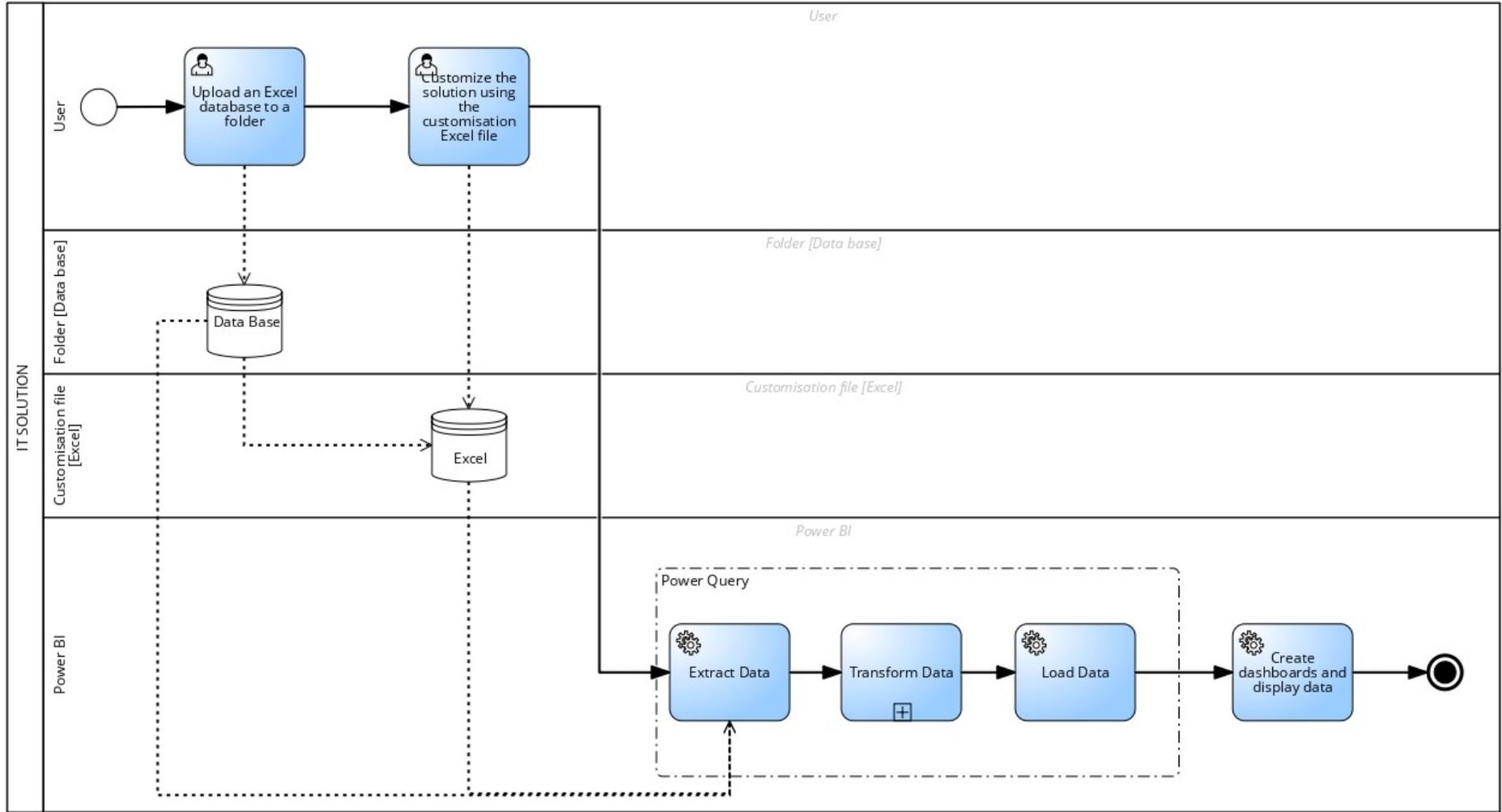


Figure 17: Diagramme de séquence : ETL

Table 3: Exigences pour l'ETL par type de personne, techniques (= cycle) et priorité

ID	NL Requirements about ETL	Cycle	Technique	Who?	Priority (1-5)
1	La solution doit pouvoir extraire des données à partir d'un fichier fourni par l'utilisateur.	1	BS	Pro - Aca	1
2	Elle doit extraire des données d'une table d'un seul fichier.	1	BS	Pro	1
3	La solution doit être capable d'extraire, de transformer et de charger des données avec une structure de données différente (différents noms de colonnes, différents nombres de colonnes, ...) tout en respectant le requirement 2.	1	BS	Pro - Aca	1
4	La solution doit extraire et orchestrer les données provenant de la source de données et du fichier de personnalisation, afin de les transformer et de les charger dans la solution de visualisation BI.	1	BS	Aca	1
5	La solution doit commencer par s'axer sur des données d'un domaine spécifique (par exemple, les ventes).	1	Sketching	Pro	1
6	La solution doit être capable d'exécuter le processus ETL automatiquement, c'est-à-dire sans assistance humaine.	1	BS	Pro	1
7	La solution ne doit pas nécessiter la connaissance et le respect de règles de gestion spécifiques pour garantir une intégration correcte de la solution.	1	BS	Pro	1
8	Elle devrait permettre le chargement incrémental des données, c'est-à-dire ne mettre à jour que les nouvelles données ou les données modifiées depuis la dernière exécution.	1	BS	Pro	1
9	L'utilisateur doit pouvoir choisir de ne pas charger l'intégralité de la base de données source, mais peut préfiltrer les données à inclure dans l'ETL en utilisant la page de personnalisation (en sélectionnant un ensemble de données parmi la base de données source).	2	Prototyping	Pro	3
10	Elle doit permettre de télécharger les données filtrées depuis le dashboarding vers une feuille Excel.	2	Prototyping	Pro	3
11	Elle doit être capable de convertir une devise vers une autre devise en fonction du choix de l'utilisateur (parmi une bibliothèque de devises enregistrées).	3	WOZ	Pro	3
12	Elle devrait extraire des données de plusieurs tables d'un seul fichier.	2	Prototyping	Aca	4
13	Elle devrait être capable d'effectuer des opérations de transformation sur les données, telles que le regroupement, le filtrage, l'agrégation, la normalisation, et tout autre processus de transformation indépendamment de jeux de données.	1	BS	Aca	4
14	La solution devrait être flexible pour tout type de source de données (BD relationnelle, fichier plat, ...).	1	Sketching - BS	Pro - Aca	4
15	La solution devrait permettre de charger les données transformées dans un entrepôt de données ou une base de données adaptée au stockage BI.	1	Sketching - BS	Pro - Aca	4
16	Il devrait offrir une interface conviviale permettant à l'utilisateur de placer et de sélectionner son ensemble de données.	1 - 3	BS - WOZ	Pro - Aca	5
17	Elle devrait prévoir des mécanismes de traitement des erreurs et de notifications en cas de défaillance ou d'anomalie.	1	BS	Pro	5

7 Phase 2 – Comment établir le contenu du dashboard?

7.1 Rigor Cycle

“Les KPI sont des métriques quantifiables et stratégiques qui mesurent les facteurs critiques de succès d’une organisation.” (Brooks (2005)).

Les **indicateurs de performance clés (KPI : Key performance Indicator)** sont des outils de reporting traditionnels permettant aux gestionnaires d’afficher des statistiques pour surveiller les performances de leur entreprise et faciliter la prise de décisions. Ils sont des mesures complexes qui permettent d’estimer l’efficacité d’une entreprise dans la réalisation de ses activités et de contrôler les performances de ses processus et de ses stratégies commerciales. Les KPI sont généralement normalisés et comparés à des valeurs cibles, seuils et minimales pour faciliter l’interprétation. Ils peuvent être classés en fonction de l’industrie ou du domaine fonctionnel, de la dimension temporelle, de la mesure des caractéristiques de l’intrant ou de l’extrait d’un processus, et comme qualitatifs ou quantitatifs, stratégiques ou opérationnels (Vaisman and Zimányi (2014), p.363).

Un élément important à relever ici est la phrase : “Ils peuvent être classés en fonction de l’industrie ou du domaine fonctionnel” (sujet discuté par Stoop (2009)). Comme nous pourrions le voir dans la **Section 7.2.2 Documentations** (Table : 5, Requirements 8-9), notre lead user “Pro” a suggéré, en se basant sur son expérience, de se concentrer d’abord sur un domaine fonctionnel afin de définir un périmètre et un environnement pour faire évoluer notre solution. En effet, il sera plus simple pour notre POC de regrouper l’ensemble des KPI du domaine des “ventes” (**domaine fonctionnel choisi pour notre solution**), plutôt que de reprendre ceux du point de vue financier, de la chaîne d’approvisionnement, etc.

Vaisman and Zimányi (2014), p.364-366 proposent des guidelines pour définir des indicateurs de performance utiles pour une organisation. Une fois que les sources ont été identifiées, il est possible de suivre plusieurs étapes pour définir les indicateurs. Pour être utiles, les indicateurs doivent remplir certaines conditions : ils doivent être spécifiques et non ambigus, clairement détenus par un département ou un service de l’entreprise, mesurables et produits en temps voulu. Eckerson (2010), p.197-222 présente aussi une section sur “Comment créer des indicateurs de performance (et métriques) efficaces” avec les différents types de KPI et métriques, les attributs d’un KPI (déjà présentés plus haut), les types de cibles d’un KPI et les caractéristiques d’une mesure de performance efficace.

Un KPI peut posséder cinq propriétés, notamment la valeur réelle, l’objectif, le statut, la tendance et le poids (Figure : 18) selon Vaisman and Zimányi (2014) et approfondi par Eckerson (2010).

Name	Trend	Status	Target	Actual	Variance %
Finance					
Decrease European Discounts	+	🟢	16.00%	15.53%	-3%
Keep APAC discounts < 17%	-	🟡	17.00%	24.77%	31%
Keep N.A. discounts < 17%	-	🟢	17.00%	15.66%	-9%

Figure 18: Displaying Performance Metrics : Eckerson (2010)

Une “métrique” (Hauser and Katz (1998)), quant à elle, est une mesure de l’activité de l’entreprise. Constituée d’une mesure et d’une dimension, elle indique ce qui est mesuré et par rapport à quoi. Par exemple, les mesures telles que “ventes par mois” et “retours par semaine” sont exprimées en termes de montant en dollars ou en quantité, tandis que les dimensions correspondent à la période spécifique, soit la semaine ou le mois. Les métriques sont les mesures sur lesquelles les KPI sont contrôlés et gérés (Stoop (2009)). En résumé un KPI est une mesure mais toutes les mesures ne sont pas des KPI.

Le travail de Stoop (2009) a été une source importante pour notre travail (nous pourrions le constater dans la **Section 8.2 Relevance Cycle** par le fait que ce dernier tente de répondre à une partie de notre problématique initiale **Section 2 Problématique**) car il présente un modèle puissant qui permet

la (semi) automatisation de la conception des indicateurs de performance clés et des tableaux de bord, notamment dans le domaine des **ventes et du marketing (même domaine que pour notre conception)**.

Le rapport de [Stoop \(2009\)](#) offre une structure claire et solide pour notre projet, en couvrant les aspects suivants :

- 1. Identification des KPI les plus utilisés en Ventés and Marketing.
- 2. Classification des KPI en fonction de leurs caractéristiques spécifiques.
- 3. Définition mathématique des KPI et étude de leurs relations.
- 4. [...]
- 5. Relation entre les KPI, le type d'entreprise et les rôles fonctionnels.
- 6. Différences dans la visualisation des données/KPI entre les ventes, le marketing et les autres départements.

Dans cette section, nous répondrons à ces questions selon [Stoop \(2009\)](#) au travers de l'étude de la littérature et de sa propre étude. Ce dernier propose que le type de mesure d'un **KPI** soit lié à la **combinaison** de la **mesure** et de la **dimension**. Tel qu'une **mesure** peut être un **flux** (ex : une période), un **stock** (ex : une valeur) ou une **unité** (ex : un coût par unité). Il classe ensuite les KPI et les mesures en fonction des quatre perspectives du BSC ([Kaplan and Norton \(1996\)](#)) :

- **Client** : Valeur et résultats de la proposition de valeur au client.
- **Interne** : Performance des processus, l'innovation, les opérations et après-vente.
- **Apprentissage et croissance**: Les compétences et les capacités internes de soutien.
- **Finances** : "Résultat" des trois autres perspectives, résultats des opérations et de la stratégie de l'organisation.

Il classe le type d'activité d'une entreprise de trois manières différentes :

- **Le type de clients** : Business-to-Consumer (B2C) ou Business-to-Business (B2B).
- **La multiplicité de ses clients** : One-to-One (O2O) ou One-to-Many (O2M).
- **Le type de ses produits** : Service (produits intangibles) ou non service (produits tangibles).

Ensuite, il a relié les KPI aux rôles fonctionnels ([Eckerson \(2006\)](#)). Dans cette thèse, le "**rôle fonctionnel**" désigne les tâches spécifiques d'un employé dans le cadre d'un certain travail. Les différents rôles sont présentés dans [Stoop \(2009\)](#), p.11.

La propriété "**leading / lagging**" d'un KPI, indiquant s'il est utilisé pour surveiller un processus en cours ou pour mesurer le résultat d'un processus terminé, a été mise de côté par l'auteur mais pourrait être un sujet intéressant à approfondir dans de futurs travaux (**Section 10 Discussion - Limitations - Future Works**).

Finalement, la relation intra KPI a été abordée afin de définir si un KPI est influencé directement par un autre KPI. La relation peut être à sens unique ou bidirectionnelle et caractérisée par une propriété **Casual, Indicator ou Unclear** qui représente respectivement une influence directe, indirecte et non définie.

Nous avons délibérément mis l'accent sur ces points, car en définissant ces critères, nous pourrions envisager de les incorporer dans notre solution (**Section 10 Discussion - Limitations - Future Works**). Cela permettrait aux utilisateurs de définir facilement des critères dans notre fichier Excel et la solution construirait des KPI adaptés à leurs besoins et à la situation commerciale de l'entreprise lors de la création d'un tableau de bord (**Section 7.3 Design Cycle**).

Grâce à cette étude, nous sommes en mesure de sélectionner les KPI (**Section 7.3 Design Cycle**) les plus pertinents pour enrichir notre POC sur le domaine fonctionnel des ventes et du marketing. Ces KPI ne seront donc pas choisis de manière arbitraire, mais en se basant sur l'étude préalable de [Stoop \(2009\)](#). Il faudra néanmoins faire attention aux éventuelles remarques et améliorations de ce dernier. En effet, cette étude n'est qu'une première analyse du problème et ne conduit pas à des conclusions formelles et solides. Plusieurs "biais" ont été révélés mais ont été conscientisés. Cependant, notre travail constitue également une première contribution dans le domaine et peut parfaitement s'appuyer sur cet article.

7.2 Relevance Cycle

Dans cette section, nous regroupons toutes les exigences perçues en termes de modélisation des KPI. Pour ce faire, nous avons suivi les conseils de [Pohl \(2016\)](#) (**Section 6.2 Relevance Cycle**).

7.2.1 Elicitations

Les éléments de cette section sont identiques à ceux discutés dans la **Section 6.2.1 Elicitations**, mais concernant les KPI.

7.2.2 Documentations

La méthodologie de documentation est reprise dans la **Section 6.2.2 Documentations**. Chaque ligne de ce tableau (Table : 5) représente une exigence, cette dernière est définie selon un ID, le NL requirements, le cycle lors de laquelle elle a été identifiée, la technique d'élicitations utilisée et sa source. La colonne "priority" nous indique l'ordre de priorité des implémentations des exigences dans une future itération de notre solution. Nous nous concentrerons uniquement sur les exigences ayant une priorité de 1 à 3, tandis que les exigences 4 et 5 seront abordées dans notre **Section 10 Discussion - Limitations - Future Works**.

7.2.3 Validation/négociation

La technique principale de validation est expliquée dans la **Section 6.2.3 Validation/négociation**.

7.3 Design Cycle

Introduite par la **Section 6.3 Design Cycle**, nous détaillerons ici le point de vue des KPI, un facteur important dans notre solution. Une fois de plus, nous discuterons ici de la solution précise que nous avons mise en œuvre pour illustrer nos propos et soutenir notre idée conceptuelle. Principalement soutenue par l'étude proposée par [Stoop \(2009\)](#), notre sélection de KPI n'a donc pas été faite au hasard.

Comme décrit dans la **Section 7.1 Rigor Cycle**, l'auteur a analysé les KPI du domaine du marketing et des ventes et les a classés en différentes catégories. Certaines n'ont pas été fructueuses, d'autres manquaient de profondeur dans les analyses, mais la **caractéristique principale** relevée a été le **type de client (B2B ou B2C)**. "*..., l'enquête a reçu beaucoup moins de réponses que prévu. Le One-to-One n'est pas du tout représenté dans le Business-to-Consumer.*" [Stoop \(2009\)](#), p.14. "*Les différences de KPI entre B2B et B2C ont été particulièrement soulignées. Sur la base de ces commentaires, nous avons choisi de différencier le type de client plutôt que tout autre type d'entreprise ou de département.*" [Stoop \(2009\)](#), p.17.

Suite à cela, il a réussi à sélectionner parmi une liste de KPI **un top 25 des KPI les plus importants dans les domaines du marketing et des ventes**, répartis selon qu'il s'agit de B2B ou de B2C. Parmi cette liste, nous avons arbitrairement choisi de sélectionner les **15 premiers** de chaque type (B2B et B2C)(Table : 4). Grâce à cela, nous avons laissé la possibilité à l'utilisateur de choisir parmi les 15 KPI les plus importants dans les secteurs du marketing et des ventes, en fonction de son type de client.

Le même processus pourrait être appliqué en fonction de la multiplicité de ses clients (O2O ou O2M), du type de ses produits (produits intangibles ou produits tangibles), du rôle fonctionnel, du caractère Leading ou Lagging, et bien sûr du domaine fonctionnel (ici marketing et ventes, mais cela pourrait également être appliqué aux finances, aux ressources humaines, etc.). Mais tout ceci sera envisagé comme un "**Future work**" de notre travail (**Section 10 Discussion - Limitations - Future Works**).

Les KPI choisis sont donc :

Table 4: Personal selection of the twenty-five most important key performance indicators in the B2B and B2C sectors : [Stoop \(2009\)](#)

KPI - B2B	SomeWhat + Very Important	KPI - B2C	SomeWhat + Very Important
Revenue	100%	Revenue	100%
Net Profit	100%	Market Growth	100%
New Customer Gain	100%	Gross Profit	100%
Gross Profit	96%	Return on Sales	100%
Customer Profitability	96%	Sales Complaints	100%
Market Demand	95%	Market Share	100%
Return on Sales	95%	Market Share Index	100%
Repeat Rate	91%	Product Complaints	100%
Net Price	91%	Delivery Complaints	100%
Market Penetration	91%	Net Profit	100%
Market Growth	91%	Market Demand	100%
Close Rate	87%	Net Price	100%
Close Rate	87%	Return on Customer	100%
Customer Loss	87%	Increased Sales	100%
Average Transaction Size	87%	Price Elasticity	100%

Afin d'être clair quant à leur définition et leur formule, [Stoop \(2009\)](#) donne pour chaque KPI ses deux caractéristiques que nous pouvons retrouver dans l'annexe [C](#).

De manière plus technique, comment fonctionne la solution?

L'ensemble de ces KPI ont été encodé au préalable dans notre solution. L'utilisateur doit d'abord définir le critère du type de client (B2B ou B2C), car comme nous l'avons vu, cela modifiera les meilleurs KPI à proposer en fonction de son environnement. Ensuite, l'utilisateur peut choisir son ou ses KPI parmi la liste proposée. Une formule pour le KPI sera également proposée à l'utilisateur pour construire son KPI en entrant ses variables (c'est-à-dire en sélectionnant parmi ses colonnes). Finalement, l'utilisateur pourra également définir son objectif à atteindre.

Nous avons choisi cette solution pour plusieurs raisons :

Premièrement, chaque entreprise possède sa propre base de données avec ses propres colonnes et noms de colonnes, ainsi que ses définitions de données. Il était préférable de laisser l'utilisateur construire son KPI pour garantir la qualité de la mesure calculée.

Une autre solution aurait pu être beaucoup plus intéressante grâce au texte mining, c'est-à-dire qu'en fonction du KPI choisi par l'utilisateur, une analyse de la base de données aurait été effectuée pour construire le KPI avec les bonnes features (nous pouvons retrouver une idée similaire dans la section [Section 8.3 Design Cycle](#) et dans notre [Section 10 Discussion - Limitations - Future Works](#)).

Pour l'insertion de ces informations dans nos visuels, nous retournons vers la [Section 6.3 Design Cycle](#).

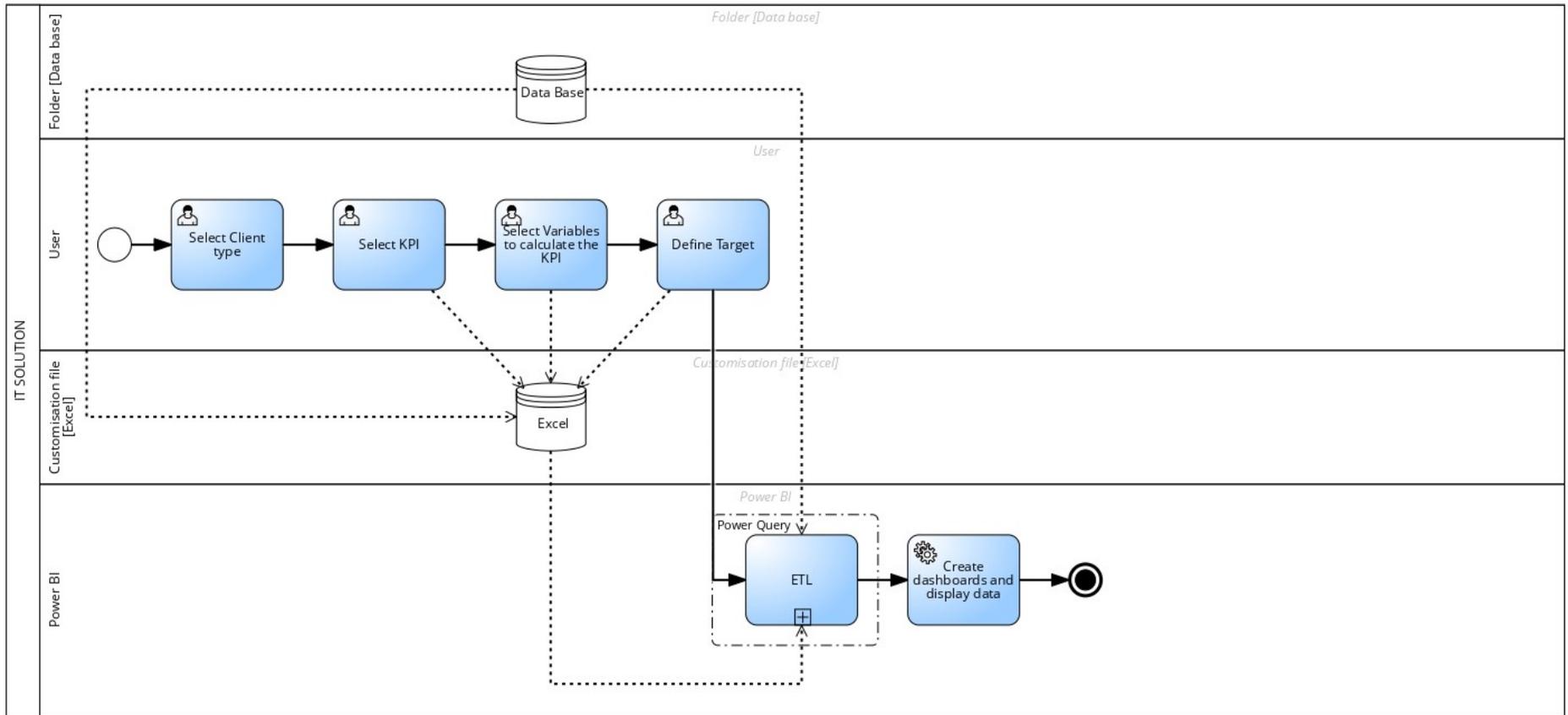


Figure 19: Diagramme de séquence : KPI

Table 5: Exigences pour les KPI par type de personne, techniques (= cycle) et priorité

ID	NL Requirements about KPI	Cycle	Technique	Who?	Priority (1-5)
1	La solution doit permettre à l'utilisateur de sélectionner les KPI pertinents pour son analyse.	1	BS	Pro - Aca	1
2	Elle doit avoir une bibliothèque de KPI prédéfinis pour faciliter la sélection.	1	BS	Pro - Aca	1
3	La solution doit être capable de calculer les KPI à partir des données fournies par l'utilisateur.	1	BS	Pro	1
4	La solution doit permettre de présenter les KPI de manière claire et conviviale dans les tableaux de bord.	1	BS	Pro - Aca	1
5	La solution doit mettre à jour automatiquement les KPI en fonction des nouvelles données fournies par l'utilisateur.	1	BS	Pro	1
6	La solution doit permettre à l'utilisateur de définir des objectifs à atteindre pour ses KPI.	1	BS	Pro	1
7	La solution ne doit pas nécessiter la connaissance et le respect de règles de gestion spécifiques pour la construction d'un KPI.	1	BS	Pro	1
8	La solution doit commencer par se concentrer sur les KPI d'un domaine fonctionnel précis.	1	Sketching	Pro	1
9	La solution doit regrouper les KPI principaux du domaine fonctionnel. <i>Verbatim</i> : "... Les indicateurs principaux de mon domaine fonctionnel" ET "Créer une liste d'indicateur par module (=domaine fonctionnel)"	1	Sketching	Pro	1
10	La solution doit permettre à l'utilisateur d'associer aux KPI une unité de mesure (poids, monétaire, ...).	2	Prototyping	Pro	2
11	La solution doit fournir un modèle à l'utilisateur pour saisir son KPI de manière simple et claire.	2	Prototyping	Aca	2
12	La solution doit afficher un pourcentage d'atteinte par rapport à l'objectif et le KPI.	3	WOZ	Pro	3
13	Elle doit également permettre à l'utilisateur de définir des KPI personnalisés en fonction de ses besoins spécifiques.	1 - 3	BS - WOZ	Pro - Aca	3
14	Elle devrait offrir différents types de visualisations adaptées à chaque KPI, tels que des graphiques, des jauges, des indicateurs clés, des cartes, etc.	1	BS	Pro	4
15	La solution devrait commencer par se concentrer sur les besoins d'un rôle fonctionnel précis.	1	Sketching	Pro	4
16	Elle devrait également prendre en compte les relations spécifiques liées à chaque KPI.	1	BS	Pro	5
17	La solution pourrait évoluer vers d'autres domaines fonctionnels que l'utilisateur pourrait sélectionner ce qui modifiera les KPI en fonction.	1	Sketching	Pro	5
18	La solution pourrait évoluer vers d'autres rôles fonctionnels que l'utilisateur pourrait sélectionner ce qui modifiera les KPI en fonction.	1	Sketching	Pro	5
19	La solution doit être capable de calculer les KPI choisis par l'utilisateur uniquement sur base de la base de données fournie par l'utilisateur. (Texte mining : Section 10)	2	Prototyping	Aca	5
20	La solution pourrait permettre à l'utilisateur d'ajouter autant de KPI qu'il le souhaite, tout en tenant compte de la lisibilité du tableau de bord.	2	Prototyping	Aca	5

8 Phase 3 – Comment établir les visuels du dashboard?

8.1 Rigor Cycle

” Un tableau de bord est un affichage visuel des informations les plus importantes nécessaires à la réalisation d’un ou de plusieurs objectifs, consolidées et disposées sur un seul écran de manière que les informations puissent être contrôlées d’un seul coup d’œil.” (Few (2006)).

Les **tableaux de bord (dashboard)** sont des outils de visualisation très populaires en veille stratégique. Ils regroupent des indicateurs clés de performance et autres graphiques pour permettre aux managers de mesurer, contrôler et gérer efficacement leurs performances/objectifs (Few (2006)). Les tableaux de bord sont utilisés pour visualiser les données organisationnelles et pour identifier les mesures à mettre en œuvre/surveiller à tous les niveaux de l’organisation. Grâce aux tableaux, les managers peuvent comparer les valeurs actuelles aux performances passées et aux objectifs futurs (Vaisman and Zimányi (2014), p.370-378 et Stoop (2009)).

Il existe trois types de tableaux de bord (Table : 6) :

- Les tableaux de bord **stratégiques** sont utiles pour avoir une vue d’ensemble de la performance (passée) de l’entreprise par rapport aux objectifs stratégiques et aident à prendre des décisions à long terme. Les données qu’ils fournissent peuvent être qualitatives ou quantitatives (Ex : (BSC) ou Six Sigma).
- Les tableaux de bord **opérationnels**, en revanche, permettent de surveiller les activités en temps réel (processus opérationnels de base) et nécessitent une interface simple pour permettre une identification visuelle rapide des mesures qui s’éloignent des objectifs et nécessitent une action immédiate.
- Les tableaux de bord **tactiques** sont utilisés pour optimiser les processus opérationnels de chaque service (finance, ventes et marketing, RH, ...) et fournir une vue d’ensemble de l’entreprise. Basés sur une plateforme BI, ils permettent aux responsables d’analyser les performances par rapport aux objectifs, en se basant sur des outils standard de reporting et d’analyse. Ces tableaux de bord affichent généralement une douzaine d’indicateurs clés de performance (ICP) et ressemblent à un portail de métriques avec des graphiques et des tableaux analytiques. Leur objectif principal est de fournir une vision du passé et d’aider les utilisateurs à identifier des axes d’amélioration.

Table 6: Main Characteristics of Performance Dashboards : Eckerson (2010).

	Operational	Tactical	Strategic
Purpose	Control operations	Optimize processes	Manage strategy
Scope	Operational	Departmental	Enterprise
Users	Staff+	Managers+	Executives+
Primary activity	Act	Analyze	Review
Focus	Current	Past	Future
Data refresh	Daily/Intraday	Daily/Weekly	Monthly/Quarterly
Information	Detailed	Detailed/Summary	Summary
Architecture	Core systems	Data warehouse	Excel or data mart
Metrics	Drivers	Drivers/Outcomes	Outcomes
“ Looks like a ... ”	Dashboard	Metrics Portal	Scorecard

Vaisman and Zimányi (2014) définissent des guidelines à suivre pour construire des tableaux de bords. La conception d’un tableau de bord efficace implique la sélection minutieuse des éléments visuels et des interactions pour répondre aux besoins des utilisateurs. Des facteurs tels que l’emplacement, l’attention, la charge cognitive et l’interactivité (Eckerson (2010) et Stoop (2009), p.47) contribuent grandement à l’efficacité d’un tableau de bord. Les informations importantes doivent être remarquées rapidement et le tableau de bord doit être conçu pour être vu d’un seul coup d’œil (1 écran, Few (2006)) sans avoir à faire défiler ou à naviguer sur plusieurs pages. Les tableaux de bord efficaces doivent répondre à 4 critères :

- Faciles à interpréter.
- Évitent une utilisation excessive de l'espace.
- Attrayants.
- Lisibles.

Eckerson (2010), p.223-250 examine comment concevoir l'aspect et la convivialité d'un tableau de bord de performance afin qu'il soit facile à utiliser et visuellement attrayant (Eckerson (2006), p.8 définit ce que ne doit pas être un dashboard). Pour cela il donne aussi des guidelines et conseils et finit par des exemples de templates classiques. Un template classique (Figure : 20) selon lui est composé de ces 5 caractéristiques :

- **Onglets** : Situés en haut de l'écran, ils regroupent les contenus fonctionnels associés. C'est donc une méthode appropriée pour différencier le contenu du tableau de bord en fonction des rôles des utilisateurs.
- **Filtres** : Placés dans une barre de navigation située dans les colonnes de droite ou de gauche de l'écran du tableau de bord, les filtres offrent aux utilisateurs la possibilité de modifier les données affichées dans un graphique. Ils représentent la forme la plus basique d'interactivité dans un tableau de bord en naviguant dans des hiérarchies ou en passant d'une dimension à une autre (**Section 5 Introduction à la BI**).
- **Texte** : Le texte est essentiel pour communiquer des informations importantes. Un texte statique et long prend de l'espace et peut encombrer l'écran. Il est préférable de créer un bouton permettant d'accéder à des informations supplémentaires plutôt que d'afficher un texte long directement sur le tableau de bord.
- **Polices de caractères** : Il est recommandé d'utiliser des polices faciles à lire telles que Times New Roman ou Arial et d'éviter les polices fantaisistes ou inhabituelles. Utiliser la même police dans l'ensemble du tableau de bord (taille 12). Une taille plus grande pour les onglets et autres. Les titres de la même taille que le texte principal, mais en gras. Les étiquettes des axes des graphiques peuvent également bénéficier d'un surlignage en gras.

Un bon thème regroupe l'ensemble de ces critiques assurant la fluidité et un équilibre des exigences de densité et d'éparpillement.

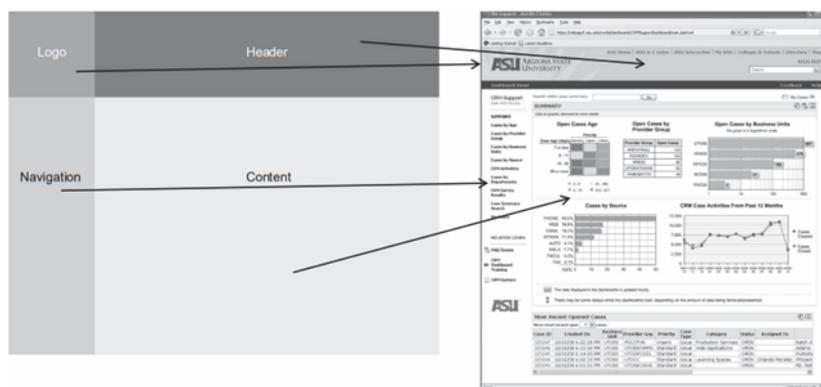


Figure 20: Classic template "Web" : Eckerson (2010)

La disposition des objets sur un tableau de bord est cruciale pour transmettre une histoire et une signification claire. Les éléments positionnés dans le quadrant supérieur gauche attirent l'attention principale, puis ce sont que les quadrants supérieurs droits et inférieurs gauches. En revanche, le quadrant inférieur droit est moins focalisé. Une des bonnes pratiques (Eckerson (2006), p.20-23) est également le regroupement des éléments similaires pour mettre en évidence leur relation et rapprochent ceux qui nécessitent une comparaison directe, facilitant ainsi la visualisation et la comparaison.

Nous avons discuté des tableaux de bord (le contenant) mais quand est-il du contenu, des graphiques et autres éléments analytiques présents dans ces tableaux de bord?

Eckerson (2010), p.240-248 dans sa section "Guidelines for design charts", nous dit que la plupart des principes décrits pour la conception d'un tableau de bord s'appliquent à la conception de diagrammes ou de graphiques. L'essentiel est de se concentrer sur les données (afin de communiquer le message clé des données sans le masquer) et non sur le graphique. Pour cela, il recommande de supprimer les éléments encombrants, tels que les quadrillages, les ombres, échelles, légendes, Par ailleurs, les étiquettes de données sont appréciées, car de nombreux utilisateurs préfèrent connaître les valeurs exactes des données plutôt qu'une approximation.

L'objectif d'un graphique est de nous permettre de "voir" la différence d'un seul coup d'œil (Few (2006)) et de faciliter les comparaisons visuelles. Forme, taille, saturation, marquages, couleur, largeur de ligne, orientation et position sont des attributs visuels permettant de faire ressortir des éléments du tableau de bord et attirer immédiatement l'attention des spectateurs. Par exemple, la couleur est une technique courante pour mettre en évidence les mauvaises performances d'un KPI. Les graphiques se présentent sous de nombreuses formes (Tufte (1983)) et tailles et le choix du bon type de graphique est un challenge crucial dans une bonne communication du message à délivrer aux travers des données Eckerson (2010), p.244-248 : "Choose the Right Graph".

Les KPI (définis à la section précédente **Section 7.1 Rigor Cycle**) sont utilisés pour montrer les données nécessitant une action et le code couleur doit être cohérent pour les différents KPI pour éviter la confusion (Few (2005)). Les outils distrayants, comme les mouvements et les animations, doivent être évités, tout comme les couleurs vives ou l'utilisation excessive de texte. Les tests utilisateurs peuvent aider à garantir la lisibilité du tableau de bord.

Comme l'indique Vaisman and Zimányi (2014), p330 : "*Les interfaces utilisateur graphiques sont nécessaires pour la découverte de connaissances, car ce sont souvent des utilisateurs non experts qui interagissent avec ces systèmes.*"

Nous pouvons retenir de cette citation la confirmation d'un réel un besoin de solution Self-service et user friendly pour les personnes n'ayant aucune ou presque pas de compétence métier (**Section 2 Problématique**).

Stoop (2009) énonce quant à lui aussi une partie de notre problème initial : "*Les fournisseurs d'outils de tableau de bord proposent des consultants chargés d'analyser les besoins d'information des utilisateurs et de délibérer sur les indicateurs clés de performance qui seront affichés dans le tableau de bord. Même si les futurs utilisateurs ont une idée claire de leurs besoins en matière d'information et que les consultants ont l'expérience de la création d'un tableau de bord adapté au type d'utilisateur, il peut s'avérer nécessaire de procéder à de nombreux ajustements avant que les bonnes informations ne soient affichées de la bonne manière.*". Cela soutient en partie à notre problématique initiale, comme expliqué dans la **Section 2 Problématique**.

8.2 Relevance Cycle

Dans cette section, nous regroupons toutes les exigences perçues en termes de modélisation des dashboards. Pour ce faire, nous avons suivi les conseils de Pohl (2016) (**Section 6.2 Relevance Cycle**).

8.2.1 Elicitations

Les éléments de cette section sont identiques à ceux discutés dans la **Section 6.2.1 Elicitations**, mais concernant les dashboards.

8.2.2 Documentations

La méthodologie de documentation est reprise dans la **Section 6.2.2 Documentations**. Chaque ligne de ce tableau (Table : 7) représente une exigence, cette dernière est définie selon un ID, le NL requirements, le cycle lors de laquelle elle a été identifiée, la technique d'élicitations utilisée et sa source. La colonne "priority" nous indique l'ordre de priorité des implémentations des exigences dans une future itération de notre solution. Nous nous concentrerons uniquement sur les exigences ayant une priorité de 1 à 3, tandis que les exigences 4 et 5 seront abordées dans notre **Section 10 Discussion - Limitations - Future Works**.

Dans cette section, nous mettrons en évidence les principaux types d'exigences non fonctionnelles (ENF) que nous supposons nécessaires pour définir l'attribut de qualité d'un système et la manière dont le système doit répondre aux exigences fonctionnelles (Pohl (2016), p.8).

- **Performance** : Elle doit offrir une expérience utilisateur interactive et fluide.
- **Serviceability** : Elle doit offrir une expérience utilisateur facile pour l'exploration et l'analyse des données.
- **Security** : La plateforme doit maintenir une garantie de conservation de tous les types de données et garantir la sécurité contre tout type de cyber-attaque.
- **Data Integrity** : Le programme doit garantir l'exhaustivité, l'exactitude, l'authenticité, la validité, l'absence d'altération des données et un format utilisable.
- **Manageability - Usability** : Le système répond à la facilité, à la rapidité et à la faible compétence pour découvrir, configurer, modifier.

8.2.3 Validation/négociation

La technique principale de validation est expliquée dans la **Section 6.2.3 Validation/négociation**.

8.3 Design Cycle

Le domaine du dashboarding est complexe et fortement dépendant du constructeur de la solution. Bien que Power BI favorise une approche de création autonome et personnalisée grâce à l'approche en Self-service, il reste néanmoins complexe, et cela ne répond pas à notre problématique de recherche. Notre approche principale a donc été de déplacer la logique métier hors de Power BI et de la traduire dans un outil plus convivial. Pour cela, nous avons développé un modèle de tableau de bord Power BI sous forme de template, où chaque élément visuel est personnalisable grâce à la paramétrisation et à l'attribution de variables (comme discuté dans la **Section 8.3 Design Cycle**, "mais à ce niveau-là je ne vous apprend plus rien"). Ce template regroupe les éléments clés et principaux que l'on retrouve dans toutes solutions BI : titres, logos, filtres, graphiques et autres éléments visuels tels que les KPI. Ces éléments sont à la fois fixes et variables :

- **Fixes**, dans le sens où leur position, leur taille, leur nombre, voir leur design, sont pré-définis et donc, pour l'instant, inflexibles en fonction des besoins de l'utilisateur. Cette limitation est principalement dûe à des contraintes techniques (et non conceptuelles) de Power BI (détaillées dans la **Section 10 Discussion - Limitations - Future Works**).
- **Variables**, dans le sens où les données (X et Y), les couleurs, les titres (taille, couleur, texte), le choix des graphiques, l'agrégation des données, les logos, etc., sont définis selon les préférences de l'utilisateur.

En raison des contraintes techniques spécifiques à Power BI, certains choix ont dû être faits de manière "arbitraire" pour gérer les attributs fixes. Par exemple, les types de graphiques disponibles pour l'utilisateur afin de représenter ses données, leur emplacement ou encore leur taille sont des choix prédéfinis. D'un côté, ces deux derniers éléments sont des caractéristiques que l'utilisateur final typique ne souhaite pas gérer (afin de répondre à notre problématique initiale). Pour résoudre ce problème, des solutions ont été identifiées lors des entretiens, mais elles n'ont pas été mises en œuvre pour des raisons techniques ou non cohérentes à l'état de POC mais nous pourrions les retrouver dans la **Section 10 Discussion - Limitations - Future Works**. Pour résumer, que trouvons-nous dans ce dashboard typique (Figure : 21)?

- 1 : Un titre
- 2 : Un logo
- 3 : Trois filtres
- 4 : Trois visualisations de type graphique (une grande, une moyenne et une petite)
- 5 : Quatre KPI



Figure 21: Exemple d'un dashboard construit grâce à notre solution (Image à titre d'exemple et ne représentant en aucun cas un cas d'utilisation précis).

Comme nous pouvons le voir, ce tableau de bord répond bien aux attentes et aux exigences énoncées à la **Section 8.2 Rigor Cycle** et mobilise les caractéristiques d'un template classique (Figure : 20). À travers ce dashboard, l'utilisateur pourra interagir, naviguer et filtrer ses données, tout comme avec une solution Power BI traditionnelle.

9 Résultats

Tout mémoire se doit d'avoir une section de résultats, cependant la nôtre ne sera pas aussi étendue que prévu. En effet, nous avons déjà abordé nos résultats à travers les trois cycles de conception (**Section 6.3, 7.3, 8.3 Design Cycle**) et nous pouvons affirmer avec certitude que nous avons réussi à proposer une solution innovante pour notre problème initial. Cette certitude est d'autant plus renforcée par nos dernières interviews (WOZ) avec nos deux types de "lead users". Ces derniers ont formellement confirmé que ce prototype répond à notre besoin de base et serait une base solide pour des travaux futurs.

Sur la base de notre proposition de solution et de la **Section 10 Discussion - Limitations - Future Works**, nous sommes en mesure de présenter une solution globale et évolutive à ce problème, afin de valoriser sa répliquabilité dans différents environnements.

Par exemple, il serait important de se détacher des outils utilisés. Car comme discuté dans la **Section 10 Discussion - Limitations - Future Works**, bien que les plateformes Microsoft aient été très utiles pour une solution simple, il serait plus conseillé de se tourner vers d'autres logiciels pour une construction plus professionnelle et réaliste, tout en conservant les mêmes relations et fonctionnalités mais en favorisant l'efficacité et l'efficacité.

De plus, nous avons construit l'intégralité de la solution de manière à ne dépendre d'aucun logiciel ou autre élément. C'est précisément ce qui confère à notre solution son caractère scalable et répliquable vers un avenir plus vaste. Premièrement, il serait important de maintenir une connexion entre l'interface de personnalisation et la base de données source, peu importe son type. Deuxièmement, un ETL trouvant sa source dans ces deux sources de données qui mènerait à la construction d'un modèle et, en fin de compte, une solution BI, principalement pour ses capacités d'interface.

L'idée principale à retenir est cette orchestration (Figure : 15) entre différents systèmes, dont l'utilisateur n'a même pas besoin de se préoccuper ou de paramétrer. D'un autre côté, la dualité entre la simplicité et la personnalisation rend la quête complexe : comment rendre la solution personnalisable sans devoir reconstruire entièrement Power BI ? Comment permettre à l'utilisateur de choisir où positionner ses visuels sans compliquer l'ensemble de la solution ? Tous ces éléments seront discutés dans la **Section 10 Discussion - Limitations - Future Works** et conduiront à de nouveaux travaux futurs.

Ces questions sont à la fois très intéressantes et stimulantes, car certaines parties relèvent parfois du travail d'une seule personne (comme les analystes ETL ou les développeurs web pour la page de personnalisation). Cependant, cette fois-ci, elles ne seront pas développées spécifiquement (en fonction de chaque client et de leurs besoins spécifiques), **mais de manière globale et unifiée, formant un ensemble cohérent dont la simplicité se ferait ressentir pour l'utilisateur mais dont la complexité métier lui serait cachée. N'est-ce pas là l'essence même du self-service ?**

"La plus grande difficulté réside dans la simplification d'une tâche complexe." Benito Giunta (PhD Researcher à l'université de Namur).

10 Discussion - Limitations - Future Works

Cette section nous permettra de discuter des **hypothèses** émises pour répondre à notre problème. Ensuite, nous aborderons également les **limitations**, principalement techniques, rencontrées. Enfin, nous traiterons de nos **futurs travaux**.

Premièrement, nos deux interviewés ont évoqué que dans ce problème, **deux sous-problèmes** pouvaient être abordés, mais seulement l'un d'entre eux pouvait être choisi pour être développé dans le cadre de ce mémoire. En effet, la discussion sur la création d'un **modèle de données variable/personnalisable (backend)** ou d'un **tableau de bord variables/personnalisables (frontend)** était deux interprétations de notre problématique. Cependant, nous avons décidé de nous **concentrer sur ce dernier** laissant la première idée en hypothèse (Figure : 8; importance du dashbaording dans le Self-service).

Une autre hypothèse a été de choisir **Power Query comme ETL** même si le choix de ce dernier peut être discuté car certains diront même que ce n'est pas réellement un ETL. Cela nous a contraint à de nombreuses limitations techniques, que nous avons pu ou non contourner. Cependant, Power Query nous a permis d'apporter une solution finale cohérente.

Une dernière hypothèse conceptuelle forte a été de nous tourner **initialement vers une base de données simples (une table plate telle qu'un fichier Excel)**. Même si cela est plus souvent utilisé que l'on ne le croit, c'était la meilleure hypothèse pour rester cohérent avec le reste de la solution et son niveau de détail.

Les **hypothèses techniques** ne seront pas abordées de manière exhaustive mais nous pouvons en citer quelques-unes. Par exemple, en raison des limitations de l'ETL choisi, **l'axe Y doit obligatoirement être une variable quantitative** (ce qui ne présente pas non plus une hypothèse forte dans le cas de l'analyse de données).

Un deuxième exemple concerne le fait que nous avons **défini arbitrairement** trois types de graphiques différents (camembert, histogramme et graphique linéaire) pour analyser les données. Une sélection devait être faite et nous avons opté pour ces trois graphiques en raison de leur présence courante dans tout tableau de bord. Le nombre de graphiques (idem pour les KPI et les filtres) et leur emplacement sur le tableau de bord sont également des hypothèses de notre part. Nous avons envisagé de personnaliser ces critères pour ces trois types d'éléments, mais nous avons rapidement rencontré des limitations.

Une dernière hypothèse concerne le fait que le graphique 3 (Figure : 21) doit contenir en Y des données de type date, en raison des limitations de l'ETL (encore une fois).

Quant aux **limitations**, elles étaient principalement techniques ou technologiques. Ces dernières ne rendent pas notre solution conceptuellement impossible, mais elles la rendent sujette à des travaux futurs plus complexes et aboutis. Parmi ces limitations, nous pouvons principalement citer la **difficulté inhérente à Power BI pour la construction de solutions variables**. En général, Power BI est utilisé dans un cadre "one shot", c'est-à-dire avec une source de données et une visualisation fixe, ce qui rend la variabilité pour les deux (source de données et construction du tableau de bord) à la fois complexe et délicate. Cependant, il est important de souligner que nous avons **confronté Power BI à ses limites techniques**. Par exemple, la possibilité de choisir précisément son type de graphe, de construire des suggestions de mise en page prédéfinie à proposer à l'utilisateur, etc., sont des idées intéressantes mais difficilement réalisables d'un point de vue technique avec Power BI. C'est pourquoi nous ne les avons pas oubliées, mais nous les avons plutôt abordées dans la section des travaux futurs qui suit.

Une autre limitation a été le choix de travailler avec Power Query comme ETL. Nous savons très bien que des ETL plus performants et efficaces sont disponibles sur le marché. En effet, nous avons dû jouer de ruse pour fusionner les données avec les choix de personnalisation de l'utilisateur. Encore une fois, nous sommes rapidement arrivés aux **limitations de cet "ETL"**. Par ailleurs, nous pouvons déjà confirmer que les étapes de transformation de données (type, ...), de gestion d'erreurs ou de valeurs manquantes seraient impossibles avec cette technologie, mais seraient beaucoup plus envisageables avec de vrais ETL (comme SSIS).

Finalement, nous pouvons présenter nos **futurs travaux**. Nous n'allons pas non plus les aborder un par un, car ceux-ci sont déjà présentés dans nos tableaux (Table : 3, 5 et 7) avec les priorités 4 ou 5. Cependant, les idées les plus intéressantes et évolutives seront présentées ci-dessous.

La plus grande contribution consisterait à ajouter la fonctionnalité de **Text Mining** (requirements 19 et 24, Table : 5 et 7). Une doctorante, Sarah Pinon, PhD Researcher à l'Université de Namur (Management of Information and Digital Transformation), mène une thèse en business intelligence, qui pour résumer de manière très simpliste, traite de la conversion de requêtes en langage naturel en requêtes techniques. La fusion avec cette idée innovante constituerait une contribution majeure à notre solution. En effet, en ajoutant une ontologie autour du dashboarding, il serait possible pour l'utilisateur d'exprimer ce qu'il souhaite, d'en sélectionner les éléments importants et de revenir à notre idée conceptuelle de base pour la suite.

Pour la suite, nous pouvons envisager principalement, d'un point de vue technique :

- Travailler avec plusieurs sources de données et de types plus complexes (plusieurs tables, bases de données relationnelles).
- Construire un modèle multi-dimensionnel plus adapté à l'analyse BI.
- Développer une interface plus conviviale pour la page de personnalisation.
- Permettre à l'utilisateur de choisir précisément le type et le nombre de graphiques, de filtres et de KPI qu'il souhaite inclure.
- Introduire des mises en page prédéfinies en fonction des exigences enregistrées (similaire à ce que Power Point peut proposer).

L'utilisation de logiciels plus professionnels (comme SSIS pour l'ETL) et flexibles pour l'insertion de commandes personnalisées permettra une plus grande liberté de développement et offrira à l'utilisateur davantage de possibilités de personnalisation.

D'un point de vue conceptuel :

- Se diversifier vers d'autres domaines conceptuels.
- Se diversifier vers d'autres types de tableaux de bord (Table : 6).
- Effectuer une étude similaire à celle de [Stoop \(2009\)](#) afin d'approfondir les résultats et définir des critères (domaine fonctionnel, rôle fonctionnel, multiplicité de clients, type de produit, etc. **Section 7.1 Rigor Cycle**) qui permettront d'affiner la solution vers un dashboard propice à l'environnement du client.
- Travailler avec des logiciels plus complexes et créer des solutions personnalisées (développement de visuels/graphiques variables en fonction du paramètre (type de graphique) passé en argument) pour éviter les contraintes fournies par les plateformes standards (ici : Microsoft).

11 Conclusion

Dans un premier temps, nous avons ancré notre mémoire dans la tendance du **Self-service** (**Section 1 Introduction**), qui se réfère aux *”équipements de l’environnement BI permettant aux utilisateurs de devenir plus autonomes et moins dépendants de l’organisation informatique”* Imhoff and White (2011).

Ensuite, nous avons exposé notre **problématique** (**Section 2 Problématique**), qui trouve sa source principalement dans l’étude menée par Analytics (2015). Cette étude met en lumière les réalités sous-jacentes à cette tendance du Self-service dans le domaine de la business intelligence. Selon les résultats, la quasi majorité des répondants considèrent qu’il est nécessaire d’accéder facilement aux données sans avoir recours aux services informatiques ni aux *”Power users”*. Cependant, les méthodes de Self-service actuellement proposées ne remplissent pas ces objectifs, car un peu plus de la moitié des répondants continuent de s’adresser aux *”Power users”* de l’IT. Cela a un impact sur la majorité des projets qui ne sont pas exécutés totalement en Self-service, contrairement à ce qui était prévu. Pourquoi cela se produit-il ? C’est en grande partie dû au fait que les outils de Self-service restent complexes (selon un tiers des répondants), ce qui conduit un grand nombre d’entre eux à estimer que le Self-service ne leur est pas vraiment d’une grande aide.

Suite à cette problématique, nous avons imaginé une **solution conceptuelle** (**Section 3 Proposition de Solution**) et l’avons concrétisée par une **implémentation technique**. Cependant, afin de garantir la rigueur de notre mémoire, nous avons opté pour une **methodologie DSR** (**Section 4 Méthodologie**) que nous avons scrupuleusement suivie. Nous avons construit un artefact, justifié les notions que nous avons mobilisées à travers le cycle de rigueur, et présenté ce prototype à la fois dans un environnement professionnel et académique. En suivant les recommandations formulées par Pohl (2016), nous avons analysé leurs retours et suggestions d’amélioration pour en modifier notre artefact et entamer un nouveau cycle.

Cela nous a finalement conduit à une **solution** que nous avons défini sous deux aspects : un **aspect conceptuel** et un **aspect technique**. L’aspect conceptuel (**Section 9 Résultats**) nous a permis de comprendre le résultat global de notre travail et de fournir une base scalable en vue de la répliquabilité. Le point de vue technique (**Section 6.1, 7.1 et 8.1 Relevance Cycle**) a été discuté au travers de nos cycles de design, mettant en avant des points essentiels de notre solution (l’ETL, les KPI et les dashboards). Ces deux points ont été soumis à des hypothèses et confrontés à des limitations lors du processus de construction, aboutissant à des suggestions pour des travaux futurs (**Section 10 Discussion - Limitations - Future Works**). Ces éléments ouvrent la voie à une toute nouvelle aventure et offrent des pistes de solutions à nos limitations.

En conclusion, la tendance du Self-service n’a pas encore atteint son plein potentiel auprès des utilisateurs. Les solutions proposées sont souvent indécises, suffisamment complexes pour parfois nécessiter des compétences métier, mais trop restrictives pour être adoptées par un data scientist. Malheureusement, pour s’orienter complètement vers les utilisateurs non experts, il faut souvent renoncer à certaines fonctionnalités (comme le dit mon père, *”choisir, c’est renoncer”*). En effet, nous avons pu le constater lors de nos discussions et développements, l’ajout de certaines fonctionnalités peut rendre la solution exponentiellement plus complexe à mettre en œuvre et à utiliser.

L’ensemble de ces points clés nous a conduit à proposer une **orchestration** entre plusieurs logiciels; une base de données, un fichier de personnalisation, un ETL et une solution de BI pour le dashboarding. Cette approche permet à l’utilisateur d’être réellement autonome car **l’essentiel de la solution a déjà été créé** et peut varier (grâce à la parémétrisation) dans n’importe quel environnement. Il lui suffit de participer en fournissant des informations via des entrées et, grâce à cela, il peut construire sa propre solution. Plus besoin de connaissances métier approfondies, plus de récréation à partir de zéro, et économie de temps et d’argent à la clé. **En d’autres termes, du véritable self-service?!**

References

- Abad, Z. S. H., Moazzam, S., Lo, C., Lan, T., Frroku, E., and Kim, H. (2018). Loud and interactive paper prototyping in requirements elicitation: What is it good for? In *2018 IEEE 7th International Workshop on Empirical Requirements Engineering (EmpiRE)*, pages 16–23. IEEE.
- Abad, Z. S. H., Sims, S. D., Cheema, A., Nasir, M. B., and Harisinghani, P. (2017). Learn more, pay less! lessons learned from applying the wizard-of-oz technique for exploring mobile app requirements. In *2017 IEEE 25th international requirements engineering conference workshops (REW)*, pages 132–138. IEEE.
- Alasadi, S. A. and Bhaya, W. S. (2017). Review of data preprocessing techniques in data mining. *Journal of Engineering and Applied Sciences*, 12(16):4102–4107.
- Alpar, P. and Schulz, M. (2016). Self-service business intelligence. *Business & Information Systems Engineering*, 58:151–155.
- Analytics, L. (2015). 2015 state of self-service. logi analytics second executive review of self-service business intelligence trends.
- Anand, N. (2012). Application of etl tools in business intelligence. *International Journal of Scientific and Research Publications*, 2(11):1–4.
- Anand, N. (2014). Etl and its impact on business intelligence. *International Journal of Scientific and Research Publications*, 4(2):1.
- Brooks, M. (2005). Defining and measuring kpis and metrics. *Business Intelligence Journal*, 10(3):44.
- Burnay, C., Jureta, I. J., and Faulkner, S. (2014). What stakeholders will or will not say: A theoretical and empirical study of topic importance in requirements engineering elicitation interviews. *Information Systems*, 46:61–81.
- Cole, R., Purao, S., Rossi, M., and Sein, M. (2005). Being proactive: where action research meets design research. *ICIS 2005 proceedings*, page 27.
- Conn, S. S. (2005). Oltp and olap data integration: a review of feasible implementation methods and architectures for real time data analysis. In *Proceedings. IEEE SoutheastCon, 2005.*, pages 515–520. IEEE.
- Dresch, A., Lacerda, D. P., Antunes Jr, J. A. V., Dresch, A., Lacerda, D. P., and Antunes, J. A. V. (2015). *Design science research*. Springer.
- Dumas, M., La Rosa, M., Mendling, J., and A Reijers, H. (2013). *Fundamentals of business process management*. Springer.
- Eckerson, W. (2006). Deploying dashboards and scorecards. *The Data Warehouse Institute*, pages 1–24.
- Eckerson, W. (2012). Business-driven bi: Using new technologies to foster self-service access to insights. *Tableau Software*.
- Eckerson, W. W. (2010). *Performance dashboards: measuring, monitoring, and managing your business*. John Wiley & Sons.
- Few, S. (2005). Dashboard design: Beyond meters, gauges, and traffic lights. *Business Intelligence Journal*, 10(1):18–24.
- Few, S. (2006). *Information dashboard design: The effective visual communication of data*. O’Reilly Media, Inc.
- Gartner (2020). 2020 gartner magic quadrant: Power bi. 2020 Gartner Magic Quadrant — Power BI. Publisher: Gartner.
- Hainaut, J.-L. (2009). Bases de données. *Concepts, utilisation et développement, Vottem, Belgique, Dunod*.

- Hauser, J. and Katz, G. (1998). Metrics: you are what you measure! *European Management Journal*, 16(5):517–528.
- Hevner, A., Chatterjee, S., Hevner, A., and Chatterjee, S. (2010). Design science research in information systems. *Design research in information systems: theory and practice*, pages 9–22.
- Hevner, A. R., March, S. T., Park, J., and Ram, S. (2004). Design science in information systems research. *MIS Quarterly*, 28(1):75–105.
- Imhoff, C. and White, C. (2011). Self-service business intelligence. *Empowering Users to Generate Insights, TDWI Best practices report, TDWI, Renton, WA*.
- Inmon, W. H. (1995). What is a data warehouse. *Prism Tech Topic*, 1(1):1–5.
- Inmon, W. H. (2002). *Building the Data Warehouse, 3rd Edition*. John Wiley Sons, Inc., USA, 3rd edition.
- Järvinen, P. (2007). Action research is similar to design science. *Quality & quantity*, 41:37–54.
- Kaplan, R. S. and Norton, D. P. (1996). The balanced scorecard harvard business school press. *Boston, MA*.
- Kimball, R. and Ross, M. (2011). *The data warehouse toolkit: the complete guide to dimensional modeling*. John Wiley & Sons.
- Lennerholt, C., Van Laere, J., and Söderström, E. (2018). Implementation challenges of self service business intelligence: A literature review.
- linternaute (2023). Paramétrisation : Définition simple et facile du dictionnaire — linternaute.fr. [Accessed 12-08-2023].
- Mhon, G. G. W. and Kham, N. S. M. (2020). Etl preprocessing with multiple data sources for academic data analysis. In *2020 IEEE Conference on Computer Applications (ICCA)*, pages 1–5. IEEE.
- Peffer, K., Rothenberger, M., Tuunanen, T., and Vaezi, R. (2012). Design science research evaluation. In *Design Science Research in Information Systems. Advances in Theory and Practice: 7th International Conference, DESRIST 2012, Las Vegas, NV, USA, May 14-15, 2012. Proceedings 7*, pages 398–410. Springer.
- Pohl, K. (2016). *Requirements engineering fundamentals: a study guide for the certified professional for requirements engineering exam-foundation level-IREB compliant*. Rocky Nook, Inc.
- Rittel, H. W. and Webber, M. M. (1973). Dilemmas in a general theory of planning. *Policy sciences*, 4(2):155–169.
- Sarikaya, A., Correll, M., Bartram, L. R., Tory, M. K., and Fisher, D. (2019). What do we talk about when we talk about dashboards? *IEEE Transactions on Visualization and Computer Graphics*, 25:682–692.
- Simon, H. A. (1996). *The sciences of the artificial*. MIT press.
- Stoop, J. (2009). Developing a reference model for kpi and dashboard reporting in sales & marketing. B.S. thesis, University of Twente.
- The Forrester (2019). Total economic impact study microsoft power bi. Enterprise BI Platforms (Vendor-Managed), Q3 2019 report. Publisher: The Forrester.
- Tufte, E. R. (1983). The visual display of quantitative information graphics press. *Cheshire, Connecticut*, 6410.
- Vaishnavi, V. K. and Kuechler, W. (2015). *Design science research methods and patterns: innovating information and communication technology*. Crc Press.
- Vaisman, A. and Zimányi, E. (2014). Data warehouse systems. *Data-Centric Systems and Applications*.

- Vangenot, C. (2003). Datawarehouse. Technical report, Technical report, Ecole polytechnique fédérale de Lausanne.
- Walls, J. G., Widmeyer, G. R., and El Sawy, O. A. (1992). Building an information system design theory for vigilant eis. *Information systems research*, 3(1):36–59.
- Wik, M. (2015). Using the wizard-of-oz technique in requirements engineering processes: A trial in a tourism context.

A Annexe : BPMN 2.0 - Business Process Model and Notation

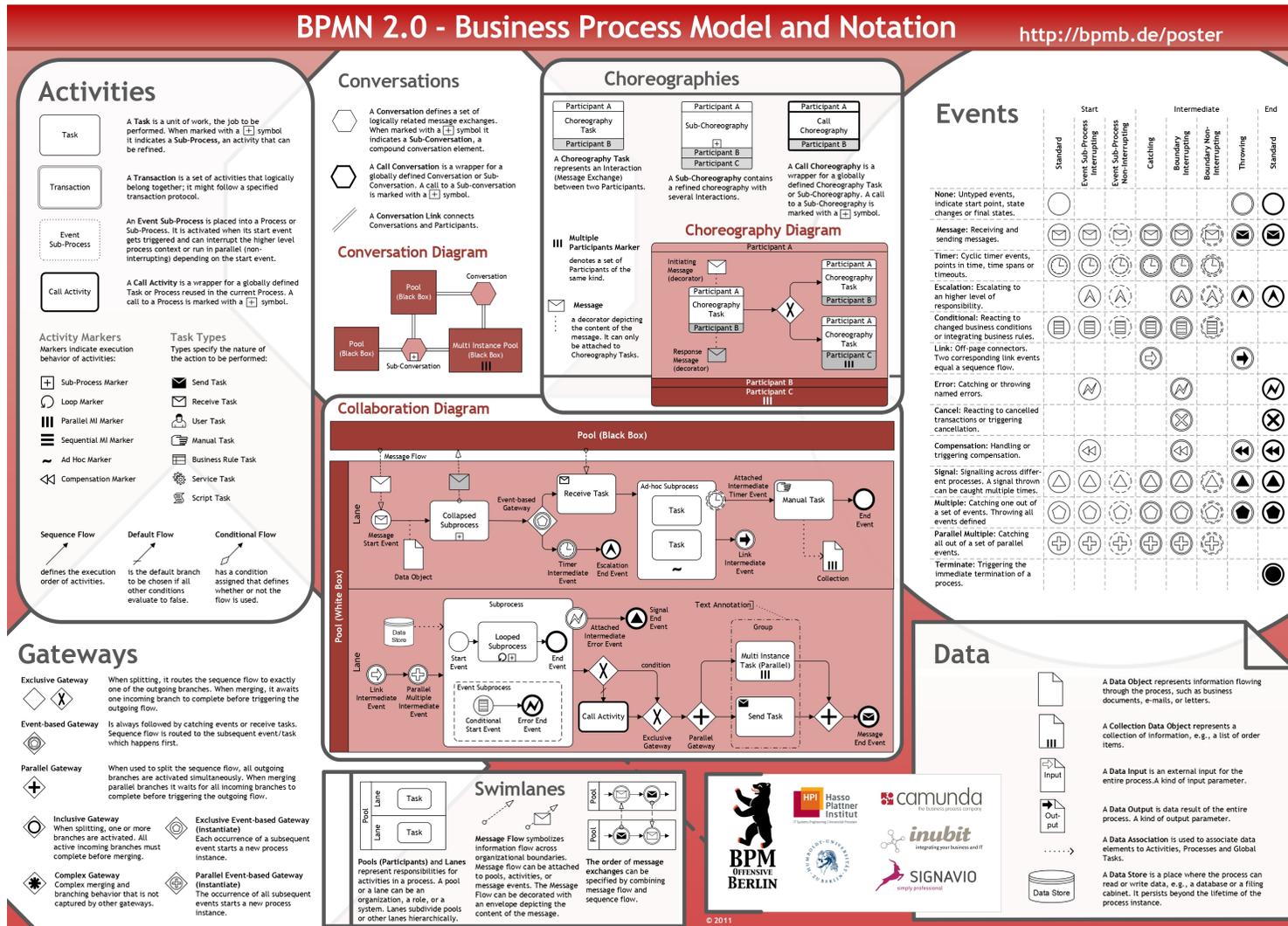


Figure 22: BPMN 2.0 - Business Process Model and Notation : <http://www.bpmb.de/index.php/BPMNPoster>

B Annexe : Documentation des exigences pour les dashboards

Table 7: Exigences pour les dashboards par type de personne, techniques (= cycle) et priorité

ID	NL Requirements about dashboard	Cycle	Technique	Who?	Priority (1-5)
1	La solution doit proposer le choix entre une sélection de graphes principaux.	1	BS	Pro - Aca	1
2	Elle doit avoir une bibliothèque de graphiques prédéfinis pour faciliter la sélection.	1	BS	Pro - Aca	1
3	La solution doit permettre à l'utilisateur de sélectionner les données à inclure dans un graphique.	1	BS	1	
4	La solution doit permettre à l'utilisateur de personnaliser l'apparence des graphiques en termes de couleurs, de légendes, de libellés, d'échelles, etc.	1	BS	Pro - Aca	1
5	La solution doit permettre aux utilisateurs d'interagir avec les tableaux de bord et les graphiques, par exemple en effectuant des zooms, des filtres, des sélections, des survols pour obtenir des informations supplémentaires, etc.	1	BS	Pro - Aca	1
6	La solution doit fournir des tableaux de bord où les éléments sont interconnectés et dynamiques entre eux.	1	BS	Pro - Aca	1
7	La solution doit prendre en charge des tableaux de bord qui se mettent à jour automatiquement en fonction des nouvelles données.	1	BS	Pro - Aca	1
8	La solution doit commencer par se concentrer sur des dashboards à destination d'un domaine fonctionnel précis. <i>Verbatim</i> : "Rapport type d'un contexte fonctionnel particulier"	1	BS	Pro	1
9	Elle doit faire en sorte que les filtres doivent être applicables à tous les visuels.	3	WOZ	Pro	2
10	Elle doit offrir des filtres interdépendants.	3	WOZ	Pro	2
11	Elle doit permettre la planification de l'actualisation des tableaux de bord à des intervalles réguliers ou sur demande.	1	BS	Pro	1
12	La solution doit permettre à l'utilisateur d'avoir une visualisation détaillée des données, par exemple sous forme de tableaux.	2	Prototyping	Pro	2
13	La solution de dashboard doit être capable d'accueillir de nouveaux éléments et pages créés de manière précise, sans avoir à passer par la personnalisation de page (via l'outil de visualisation).	2	Prototyping	Pro	2
14	La solution doit proposer à l'utilisateur de définir ses couleurs non pas par code HTML, mais de manière visuelle.	3	WOZ	Aca	3
15	La solution doit proposer à l'utilisateur de définir son logo par url ou en chargeant une image.	3	WOZ	Aca	3
16	La solution doit proposer l'insertion de données telles que l'auteur, le département ou d'autres informations pertinentes de l'environnement du dashboard final.	3	WOZ	Aca	3

Table 8: Exigences pour les dashboards par type de personne, techniques (= cycle) et priorité

ID	NL Requirements about dashboard	Cycle	Technique	Who?	Priority (1-5)
17	La solution doit proposer un résumé textuel compréhensible de ce que l'utilisateur a saisi comme entrées pour ses graphiques.	3	WOZ	Aca	3
18	La solution devrait commencer par se concentrer sur des dashboards à destination d'un rôle fonctionnel précis.	1	Sketching	Pro	4
19	La solution devrait donner la possibilité à l'utilisateur de pouvoir choisir la disposition de ses graphiques parmi une liste de propositions suggérées.	2	Prototyping	Aca	4
20	La solution devrait donner la possibilité à l'utilisateur de pouvoir choisir son graphique sur la page de customisation (Excel).	2	Prototyping	Pro - Aca	4
21	La solution devrait évoluer vers d'autres domaines fonctionnels que l'utilisateur pourrait sélectionner ce qui modifiera les dashboards en fonction.	1	Sketching	Pro	5
22	La solution devrait permettre à l'utilisateur d'ajouter autant de graphique qu'il le souhaite, tout en tenant compte de la lisibilité du tableau de bord.	2	Prototyping	Aca	5
23	La solution devrait évoluer vers d'autres rôles fonctionnels que l'utilisateur pourrait sélectionner ce qui modifiera les dashboards en fonction.	1	Sketching	Pro	5
24	La solution devrait être capable de proposer un graphique uniquement sur base de la base de données fournie par l'utilisateur. (Texte mining : Section 10)	2	Prototyping	Aca	5
25	La solution devrait pouvoir traiter et visualiser des données en temps réel, si nécessaire (stream).	1	BS	Pro	5
26	La solution doit offrir une variété de types de graphiques, tels que les graphiques à barres, les graphiques linéaires, les graphiques circulaires (camemberts), les diagrammes en cascade, les cartes géographiques, etc.	1	BS	Aca	5

C Annexe : KPI list definition and formula

KPI Name	Use	Details	Formula
Product Category Volume	Calculates the percentage share of store brand-sales in a certain product category.		$PCV = \text{sales}(\text{category_Brand_Stores}) / \text{sales}(\text{category_All_Stores})$
Revenue	Shows financial gain from production and sales of products and services.	Lead driver of performance measurement. Not usable without further refinement	$\text{Revenue} = \text{price} * \text{sales}(\text{Units})$
Gross Profit	Gross profit is calculated before operating expenses.	High gross profit can suggest good business execution.	$\text{Gross Profit} = \text{Revenue} - \text{costs}(\text{COGS})$
Value to Volume Ratio	Determines how efficient operating efforts are compared to competition	Scores below 1--% indicates needed improvements.	$VVR = \text{Market_Share}(\text{Gross_Profit}) / \text{Market_Share}(\text{Revenue})$
Net Profit	Shows total financial gain after subtraction of all costs	Can be calculated in multiple ways. Equals Earnings	$\text{Net_Profit} = \text{sales}(\text{Units}) * \text{Margin} - \text{costs}(\text{Total})$
Return on Sales	Measures efficiency of sales effort; the amount of profit relative to each sales-dollar/euro	Can vary significantly inter-/intra-industry.	$\text{ROS} = \text{Net_Profit} / \text{sales}$
Return on Customer Marketing Cost per Unit	Measures efficiency of customer investments. Measures how much marketers spend to market one unit.	Can be measured in different ways. Change in MCPU can indicate change in product lifecycle stage.	$\text{ROC} = \text{customer}(\text{net_Income}) / \text{customer}(\text{net_Investment})$ $\text{MCPU} = \text{costs}(\text{Marketing}) / \text{sales}(\text{Units})$
Program / Non-Program Ratio	Compares the amount of money spent on (marketing) activities to create value to the amount spent to support those activities.	Determining program/non-program expenses is the key problem. Low value can indicate too much overhead costs.	$\text{PNPR} = \text{costs}(\text{Marketing_Program}) / \text{costs}(\text{Total})$
Market Growth	Calculates the change in market aspects between periods.	Helps comparing company's performance to the performance of the rest of the market. Measuring different aspects gives different results in growth.	$\text{Market_Growth} = \Delta \text{sales}(t) / \text{sales}(t-x)$
Market Share	Measures the share of the company compared to the rest of the market, industry or product area.	Measures the succes of attempted market penetration. Must be compared to other (market) aspects.	$\text{Market_Share} = \text{sales}(\text{by_Company}) / \text{sales}(\text{total})$
Market Share Index	Helps identifying areas in need of improvent to increase Market Share.	Preference indicates product attractiveness to customers	$\text{Market_Share_Index} = \text{Awareness} * \text{Preference} * \text{Intention} * \text{Availability} * \text{Satisfaction}$
Net Sales Contribution	Helps identifying groups of customers/products based on needs and characteristics, so that the company can work and market more efficient.	Four rough segments; demographic, psychographic, geographic, product (use).	$\text{NSC} = \text{sales}(\text{Segment}) / \text{sales}(\sum \text{Segments})$
Market Demand	Helps measuring total demand in a period. Gives a basic insight in customer distribuion and marketing effectiveness.	Measurements must be solid.	$\text{Market_Demand} = \text{sales}(\text{Repeat}) + \text{sales}(\text{Trial})$

Net Price price(Net)	The actual price paid for a product by customers	Determining prices/discounts is subject to marketing strategy.	$Net_Price = price(Listed) - discounts$
Share of Voice	The percentage of advertising in a given product-market.	Message and medium fit can have equal or more impact then the amount of money spent.	$Share_of_Voice = \frac{costs(company_Advertising)}{\sum costs(market_Advertising)}$
Advertising to Sales Ratio	Calculates the effectiveness of advertising to create sales.	The lower the better, but varies greatly among industries.	$ASR = \frac{costs(Advertising)}{sales}$
Reach	Measures the percentage of a defined population reached through a certain media vehicle.	The amount of people reached does not equal the amount of people who actually consciously saw the ad.	
Impressions	The number of times and advertisement is viewed		
Frequency	Measures the average number of impressions per individual reached by an advertisement		
Gross Rating Points	The total of Impressions expressed as a percentage of a defined poulation	Can exceed 1--%. GRP varies through each product lifecycle stage. Quality of impression should be considered.	$GRP = Reach * Frequency$ $GRP = \frac{impressions}{customer(target_Population)}$
Cost per (Gross Rating) Point	Calculates the cost of one Rating Point.	Helps marketers determine the best advertisement medium.	$CPP = \frac{costs(Advertisement)}{GRP}$
Sales Premiums	The amount of promotional items given to future customers as incentives.		
Baseline Sales	Expected sales excluding the marketing programs under evaluation		
Incremental Sales	The (expected) sales of a certain marketing program		
Promotion Profit	Calculates the profit from promotional activities.	Estimating certain parameters can be hard.	$PP = \Delta sales(Discounted) * (Margin - price(Discount)) + (\Delta sales(Undiscounted) * Margin) - sales(base_Discounted) * price(Discount) - costs(Promotion) \pm carryover-effects$
Brand Equity Price Premium	The total value of a brand The relative price of a product compared to a benchmark price (average retail).	Closely related to pricing strategy.	$Price_Premium = price(Brand) - price(Average)$
Recall	Measures if consumers have knowledge of a recent ad.	Top of Mind/Dominance Aided/Unaided recall	
Recognition / Awareness	Measures if consumers have knowledge of a certain brand.	Values are highly dependent on industry.	
Transaction per Customer	Measures the amount of potential customers that convert to buying customers.	Equals Percentage Yield Rate Equals Walk to Buy Ratio	$TPC = \frac{customer(Transactions)}{customer(Traffic)}$
Returns to Net Sales	Measures the value of returned products compared to net sales.	Higher values can suggest poor quality/service.	$RTS = \frac{sales(Returned)}{sales(Net)}$

Inventory Turnover	Measures how often inventory turns over; sold completely and refilled.	High turnover is preferred; can indicate good sales performance. Very dependent on industry.	$\text{Inventory_Turnover} = \text{sales}(\text{Total}) / \text{inventory}(\text{Average})$
Percent Inventory Carrying Costs	Calculates the cost of carrying inventory compared to net sales.	Lower is better. Highly dependent on strategy.	$\text{PICC} = \text{costs}(\text{Inventory}) / \text{sales}(\text{Net})$
Sales per Square Foot	Measures retailer's effectiveness in generating sales compared to floorspace.	Can be a measure of store performance.	$\text{SPSF} = \text{sales}(\text{Total}) / \text{space}(\text{square_Feet})$
Sales/Profits per Employee	Measures the financial performance of the company on an employee basis.	Measures productivity; should be measured considering trends and seasonality.	$\text{SPPE} = \text{sales} / \text{employee}(\text{Total})$ $\text{SPPE} = \text{Net_Profit} / \text{employee}(\text{Total})$
Average Transaction Size	Calculates the average size or value of each transaction/purchase.		$\text{ATS} = \text{sales}(\text{Total}) / \text{purchase}(\text{Total})$
Retailer's Margin Percentage	Calculates the profit margin retailers earn.		$\text{RMP} = (\text{price}(\text{Consumer}) - \text{price}(\text{Wholesaler})) / \text{price}(\text{Consumer})$
Markdown	The amount of shop-floor reductions in price of a product		$\text{Markdown} = \text{price}(\text{Initial}) - \text{price}(\text{Actual})$
Markdown Goods Percentage	Measures the amount of products sold at markdown prices.	Helps retailers determine effectiveness of selling full-price products.	$\text{MGP} = \text{sales}(\text{net_Markdown}) / \text{sales}(\text{Net})$
Percent Utilization of Discounts	Measures the value of discounts compared to total sales.	Used by suppliers to increase the placement or volumes of (new) products.	$\text{PUD} = \text{discount} / \text{sales}(\text{Total})$
Sales Force Salary	Measures the amount of money the sales force is payed.	Can be commission based, bonus based, straight or otherwise implemented.	
Sales Performance Quota	Guides, motivates and benchmarks the sales force.	Unit volume quota Point quota system Profit quota Activity quota	
Average Sales per Call	Measures the Revenue per call.		$\text{SPC} = \text{sales} / \text{calls}$
Close Ratio	Calculates sales force performance by comparing closed sales to potential sales.		$\text{CR} = \text{sales}(\text{Closed}) / \text{sales}(\text{Potential})$
Cost per Call	Measures the cost per call.		$\text{CPC} = \text{costs}(\text{Total}) / \text{calls}$
Sales Productivity	Measures the productivity of the sales force.	Can be measured in sales per person, profits per person and volume per person.	$\text{SP} = \text{sales}(\text{Total}) / \text{employee}(\text{sales_Force})$
Sales Value Variance	Measures the difference in budgeted and actual sales.		$\text{SVV} = \text{sales}(\text{Actual}) / \text{sales}(\text{Budgeted})$
Sales Price Variance	Measures the difference in budgeted and actual price.		$\text{SPV} = \text{sales}(\text{Actual}) * (\text{price}(\text{Actual}) - \text{price}(\text{Budgeted}))$
Sales Volume Variance	Measures the difference in budgeted and actual volume sold.		$\text{SVV} = \text{price}(\text{Budgeted}) * (\text{sales}(\text{Actual}) - \text{sales}(\text{Budgeted}))$
Product Returns	Measures the amount of products returned to the company.		
Product Complaints	Measures the amount of complaints regarding product defects compared to the total units sold.		$\text{Product_Complaints} = \text{complaints} / \text{sales}(\text{Units})$

Delivery Complaints	Measures the amount of complaints regarding faulty, early or late deliveries.		$\text{Delivery_Complaints} = \text{complaints} / \text{deliveries}$
Sales Complaints	Measures the amount of complaints regarding pre-purchase-procedures.		
Off-time Deliveries	Measures the amount of early or late deliveries.	Different deviations in delivery times can be weighed to get a more accurate / meaningful impression of delivery status.	$\text{Off-time_Deliveries} = \text{deliveries}(\text{Off-time(Weighted)}) / \text{deliveries}(\text{Total})$
Faulty Deliveries	Measures the amount of faulty deliveries.	Different faults in deliveries (wrong item / missing item) can be weighted to get a more accurate / meaningful impression of delivery status	$\text{Faulty_Deliveries} = \text{deliveries}(\text{Faulty}) / \text{deliveries}(\text{Total})$
Adjusted Trial Rate Target Market Fit	Trial Rate considering ACV and Awareness. The percentage of customers who fit the predetermined descriptor set for the SKU.		$\text{ATR} = \text{Trial_Rate} * \text{Awareness} * \text{ACV}$
Cannibalization	Calculates the reduction in sales of existing products due to New Products.		$\text{Cannibalization} = \text{sales}(\text{existing_Lost}) / \text{sales}(\text{New})$
Quotation Speed	Measures the average amount of time it takes to reply to a quotation request.		
Net Promoter Score	Calculates the percentage of customers who would recommend the company to others.	Ask customers to rate (aspects of) the company; more than one interview question is better. promoters($x > 8$) detractors($x < 7$)	$\text{NPS} = \text{promoters} - \text{detractors}$
Complaint Resolval Time	Measures the amount of time it takes before customer complaints are fully resolved.		
Price Elasticity	Calculates the responsiveness in quantity demanded to a small change in price.		$\text{PE} = \Delta \text{sales}(\text{Units}) / \Delta \text{price}$
(Weighted) Numeric Distribution All Commodity Value	Measures the number of outlets carrying the brand. Calculates the percentage of sales in all categories that are generated by the stores that stock a given brand.	Can be weighted on various factors. Weighted measure of product availability based on total store sales. Does not consider specific SKU categorie-sales	$\text{ACV} = \text{sales}(\text{stores_Carrying_Brand}) / \text{sales}(\text{all_Stores})$
Category Perform Ratio	Calculates the relative performance of a retailer in a certain category compared with its performance in all categories.		$\text{CPR} = \text{PCV} / \text{ACV}$

Facings	Measures the amount of frontal views of a single product-package on a fully stocked shelf.	
Share of Shelf	Compares brand Facings to total Facings.	$SOS = \text{Facings}(\text{brand}) / \text{Facings}(\text{Total})$
Average Order Turnover	Calculates the average amount of turnover each order generates.	
Reservation Price	Measures the maximum price a customer is willing to pay for a product.	
Percent Good Value	Calculates the proportion of customers who perceive a product to represent a good value (below their Reservation Price).	
Pageviews	Measures the number of times a website is viewed.	
Clickthrough Rate	Calculates the percentage of impressions that motivate users to click on a link.	$CTR = \text{clicks} / \text{Impressions}$
Cost per Impression	Calculates the average costs of Impressions.	
Cost per Click	Calculates the average costs of Clicks.	
Cost per Order	Calculates the average costs of Orders.	