



**HAL**  
open science

## Développement spécifique ou progiciel ? Quels sont les critères de décision à prendre en compte dans le choix entre développement spécifique et progiciel ?

Marion Bourlier

### ► To cite this version:

Marion Bourlier. Développement spécifique ou progiciel ? Quels sont les critères de décision à prendre en compte dans le choix entre développement spécifique et progiciel ?. Gestion et management. 2014. dumas-01120093

**HAL Id: dumas-01120093**

**<https://dumas.ccsd.cnrs.fr/dumas-01120093>**

Submitted on 24 Feb 2015

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Mémoire de stage

## Développement spécifique ou progiciel ?

Quels sont les critères de décision à prendre en compte dans le choix entre développement spécifique et progiciel ?



Présenté par : **BOURLIER Marion**

Nom de l'entreprise : **Schneider Electric**

Tuteur entreprise : **LINOSSIER Jean-Benoît et SACILOTTO Sylvie**

Tuteur universitaire : **CARTON Sabine**

Master 2 Professionnel FI  
Master MSI option MSIO 2  
Spécialité Système d'Information et d'Organisation  
2013 - 2014



#### Avertissement :

L'IAE de Grenoble, au sein de l'Université Pierre-Mendès-France, n'entend donner aucune approbation ni improbation aux opinions émises dans les mémoires des candidats aux masters en alternance : ces opinions doivent être considérées comme propres à leur auteur.

Tenant compte de la confidentialité des informations ayant trait à telle ou telle entreprise, une éventuelle diffusion relève de la seule responsabilité de l'auteur et ne peut être faite sans son accord.

# REMERCIEMENTS

Je tiens à remercier dans un premier temps Jean-Benoît Linossier, mon tuteur de stage entreprise, pour son soutien ainsi que son suivi tout au long du stage.

Je remercie également ma tutrice de stage université, Sabine Carton qui m'a accompagnée dans toutes les étapes de mon stage.

Enfin je tiens à remercier toute l'équipe IS : Jean-Pierre Schweitzer et Sylvie Sacilotto mais aussi Elsa Giraud-Virissel, Emmanuel Berger, Franck Dervin, Virginie Carpena, Walter Mazin et Yves Langlais pour leur bonne humeur et leur accueil chaleureux.

# RESUMES

*Mots Clés : développement spécifique, progiciel, maquettage, processus décisionnel, solution informatique*

Dans le contexte d'un projet informatique ou un besoin informatique est identifié, la question du choix quant à la solution à mettre en place se pose : doit-on construire la solution ou faut-il adopter une solution toute faite ? Développement spécifique ou progiciel ? Il nous faut ainsi déterminer les critères de décision à prendre en compte dans ce choix.

En réalisant dans un premier temps un état de l'art, nous avons défini les principales thématiques existantes en termes de développement spécifique et progiciel.

La mission effectuée au sein de Schneider-Electric a permis une approche différente. Nous avons vu la démarche mise en place, la problématique de management qu'elle pouvait soulever, et les arguments, contraintes et avantages recensés.

Enfin dans le but d'établir une démarche à suivre quant la décision entre développement spécifique et progiciel, nous avons analysé les différents critères de décision et les processus décisionnels.

*Keys words: specific development, software package, modeling, decision making process, IT solution*

In a project where IT need is identified, we ask ourselves what's the ideal solution to implement: should we build the solution or does it take a ready-made solution? Specific development or software package? We must determine the decision criteria to consider when choosing between these two solutions.

First of all, by realizing a state of the art, we have managed to identify key themes existing in terms of development specific and software package.

With the mission at Schneider-Electric, we can observe a different approach. We saw the process developed, the issue of management, and the arguments, constraints and benefits that we could identified.

Finally, in order to establish a process to follow, we analyzed the various decision criteria and decision making.

# SOMMAIRE

Remerciements .....	3
Resumes.....	4
Table des illustrations.....	6
Avant-Propos.....	7
Introduction .....	9
1. Choix entre développement spécifique ou progiciel : la théorie .....	10
1.1. Définitions.....	10
1.1.1. Notion de logiciel et développement spécifique .....	10
1.1.2. Notion de progiciel.....	11
1.2. Etudes de cas, avantages et contraintes des deux solutions .....	11
1.2.1. Etude de cas : « Faut-il développer soi-même en PME ? ».....	12
1.2.2. Etude de cas : La vision de Martin Fowler.....	14
1.3. Identification des thématiques.....	16
2. L'analyse de terrain IS, à Schneider Electric.....	18
2.1. Le contexte de la mission.....	18
2.2. La démarche mise place .....	19
2.2.1. De l'analyse jusqu'au questionnement : développement spécifique ou progiciel ?....	19
2.2.2. Maquettage et démarche itérative.....	20
2.2.3. Démarche en terme de management .....	21
2.2.4. Résultats .....	23
2.3. Les critères de décision et arguments recensés .....	24
3. La confrontation de la théorie et de la pratique .....	27
3.1. Analyses et comparaison des arguments théoriques et pratiques.....	27
3.1.1. Les thématiques et arguments .....	27
3.1.2. Les processus décisionnels présentés .....	29
3.2. Proposition d'une démarche à suivre .....	32
Conclusion.....	34
Bibliographie.....	35
Annexes .....	36

# TABLE DES ILLUSTRATIONS

Figure 1 : Schneider Electric France - Entités.....	7
Figure 2 : Schneider Electric France - PEC.....	7
Figure 3 : Retro-planning .....	22
Figure 4 : Tableau récapitulatif des avantages et contraintes des deux solutions .....	26
Figure 5 : Processus décisionnel identifié par developpez.com .....	30
Figure 6 : Le processus décisionnel de l'expérience terrain à Schneider-Electric .....	31
Figure 7 : Une proposition de démarche à suivre.....	32
Figure 8 : Grille de critères.....	33
Figure 9 : Planning prévisionnel .....	37
Figure 10 : Tableau des SI.....	38
Figure 11 : Mécanisme de chargement existant pour Bridge .....	39
Figure 12 : Mécanisme de chargement existant pour DocEmedia .....	39
Figure 13 : Processus et scénarios identifiés .....	40

# AVANT-PROPOS

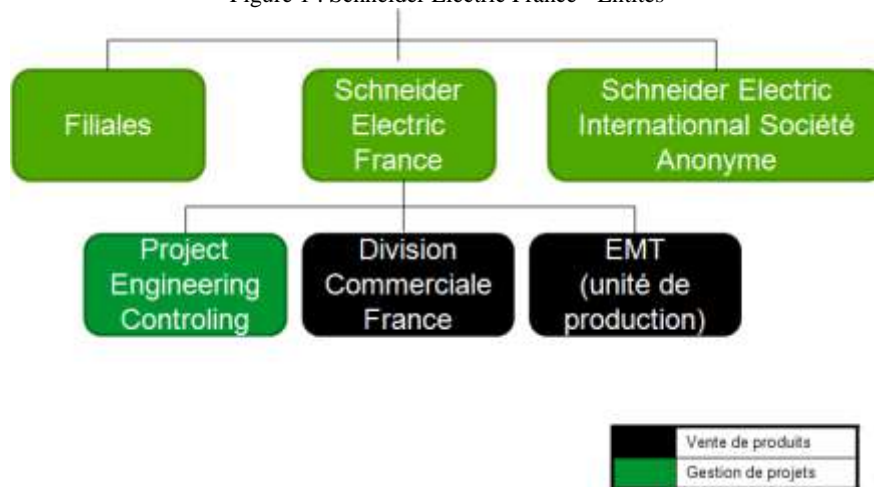
« L'organisation est une machine à maximiser les forces humaines. » P. Drucker

🌀 Présentation de l'entreprise et de l'équipe :

Le groupe Schneider Electric est implanté sur plus de 130 pays, il est spécialisé dans la gestion de l'énergie. Il existe pour chacun de ces pays une entité dédiée.

Sur le territoire français, plusieurs entités sont présentes: Schneider Electric France, des filiales du groupe et la société SEI (Schneider Electric International).

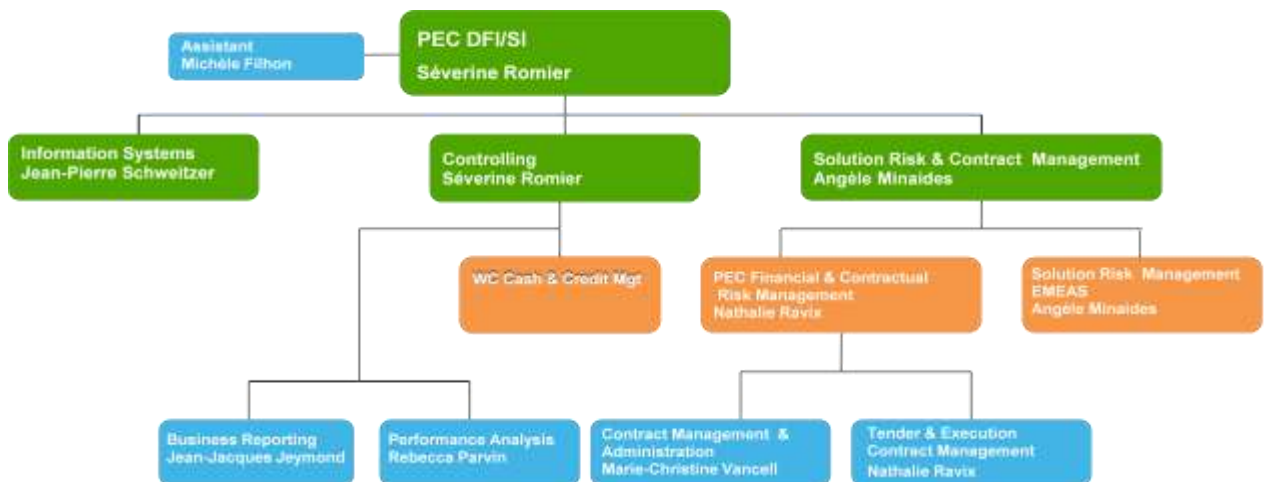
Figure 1 : Schneider Electric France - Entités



Au sein de Schneider Electric France, les entités DCF (Division Commerciale France) et EMT (Unité de production) vendent des produits via le catalogue de produit du groupe.

L'entité Project Engineering Center de Schneider Electric France (PEC) quant à elle, est en charge de la réalisation de projets et de solutions en France et à l'international.

Figure 2 : Schneider Electric France - PEC





C'est au sein du département Information Systems (IS), situé sur le site S2 de Grenoble que se déroule cette mission.

### Présentation de la mission de stage :

La mission de stage, d'une durée d'environ six mois, consiste à concevoir et mettre en place un système de génération de données en masse pour différents SI.

Afin de répondre à cet objectif, nous avons utilisé l'architecture des systèmes d'information existants. Il nous a fallu manipuler les données de départ et développer une interface partagée pour tous les SI concernés.

# INTRODUCTION

« Une question bien posée est une question à moitié résolue. » E. Michelin

- 🌀 Le but de ce stage est de concevoir et de mettre en place un système de génération des données de masse pour différents systèmes d'information de PEC. Pour cela, un besoin à été établi : il faudra s'appuyer sur les procédés de chargement existants pour chaque SI, afin de transformer les données de départ dans un format compatible avec ces procédés. Puis dans la mesure du possible, nous devons mettre en place une interface commune à tous les SI.
- 🌀 Ce besoin défini à l'origine du projet conduit à une démarche de développement informatique. Il amène à se questionner sur la solution informatique adéquate à mettre en place.
- 🌀 Ainsi, comme dans tout projet où un besoin informatique est identifié, une question essentielle se pose : faut-il opter pour un logiciel existant ou du développement spécifique ? Plus précisément, nous nous questionnons sur les critères sur lesquels baser notre choix ? C'est cette problématique des **critères de décision à prendre en compte dans le choix entre développement spécifique ou progiciel** que nous allons développer.
- 🌀 Afin de proposer des réponses à ce questionnement, nous allons dans un premier temps analyser l'existant sur ce sujet, les différentes théories et avis présents dans la littérature. Puis nous passerons à une analyse de terrain, avec un diagnostic et une analyse de l'expérience acquise lors de mon stage au sein de l'équipe IS à Schneider Electric. Enfin nous terminerons par une partie de confrontation entre théorie et pratique afin de dégager une proposition de solution.

# 1. Choix entre développement spécifique ou progiciel : la théorie

« *La théorie, c'est quand on sait tout et que rien ne fonctionne [...]* » A. Einstein

Dans le but d'identifier et de cerner correctement les thématiques existantes en terme de développement spécifique et progiciel, il convient tout d'abord de réaliser un état de l'art. Pour cela nous définirons les termes importants de notre sujet, puis grâce à l'analyse d'études de cas, nous identifierons les principales thématiques autour desquelles on analyse les avantages et contraintes des deux solutions.

## 1.1. Définitions

### 1.1.1. Notion de logiciel et développement spécifique

🌀 En informatique, nous pouvons définir un **logiciel spécifique** comme un *logiciel développé spécifiquement pour répondre à un besoin dans une organisation donnée.*<sup>1</sup>

Un logiciel spécifique ou sur mesure est donc un logiciel conçu par ou pour une entreprise et développé par cette entreprise, ou par un tiers pour cette entreprise.

Ainsi, le développement de logiciel sur mesure ou **développement spécifique** est l'action qui consiste à créer ce logiciel.

Le développement spécifique doit prendre en compte l'activité de l'entreprise, son contexte, son environnement et ses besoins spécifiques.

L'objectif final de ce type de développement est d'aider l'entreprise à mieux gérer certains aspects de son activité. Son principe est de permettre de créer un logiciel qui soit pleinement intégré à cette activité. De cette façon, il est possible de développer toute sorte de logiciel pour soutenir les grandes fonctions de l'entreprise (gestion, commerciale, SCM, gestion de stocks, CRM, workflow, gestion de projets, ressources humaines...).

Le logiciel ainsi créé dispose d'une **spécificité propre** à l'entreprise puisqu'il a été conçu exclusivement pour cette entreprise.<sup>2</sup>

---

<sup>1</sup> Source : Wikipédia – Logiciel spécifique (cf. BIBLIOGRAPHIE)

<sup>2</sup> Source : toutlocal.fr – Le développement de logiciel sur mesure (cf. BIBLIOGRAPHIE)

### 1.1.2. Notion de progiciel

🌀 Nous opposons au logiciel spécifique, développé pour une organisation, le terme de **progiciel**, contraction de **produit** et **logiciel** qui est *un logiciel développé pour répondre aux besoins d'un marché.*<sup>3</sup>

🌀 Le progiciel est un logiciel conçu par un éditeur pour répondre du mieux possible à des besoins partagés par le plus grand nombre. C'est un **logiciel « prêt à porter »** et **standardisé**.

Un progiciel est souvent un ensemble complet composé de plusieurs modules, couvrant chacun une fonction et communiquant entre eux.

🌀 C'est un produit conçu et développé par une entreprise, non pour elle-même, mais pour le vendre à d'autres entreprises qui l'utiliseront. Les progiciels applicatifs automatisent les fonctions de l'entreprise liées à son activité : comptabilité, gestion du personnel, achat, commercial, GPAO,...

Un progiciel ne peut être utilisé qu'avec les plateformes informatiques et les bases de données qu'il supporte. Un progiciel stratégique pour l'entreprise peut conduire à l'introduction ou au changement de plateforme.<sup>4</sup>

## 1.2. Etudes de cas, avantages et contraintes des deux solutions

🌀 Dans le but d'**identifier les principales thématiques** autour desquelles se construit le choix entre développement spécifique et progiciel, nous allons **analyser deux études des cas** :

- La mise en place d'une application métier chez trois PME : La Mutuelle des Motards, Rousselon et Playmobil France dans un article d'Aurélié Chandèze, journaliste (21 juin 2004)<sup>5</sup>
- La vision de Martin Fowler expliquée par Guillaume Ehret - Architecte, programmeur agile (4 Décembre 2012)<sup>6</sup>

<sup>3</sup> Source : Wikipédia – Logiciel spécifique (cf. BIBLIOGRAPHIE)

<sup>4</sup> Source : Cours évaluation des SI IAE-M2 O.Audaire (cf. BIBLIOGRAPHIE)

<sup>5</sup> Source : Developpez.com – Faut-il développer soi-même en PME ? (cf. BIBLIOGRAPHIE)

<sup>6</sup> Source : javamind-fr.blogspot.fr – NoSoftwarePackage (7/7) : développement spécifique ou progiciel (cf. BIBLIOGRAPHIE)

### 1.2.1. Etude de cas : « Faut-il développer soi-même en PME ? »

Le site Developpez.com a été contacté par la journaliste Aurélie Chandèze, dans le but de récupérer des témoignages autour du thème : « Faut-il développer soi-même en PME? ». Ce recueil a fait l'objet d'un article, qui recense les propos d'Aurélie Chandèze ainsi que les commentaires de membres de l'équipe Developpez.com.

Dans cette étude de cas, la journaliste nous présente la mise en place d'une application métier au sein de trois PME : la Mutuelle des Motards, Rousselon et Playmobil France. Plus précisément elle cherche à **analyser la démarche entreprise par les sociétés afin de choisir entre deux possibilités, l'achat d'un progiciel ou la réalisation d'une application sur mesure.**

La Mutuelle des Motards, PME de 350 salariés et spécialiste de l'assurance 2-roues et de la protection des conducteurs de moto et scooter souhaite moderniser son système d'information.

Le métier de cette entreprise diffère de celui des assureurs traditionnels, notamment à cause du risque élevé d'accidents corporels chez ses sociétaires.

Le **besoin métier** est donc **très spécifique**, et le responsable des systèmes d'information de l'entreprise, Thierry Caillard, déclare alors la nécessité de refondre l'outil existant : « le type de dommages était mal pris en compte dans les progiciels du marché [...] aucun progiciel pour l'assurance ne répond vraiment à ses besoins. »

La société a choisi d'opter pour une **solution mixte**<sup>7</sup>. Pour elle un progiciel, qui permet de gérer les principales règles métiers en offrant plusieurs modules du nouveau système d'information, tout en gardant une ouverture pour une personnalisation est la solution idéale. Thierry Caillard déclare ainsi que « l'autonomie qu'apportent ces développements sur mesure est un avantage, ce n'est plus l'organisation qui s'adapte à l'outil, mais l'outil qui est conçu pour répondre à nos besoins ».

---

<sup>7</sup> La Mutuelle des Motards a choisi l'offre de Prima Solutions pour les briques de leur nouveau système d'information. Elle réalisera les différents développements plus spécifiques avec l'aide de la société de services Open Wide.

🟢 La société Rousselon est une coutellerie de 50 personnes basée à Thiers, contrairement au spécialiste de l'assurance, cette entreprise a misé sur les progiciels. Le directeur général de la société Lionel Sol, déclare « Dans notre secteur, c'est une fadaise de croire qu'on ne peut trouver de progiciel bien adapté à notre entreprise ». Cette entreprise ne disposant pas des ressources nécessaires aux développements spécifiques, a donc choisi de gérer son activité grâce à un **progiciel de gestion**.<sup>8</sup>

Néanmoins quelques développements ont été réalisés afin d'adapter la solution aux besoins de l'entreprise. Aucun de ces développements n'a été fait en interne mais par une société externe. Toutefois la solution durable pour Lionel Sol reste la mise en place d'un progiciel qui ne nécessiterait aucun développement supplémentaire. En effet, en passant à une autre version du progiciel, la société à cause des développements spécifiques qu'elle a fait faire a rencontré plusieurs difficultés.

🟢 Pour Playmobil France, filiale du groupe allemand Geobra<sup>9</sup>, aucun progiciel présent sur le marché ne répondait à ses besoins très spécifiques.

En effet la maison mère fournit à ses filiales une gestion commerciale insuffisante selon Catherine Charlesson, directrice informatique de Playmobil France, étant donnée la diversité des besoins de chaque filiale.

C'est donc pour faire le **lien avec le logiciel** de gestion commerciale de la maison mère que le service informatique de la filiale Playmobil France a choisi de **développer une application en interne**.

Pour Catherine Charlesson un développement en interne au sein d'une PME comporte deux principaux risques : les équipes étant réduites<sup>10</sup>, le départ d'une personne peut avoir des conséquences fatales pour une application. L'autre risque est de ne pas pouvoir suivre les évolutions technologiques de l'application mise en place.

---

<sup>8</sup> La société Rousselon gère son activité avec un ancien progiciel de gestion d'Interlogiciel, G3. C'est la société DBI qui a réalisé les quelques développements spécifiques nécessaires.

<sup>9</sup> PLAYMOBIL® appartient au groupe Geobra Brandstätter GmbH & Co.KG, le premier fabricant de jouets en Allemagne. La société possède 12 filiales dont une en France : Playmobil France.

<sup>10</sup> Playmobil France est une entreprise d'une centaine de personnes.

### 1.2.2. Etude de cas : La vision de Martin Fowler

- 🕒 Guillaume Ehret, architecte et programmeur agile, nous expose dans un article publié sur le site Java Mind, l'idée du « Not Only Software Package » traduit « Pas Seulement du Progiciel » ou comment le développement spécifique peut devenir une *alternative plausible et parfois nécessaire*.
- 🕒 Selon Guillaume Ehret la mise en place d'un progiciel requiert une *intransigeance* sur les personnalisations du progiciel. D'après lui, il convient d'estimer les **coûts réel de la solution** et surtout *ne pas sous estimer les coûts d'un progiciel et sur estimer les coûts des développements internes*. Il met en garde également contre les différents coûts cachés existants, en effet le coût d'un progiciel n'est pas seulement le prix de sa licence. Le programmeur agile prône ici, une construction logiciel incrémentale, que la solution soit un progiciel ou un développement spécifique interne.
- 🕒 Dans son étude, Guillaume Ehret nous retransmet la **vision de Martin Fowler**<sup>11</sup> qu'il décrit comme un *pionnier et une référence au niveau développement [...]*.
- 🕒 L'idée de Martin Fowler est la suivante : **le cœur de métier d'une entreprise ne doit pas reposer sur des progiciels**, ses domaines stratégiques et concurrentiels doivent se baser sur du développement spécifique. Il explique que le développement spécifique est la stratégie qui répond aux besoins de réactivité et de sécurité pour la stratégie d'entreprise.
- 🕒 Martin Fowler illustre cette théorie avec le modèle : **UtilityVsStrategicDichotomy**. C'est une réflexion qui permet de classifier le besoin, de savoir quel est le facteur de distinction entre un projet « utilité » et un projet « stratégique » ? Ainsi lorsque l'on souhaite classifier un logiciel (ou une solution), il s'agit de savoir si la fonction métier derrière ce logiciel est un facteur de différenciation ou non. Autrement dit, si la façon dont l'entreprise réalise ce métier est une partie cruciale de ce qui la rend meilleure, plus performante que la concurrence. Si cette fonction est stratégique, la solution qui la prend en charge doit être *aussi bonne que possible*. Toutefois Martin

---

<sup>11</sup> Pour plus d'informations sur Martin Fowler : [martinfowler.com](http://martinfowler.com) (cf. BIBLIOGRAPHIE)

Fowler nous met en garde sur la distinction UtilityVsStrategic qui se fait bien sur la fonction, sur le métier de l'entreprise et non sur le logiciel en lui-même.

🕒 Dans son article, Guillaume Ehret prend l'exemple de la fonction suivi de la masse salariale. En suivant la pensée UtilityVsStrategicDichotomy, il classe le logiciel de gestion de la masse salariale dans la catégorie utilitaire. En effet le suivi de l'indicateur de la masse salariale est une fonction importante pour l'entreprise mais c'est une fonction commune à toutes les sociétés, et la mise en place d'un logiciel de suivi ne donnera pas un avantage concurrentiel à une entreprise.

De plus, toujours selon Guillaume Ehret, **l'effort dispensé** pour le déploiement d'un projet utilitaire ne doit pas être le même que pour la mise en place d'un logiciel dit stratégique.

🕒 Cependant la différenciation entre un projet utilitaire et un projet stratégique n'est pas si évidente, nous avons tendance à penser que tous les logiciels sont stratégiques pour le métier qui l'utilise, ainsi un contrôleur de gestion jugera son logiciel de suivi de la masse salariale comme stratégique. Mais une **solution stratégique pour un métier ne signifie pas toujours une solution stratégique pour l'entreprise**. C'est là, la distinction que fait le modèle UtilityVsStrategicDichotomy.

🕒 Après avoir classifié les projets, les solutions utilitaires ou stratégiques se développent différemment. Selon Guillaume Ehret il n'est ainsi pas *vital* pour l'entreprise d'allouer d'importantes ressources sur des projets de types utilitaires.

Pour ces projets, le progiciel est un *bon candidat*, il permet en effet de répondre au besoin de fonctionnement qu'exprime ce métier d'utilitaire.

A contrario, pour un projet stratégique le coût ne doit pas être compté car *le plus gros risque financier est de ne pas pouvoir agir à temps pour gagner un marché*. La solution stratégique développée doit ainsi permettre une intervention rapide, une **réactivité** de la part de l'entreprise. La solution doit absolument **éviter la rigidité des progiciels** qui risque de paralyser la capacité de l'entreprise à **se différencier**.



### 1.3. Identification des thématiques

🌀 Le raisonnement commun recensé lors de cet état de l'art est la connaissance du besoin de départ. Dans le choix entre progiciel et développement spécifique tous s'accordent à dire que **l'élément moteur est le besoin du demandeur.**

🌀 Dans le but d'identifier des thèmes sur lesquels baser notre réflexion, nous devons nous poser certaines questions.

- De quelles **ressources** disposons-nous ?

Afin de mener à bien un projet informatique, il convient de définir les ressources essentielles au projet, des ressources informatiques, humaines ou financières.

La mise en place d'un progiciel requière des ressources informatiques adéquates ainsi que les ressources humaines conséquentes pour la formation aux utilisateurs de l'outil installé.

Dans le cas d'un développement spécifique, le rôle des ressources métier est important. En effet si l'entreprise dispose d'un développeur ayant la connaissance du métier, la conception en interne peut se révéler réellement avantageuse. Néanmoins, certains problèmes apparaissent en phase de développement lorsqu'elle est réalisée en interne : en général le développeur n'a pas les pleines compétences métier pour développer un programme (programmes inachevés, problèmes de maintenance ou de fonctionnalités).<sup>12</sup>

Enfin un autre élément est à prendre en compte : le coût de développement. Suivant les projets les différences entre un progiciel et un développement spécifique peuvent être vraiment importantes, or le coût<sup>13</sup> est devenu un facteur décisif dans le choix d'un progiciel ou d'un développement spécifique.

---

<sup>12</sup> Source : Jean-Marc Rabilloud dans l'article : Developpez.com – Faut-il développer soi-même en PME ? (cf. BIBLIOGRAPHIE)

<sup>13</sup> Ici les coûts représentent les coûts de main d'œuvre, les coûts de licence, les coûts de matériels et les coûts de maintenance

- De quel niveau d'**adaptabilité** avons-nous besoin ?

Lors d'un développement spécifique, le projet développé contient exclusivement le besoin de l'entreprise, il s'adapte parfaitement à son organisation, développe les fonctionnalités attendues sans s'entourer d'apparat. Un progiciel sera quant à lui très adapté pour les besoins standards bien au-delà du besoin exprimé, mais il faudra s'assurer que les besoins exprimés sont pris en compte par le progiciel.

- De quel niveau de **réactivité** avons-nous besoin ?

Le développement spécifique permet une intervention dans le projet<sup>14</sup>, qu'elle que soit sa phase de développement, une modification peut être effectuée bien plus rapidement en interne que dans le cadre d'une relation client-fournisseur avec un éditeur de progiciel.

Les notions de dépendance, de support et de délai sont aussi des critères à intégrer dans la décision de développement (spécifique ou progiciel).

Au niveau de la réactivité la dimension stratégique ou non de la fonction supportée par la solution est également à considérer.

- De quel niveau de **portabilité** avons-nous besoin ?

Pour ce critère de portabilité, il nous faut prendre en compte l'infrastructure existante, mais également la technologie à développer et sa souplesse, notamment dans le but d'assurer la pérennité de la solution.

A présent que nous avons identifié et cerné les thématiques existantes en terme de développement spécifique et progiciel, que nous avons passé en revue certains des arguments présents autour de ce sujet, il devient intéressant d'analyser une situation de terrain.

---

<sup>14</sup> Une intervention dans le projet peut-être une modification que le client souhaite avoir dans la solution.

## 2. L'analyse de terrain IS, à Schneider Electric

*« Ne pas planifier, c'est programmer l'échec. » A. France*

Afin d'apporter une autre vision concernant le choix entre développement spécifique et progiciel, nous allons analyser la mise en place d'un système de génération de données en masse pour différents SI à Schneider Electric. Pour cela nous établirons le contexte de cette mission réalisée au sein de l'équipe IS, puis nous étudierons la démarche mise en place pour mener à bien ce développement. Enfin nous aborderons les différents critères de décision, les arguments, contraintes et avantages recensés lors de la mission.

### 2.1. Le contexte de la mission

- 🌀 Lors d'une mission où un besoin informatique est identifié, la question du choix de la solution à développer se pose. C'est la problématique que j'ai pu rencontrer lors de la mission effectuée au sein du service IS de l'entité Project Engineering Center (PEC) de Schneider Electric France.
- 🌀 La mission consistait à mettre en place un système de génération de données en masse pour différents systèmes d'information de PEC.
- 🌀 Un besoin informatique a tout de suite été constaté, le principe était de s'appuyer sur des mécanismes de chargement existants pour chaque SI, pour d'une part transformer les données de départ dans un format compatible avec ces mécanismes, et d'autre part pour proposer dans la mesure du possible une interface commune à tous les SI.<sup>15</sup>  
Ce besoin défini à l'origine du projet nous a conduit à une démarche de développement informatique. De ce fait, il nous amène au cœur de **notre questionnement : quelle est la solution informatique adéquate à mettre en place ?**
- 🌀 C'est donc dans le but de répondre à cette question que nous avons mis en place une **démarche** de projet.

---

<sup>15</sup> Besoin énoncé dans le contenu de l'offre 21884\_Schneider\_electric\_Stage\_MSI\_IAE\_vA\_1s

## 2.2. La démarche mise place

### 2.2.1. De l'analyse jusqu'au questionnement : développement spécifique ou progiciel ?

- 🌀 Afin de proposer une solution adaptée et fonctionnelle, il a fallu établir une démarche projet et un **planning** à suivre.<sup>16</sup>
  
- 🌀 Nous sommes tout d'abord passés par des **phases d'analyses**, nous permettant ainsi de cerner de façon complète le besoin identifié. Des analyses du contexte et de l'existant ont été effectuées, ce qui nous a permis une familiarisation avec les SI du périmètre étudié<sup>17</sup>. Nous avons également réalisé un **diagnostic du système à l'étude** consistant d'une part à l'identification et la compréhension des mécanismes de chargement concerné par l'étude (quels mécanismes de chargement ? quelles données engagées ? quelles procédures et quels cas d'utilisations ?)<sup>18</sup>, et d'autre part il a fallu identifier et analyser les « **scénarios** ». Les scénarios sont des processus<sup>19</sup> avec des données d'entrée et de sortie, des règles métier et de conversion. Ce sont ces scénarios que nous voulions industrialiser en mettant en place une solution informatique.
  
- 🌀 Ce n'est qu'à partir du recueil des besoins et des analyses accomplies, que nous avons pu commencer à imaginer vers quel type de solution nous souhaitions nous diriger. Puis dans le but de préciser et d'exprimer en détails ces règles définissant les scénarios, nous avons entamé un travail sur les **spécifications**.
  
- 🌀 A partir de là, plus le besoin se précisait, plus **le choix quant à la solution à développer se dessinait**. En effet, nous avons alors émis plusieurs idées concernant l'outil à mettre en place :
  - Un développement VBA Excel
  - Une solution ETL<sup>20</sup> (la solution Talend ou la suite Pentaho Data Integration)

---

<sup>16</sup> Planning prévisionnel (cf. ANNEXE A : Planning prévisionnel)

<sup>17</sup> Les SI concernés par l'étude (cf. ANNEXE B : Les Systèmes d'Information)

<sup>18</sup> Les mécanismes de chargements (cf. ANNEXE C : Les mécanismes de chargement)

<sup>19</sup> Les Workflows (cf. ANNEXE D : Les workflows)

<sup>20</sup> ETL pour d'Extract Transfert Load

- Afin de décider au mieux de l’outil à développer, nous avons entrepris la réalisation de maquettes. Trois maquettes qui permettraient d’évaluer la **faisabilité** de chaque solution envisagée, d’estimer son **adéquation aux besoins exprimés**, et d’apprécier les différents retours des principaux futurs **utilisateurs**.

### 2.2.2. Maquettage et démarche itérative

- Lors d’un projet informatique, la fabrication de la **maquette** d’un système va permettre de **tester la valeur de l’option envisagée**.

- C’est bien dans cet objectif de « tester la valeur » des différentes options, que nous avons réalisé nos maquettes.

Un outillage Excel macro VBA avait été développé dans le but de générer certaines données pour un des systèmes d’information de PEC. Nous sommes partis de cet outillage pour réaliser notre première maquette : application Excel macro VBA.

Après avoir effectué de nombreuses recherches sur les solutions ETL<sup>21</sup>, nous avons pu présélectionner deux solutions et ainsi construire deux maquettes : la maquette ETL 1 solution Talend, la maquette ETL 2 solution Pentaho Data Integration<sup>22</sup>.

- Ces maquettes ont tout d’abord été réalisées en local sur une seule machine<sup>23</sup>. Ce développement a permis d’avoir un premier aperçu en termes de **besoins exprimés**.

Puis nous avons choisi d’aller un peu plus loin avec les maquettes pour les solutions ETL. Nous avons cherché à installer les applications sur un serveur de pré-production<sup>24</sup> afin de **vérifier la faisabilité des solutions**. De cette façon, nous avons éliminé la solution Talend qui ne répondait pas à certains critères de faisabilité énoncés.

- Afin de répondre le mieux possible au besoin des utilisateurs<sup>25</sup>, nous avons présenté **l’avancement du développement** (les maquettes) lors de certaines réunions d’équipe hebdomadaires. Une présentation ou parfois un simple compte-rendu permettait un suivi

---

<sup>21</sup> ETL : outil d’extraction, de transformation et de chargement – solutions qui, a priori, répondaient à nos besoins. Recherche pour une solution libre open source.

<sup>22</sup> Pentaho Data Integration nommé aussi Kettle

<sup>23</sup> Sur mon ordinateur personnel

<sup>24</sup> Serveur pré-production Schneider-Electric

<sup>25</sup> Utilisateur=membres de l’équipe IS

régulier et nous donner ainsi la possibilité d'avoir un **retour des utilisateurs** et **d'ajuster notre développement** en fonction de ces remarques. De cette manière, en suivant toujours le besoin utilisateur, nous pouvions effectuer correctement notre choix entre une solution progiciel (ETL) ou un développement spécifique (application Excel macro VBA).

### 2.2.3. Démarche en terme de management

🕒 Dans une démarche projet, la **dimension de planning** et de respect du délai est un principe important. Dans notre analyse des critères de décision à prendre en compte dans le choix entre développement spécifique et progiciel, le délai fait parti d'un de ces critères à considérer.

🕒 Au cours de la mission, nous avons dû repousser la **phase de développement de l'outil car le choix de la solution n'était pas encore arrêté** (choix entre la solution ETL Pentaho ou développement Excel VBA). En effet, des erreurs apparues en pré-production sur la maquette ETL remettaient gravement en cause la faisabilité même de l'application souhaitée, ainsi que des incertitudes concernant des fonctionnalités indispensables dans l'environnement « cible ».

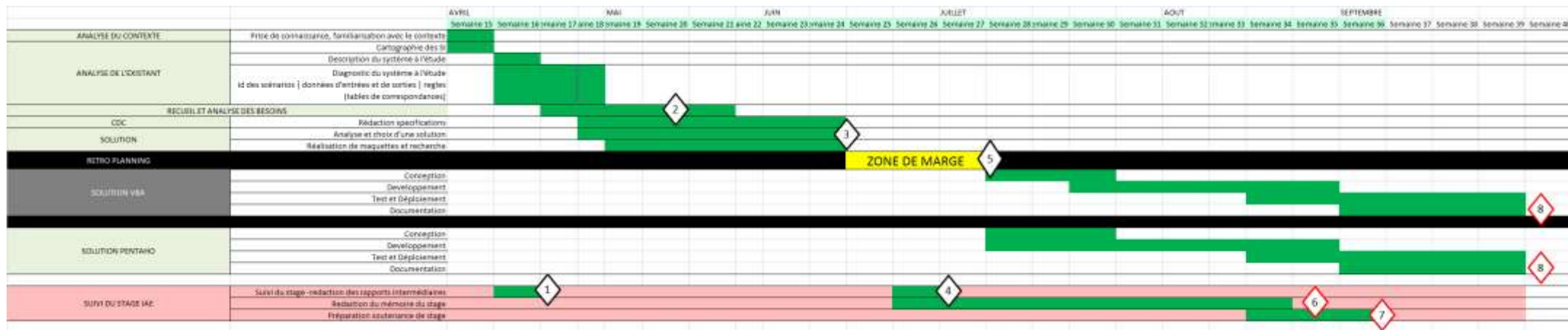
Suite à ce genre d'événement, une bonne **réactivité** est primordiale, ainsi une réunion a été rapidement organisée, et nous avons décidé de mettre en place un **retro-planning**. Un retro-planning est *une planification inversée, conçu en partant de la date de fin du projet puis en remontant dans le temps afin de positionner les jalons. Il est quelquefois plus facile d'utiliser cette méthode pour réaliser une planification, lorsque la date de fin de projet est fixée et inébranlable.*<sup>26</sup> Nous avons défini un jalon de début de développement, soit la date au plus tard à partir de laquelle une solution pouvait être développée et permettait de produire une application qui soit fonctionnelle à la fin de la mission. De ce fait, nous avons pu identifier une **date de décision** (un jalon) à partir de laquelle le choix de la solution à mettre en œuvre devait être fait.

La **marge de manœuvre** ainsi disponible nous a permis de continuer notre phase de maquettage, afin de prendre la meilleure décision possible quant à la solution à mettre en œuvre, tout en garantissant un livrable à la fin du projet.

---

<sup>26</sup> Source : Wikipédia – Retro-planning (cf. BIBLIOGRAPHIE)

Figure 3 : Retro-planning



## Légende :

- 1- Livrable n°1 suivi-IAE : Fiche de stage
- 2- Présentation à l'équipe IS et discussion du besoin
- 3- Prise de décision d'élaboration d'un retro planning, quelle solution à mettre en œuvre ? possibilité de continuer la "recherche" ETL : oui /non ?

La zone de marge (en jaune dans le graphique) déterminé grâce au retro-planning, représente la marge de manœuvre disponible.

- 4- Livrable n°2 suivi-IAE : Avancement du stage
- 5- Choix définitif de la solution à mettre en œuvre, date au plus tard pour le démarrage du développement
- 6- Mémoire de stage
- 7- Soutenance de stage
- 8- Livrable Schneider Electric

## 2.2.4. Résultats

En tenant compte de la contrainte de la marge de manœuvre disponible nous avons effectué des recherches, réalisé des modifications et de nombreux tests sur la maquette ETL Pentaho (serveur pré-production), afin de chercher à résoudre ces problèmes qui entamaient la faisabilité de la solution.

La réalisation de ces fonctionnalités requises sur la maquette ETL Pentaho, a permis de clore la phase de maquettage, et nous a permis d'effectuer notre **choix entre la solution progiciel (ETL) ou le développement spécifique** (application Excel macro VBA).

Après avoir étudié les maquettes sous différents critères et analysé tout les arguments (cf. partie suivante : 2.3) nous avons choisi de développer la **solution progiciel ETL Pentaho**.

Cette solution, ETL Pentaho, est un **module appartenant à la suite Pentaho**, une suite logicielle décisionnelle complète ayant pour principal objectif d'effectuer la distribution de diverses fonctionnalités à plusieurs utilisateurs par l'intermédiaire d'une interface web. Pentaho est un progiciel de gestion intégré qui s'inscrit dans la famille des logiciels **open source**.<sup>27</sup>

Le progiciel Pentaho se compose de plusieurs modules, la **présence d'un de ces modules dans le service IS**, a favorisé le choix quant à intégrer un second module de la même suite logicielle. En effet le portail Pentaho nommé Business Intelligence Server a été mis en place en 2013 dans le cadre d'une mission de stage. Ce portail, utilisé par l'équipe IS dans le but de réaliser des analyses de données servira également d'**interface web pour l'outil ETL** que nous souhaitons à mettre en place.

En définitive, le développement de la solution se décomposera en plusieurs tâches : Il faudra **installer et paramétrer le module ETL**, nommé Pentaho Data Integration, un outil d'alimentation et de transformation ETL. Puis nous devons **paramétrer le module/portail Pentaho** déjà en place afin de pouvoir utiliser l'interface web.

---

<sup>27</sup> Pour plus d'information sur la suite Pentaho : <http://community.pentaho.com/>



### 2.3. Les critères de décision et arguments recensés

- ④ Afin d'effectuer notre **choix** quant à développer une solution progiciel (ETL) ou un développement spécifique (macro Excel), nous nous sommes basés sur différents **critères**. Nous avons également listé plusieurs **arguments** répertoriant les contraintes et avantages des deux solutions envisagées.
- ④ Tout d'abord, l'un des principaux critères sur lequel nous avons fondé notre réflexion est **le besoin utilisateur**. En effet tout au long de la démarche, nous nous sommes référés à l'utilisateur, à son besoin, ses différents retours et remarques vis-à-vis de la solution à développer.
- ④ La mise en place d'une **phase de maquettage** est également primordiale. Cette étape permet en effet d'étudier la **faisabilité de l'outil**, les principales **fonctionnalités** qui lui sont **indispensables**, et la maquette permet aussi d'apprécier une première vue de **l'interface utilisateur**.
- ④ Un critère essentiel à prendre en compte lors de la conception de la solution est la problématique de **maintenabilité de l'outil**. En effet l'application devra pouvoir être maintenue par les utilisateurs<sup>28</sup>, que ce soit au niveau de la **pérennité** de la technologie utilisée ou en termes de **facilité** d'administration (mise à jour ou modification de l'outil).
- ④ Concernant les différents avantages et contraintes des **deux solutions**, nous allons lister les **principaux arguments recensés**.  
Nous choisissons ici de ne pas détailler les différents critères et arguments. Il s'agit, dans cette partie du mémoire, plus d'un recensement que d'une complète étude des critères de décision. Cette analyse des critères s'effectuant dans la troisième partie du mémoire : La confrontation de la théorie et de la **pratique**.
- ④ Nous commençons par répertorier les arguments de la solution progiciel - ETL Pentaho Community, module Pentaho Data Integration.

---

<sup>28</sup> Rappelons que dans notre cas utilisateurs=administrateurs=membres de l'équipe IS.

Le **paramétrage** nécessaire afin d'adapter le module Pentaho Data Integration à la structure souhaitée est conséquent.


En effet, la **spécificité open source** de la solution nécessitera d'effectuer de nombreuses **recherches** pour développer l'application. Cependant son **aspect communautaire** est un véritable avantage pour la pérennité de l'outil et pour son développement. Cette partie « recherche et paramétrage » plutôt chronophage devra être correctement évaluée et pris en compte, notamment lors de l'évaluation des tâches.

La **richesse fonctionnelle de la solution ETL Pentaho Community** va bien au delà de notre besoin, et se traduit donc par la présence de nombreuses fonctionnalités inutiles pour ce projet.

La solution Pentaho offre une **interface très ergonomique** pour l'utilisateur (le portail web).

La présence d'un module Pentaho déjà en activité dans le service favorise le choix d'intégrer un autre module de la même **suite : Pentaho**.

La **maintenabilité** de l'outil Pentaho ETL sera beaucoup plus accessible pour un utilisateur « non développeur ».

 Pour la solution de développement spécifique – Macro VBA Excel, voici la liste des arguments répertoriés :

La solution Excel propose une **interface** sommaire et fonctionnelle, mais beaucoup **moins ergonomique** pour l'utilisateur.

Concernant les compétences techniques, le **développeur** possède des **compétences** plus performantes pour ce développement. Cependant, la **spécificité « stage »** de la mission implique un aspect pédagogique, et le développement de la solution ETL apporterait plus en termes de **compétences** et **d'apprentissage** que le développement Macro VBA Excel.

Le développement Excel est un développement « sur mesure », toutes les fonctionnalités exprimées peuvent être développées, il n'y a donc **pas d'incertitude fonctionnelle**.

La **maintenabilité** de l'outil Excel sera plus accessible pour un utilisateur « développeur VBA ».

🔄 Le tableau ci-après nous résume les avantages et contraintes pour chaque solution.

Figure 4 : Tableau récapitulatif des avantages et contraintes des deux solutions<sup>29</sup>

Solution	Progiciel (ETL Pentaho)	Développement spécifique (Macro Excel)
Critères		
Besoin Faisabilité et fonctionnalités de l'outil	<p>Paramétrage intégré</p> <p>Fonctionnalités inutiles</p> <p>Fonctionnement Client/serveur</p>	<p>Pas d'incertitude fonctionnelle</p> <p>Mono Poste</p>
Interface de l'outil	Interface plus ergonomique	Interface moins ergonomique
Technicité et maintenabilité de l'outil	<p>Maintenabilité (graphique)</p> <p>Open source : recherches</p> <p>Open source : communauté</p> <p>Module Pentaho déjà présent</p>	<p>Maintenabilité (code)</p> <p>Capacité du développeur plus performant sur ce développement</p>
Aspect enseignement (stage)	Apport en termes de compétences et d'apprentissage	

Après cette analyse de terrain, où nous avons décrit la démarche mise en place lors de la mission, nous avons recensé les différents résultats, critères de décisions et arguments applicables à cette expérience, nous allons à présent analyser ces arguments théoriques et pratiques.

<sup>29</sup> Dans le tableau, les contraintes des solutions sont notifiées de couleur rouge et les avantages de couleur verte, le code couleur orange représente un argument ambigu qui peut être considéré comme avantage ou inconvénient selon l'utilisateur.

## 3. La confrontation de la théorie et de la pratique

« *In theory, there is no difference between theory and practice. In practice, there is.* » Y.Berra

Dans l'objectif de proposer une démarche à suivre et une réponse quant aux critères de décision à prendre en compte dans le choix entre développement spécifique et progiciel, nous allons maintenant confronter les apports des arguments mis en lumière grâce à l'analyse théorique et ceux recensés grâce à l'expérience de terrain.

### 3.1. Analyses et comparaison des arguments théoriques et pratiques

#### 3.1.1. Les thématiques et arguments

🌀 **Avant de choisir, il faut connaître son besoin**, c'est là le principe fondamental à toute démarche de développement. Que le projet s'oriente vers une solution progicelle ou vers un développement spécifique, le premier critère à prendre en compte avant de prendre une décision, est le besoin du client.

🌀 Un argument également très présent dans la littérature concerne **les fonctionnalités de l'outil** à développer. En effet le niveau d'adaptabilité de l'outil au besoin, est primordial à définir. Il est important de connaître le degré d'adéquation nécessaire entre les besoins et attentes du client et les fonctionnalités qu'il est possible de mettre en place. Quelles sont les principales fonctionnalités indispensables au futur outil ? en matière de faisabilité ? Quelle interface d'utilisation souhaitons-nous avoir ? en termes d'ergonomie mais aussi pour l'interface client-outil ? Autant de questionnements qui nous ont été décisifs dans notre choix de développement lors de la mission au sein de Schneider.

Il faut également identifier s'il est essentiel ou non pour le projet que l'outil contienne exclusivement le besoin de l'entreprise. Ainsi, dans certaines entreprises qui ont opté pour un progiciel, l'utilisation réelle de l'outil concernait seulement 5 à 20 % de ses fonctionnalités. Un développement spécifique aurait pu répondre aussi rapidement aux attentes sans ajouter de fonctions inutiles, mais l'entreprise a choisi un progiciel car ce type de développement correspondait à ses besoins et sa stratégie.<sup>30</sup>

---

<sup>30</sup> Source Gaël Donat dans Developpez.com – Faut-il développer soi-même en PME ? (cf. BIBLIOGRAPHIE)

🌀 La **problématique des ressources disponibles** est un critère très important à prendre en compte. Il est primordial d'allouer les ressources adéquates au projet, que se soit en termes de ressources informatiques, financières ou de ressources métier, il faut les identifier.

Ensuite en fonction de ces ressources mises à disposition, nous allons continuer notre démarche pour réaliser notre choix entre un développement spécifique ou progiciel. Parfois ce critère est un élément déterminant dans la décision de développement. C'est le cas de plusieurs PME qui choisissent des progiciels au détriment d'une application montée en interne par défaut de ressources en interne pour faire le développement.<sup>31</sup>

Lors de la mise en place d'un système de génération de données en masse à Schneider-Electric, la compétence du développeur (et des membres de l'équipe IS en matière de développement) était un des facteurs clé dans le processus de décision. Cependant la particularité de la mission (un stage) nous a menés vers un autre aspect que le simple savoir-faire de la ressource. En effet lors d'un stage, la **notion d'apprentissage** est à intégrer, et gagner en compétences sur une nouvelle démarche, sur une technique ou solution représente un grand intérêt.

🌀 Les concepts de **pérennité**, **maintenabilité** de l'outil et de **réactivité** sont essentiels à considérer. Le niveau de maintenabilité est un élément qui a été étudié lors de la mission. La solution à mettre en place devait être supportée par une technologie stable, et elle devait pouvoir être maintenue par l'utilisateur (équipe IS). Ainsi de nombreuses contraintes ont été soumises quant à la maintenabilité de la future solution. Nous devons notamment pouvoir intégrer des évolutions de structure aisément (structure des fichiers input et output), sans avoir à aller « se dépatouiller dans le code ».

Ces critères de pérennité et de réactivité sont d'autant plus forts que la dépendance au fournisseur est importante. En effet, que l'on choisisse un développement spécifique en interne, les services d'une société externe ou un progiciel d'un éditeur, la dépendance de la relation client-fournisseur est primordiale à prendre en compte. La question du support survient alors, de quel support pouvons-nous disposer ? Là encore c'est selon le besoin de support, qu'il faudra choisir la solution à développer.

Pour le développement de l'outil de génération de données, le choix d'un progiciel ETL **open source** apportait une dimension singulière au support : **l'aspect communautaire**.

---

<sup>31</sup> Source Gaël Donat dans Developpez.com – Faut-il développer soi-même en PME ? (cf. BIBLIOGRAPHIE)

La communauté open source apporte une certaine garantie dans la pérennité de l'outil. En effet les développeurs assurent un support en termes de qualité et de quantité de fonctionnalités. La présence de nombreux forums aide à ce support. Le développement communautaire favorise également la réactivité lorsqu'il s'agit par exemple de corriger un bug.

- 🌀 Puis il y a le critère de la **cohérence de la solution**. Il faudra ainsi examiner l'infrastructure existante et faire son choix en fonction de la souplesse nécessaire pour porter le futur outil dans la structure de l'entreprise.

Dans notre cas terrain à Schneider-Electric, la présence d'une technologie compatible et issue de la même suite logicielle a orienté notre choix quant au développement progiciel ETL ou non. C'est donc un critère capital à considérer, savoir comment la solution va pouvoir s'assembler avec les technologies, l'architecture et l'organisation déjà en place.

- 🌀 Enfin, nous avons vu qu'en fonction de la **dimension utilitaire** ou stratégique du projet, la solution à mettre en place s'orientait d'avantage vers du spécifique dans le cas d'un projet plus stratégique, ou vers une solution progiciel lorsque que le projet est purement utilitaire. Lors du développement de l'outil au sein de l'équipe IS à Schneider-Electric, le projet a été clairement identifié comme une mission utilitaire, en effet l'outil a été conçu comme un support de travail pour certains membres de l'équipe.

Ce critère de dimension utilitaire vs stratégique nous permet ainsi de soutenir une certaine cohérence dans notre raisonnement concernant le choix progiciel ETL.

### 3.1.2. Les processus décisionnels présentés

- 🌀 Après avoir analysé les différents aspects et contraintes à prendre en compte dans la décision entre développement spécifique et progiciel, nous allons pouvoir analyser les **processus décisionnels proposés**.

- 🌀 Le site developpez.com, dans son étude de cas : Faut-il développer soi-même en PME<sup>32</sup>, nous présente le processus décisionnel qu'il a identifié.

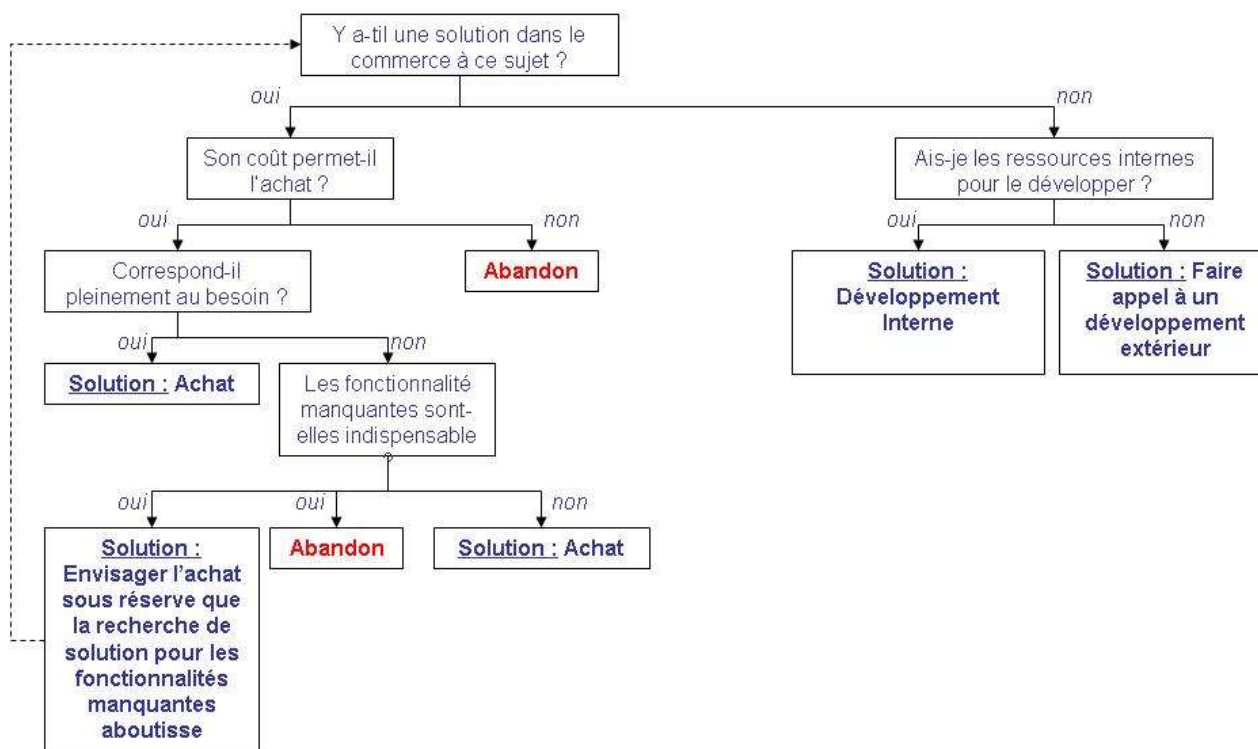
---

<sup>32</sup> Source : Developpez.com – Faut-il développer soi-même en PME ? (cf. BIBLIOGRAPHIE)

Selon Maxence Hubiche (auteur du commentaire sur developpez.com), pour chaque étape il convient de passer par une réflexion approfondie, tout en restant *en phase* avec les critères initialement retenus (l'étude de cas avait identifié des critères en termes de coûts, de dépendance, de support, de technologie, de délai et de souplesse).

Dans la figure ci-dessous, nous pouvons voir que le **processus est très centré sur les ressources**, notamment en termes de coûts. En effet, le workflow conduit à l'abandon lorsque le coût de la solution progiciel présente sur le marché ne permet pas son achat par l'entreprise. De plus l'entrée du processus ne semble pas être le besoin mais plutôt la présence ou non d'une solution clé en main dans le commerce : la première étape du processus est la suivante « Y a-t-il une solution dans le commerce à ce sujet ? ». La phase de correspondance avec les besoins n'arrive qu'à l'étape trois « correspond-t il pleinement au besoin ? ».

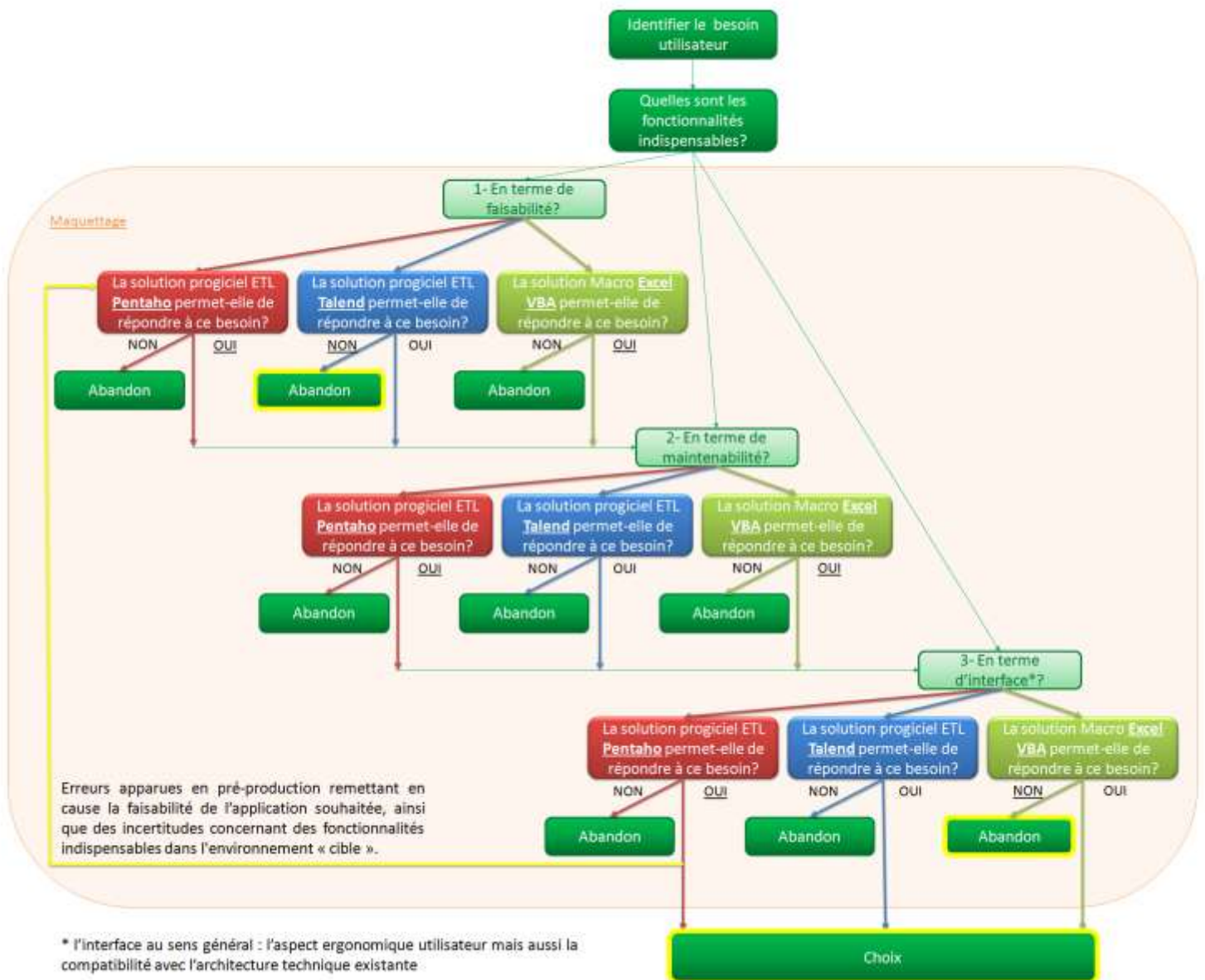
Figure 5 : Processus décisionnel identifié par developpez.com<sup>33</sup>



<sup>33</sup> Source : Developpez.com – Faut-il développer soi-même en PME ? (cf. BIBLIOGRAPHIE). Le processus décrit représente bien une conclusion de l'étude de cas, il résume les pratiques recueillies et recensées dans l'article.

- 🌀 Dans le but d'analyser la décision prise lors de la mission au sein de Schneider-Electric quant au choix entre progiciel ou développement spécifique, nous allons décrire le **processus décisionnel établi pendant cette expérience de terrain.**

Figure 6 : Le processus décisionnel de l'expérience terrain à Schneider-Electric



- 🌀 Nous sommes tout d'abord partis du **besoin utilisateur** afin des recenser les **fonctionnalités indispensables** dans l'environnement de la futur solution. Contrairement au processus précédent (Figure 5 : Processus décisionnel identifié par developpez.com), qui se concentrait sur les coûts, nous mettons ici l'accent sur les fonctionnalités de l'outil.



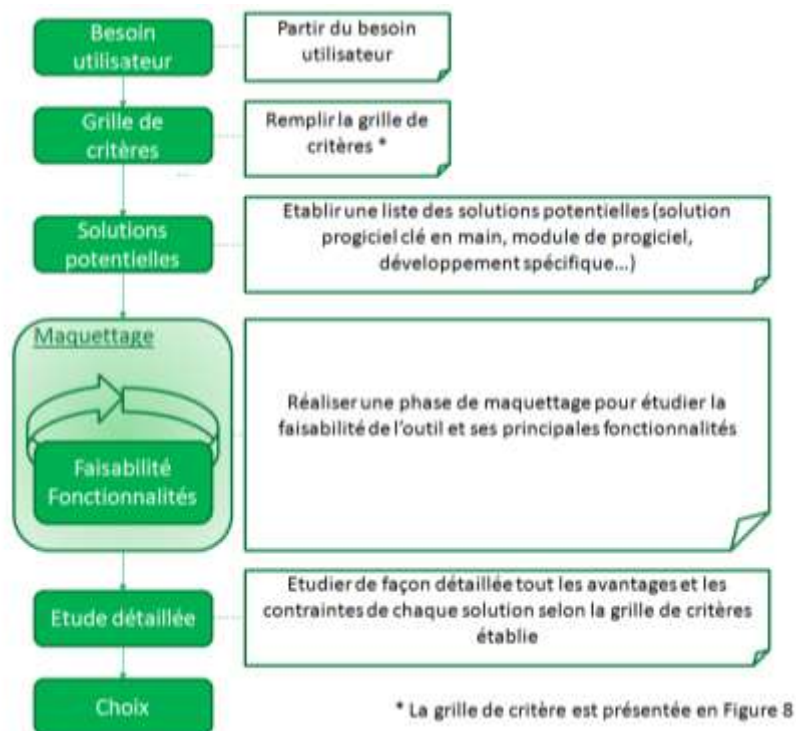
### 3.2. Proposition d'une démarche à suivre

🌀 Dans le but d'apporter une réponse à notre problématique de départ, à savoir « Quels sont les critères de décision à prendre en compte dans le choix entre développement spécifique et progiciel ? », nous souhaitons proposer une **démarche à suivre quant la décision entre développement spécifique et progiciel**.

Cette proposition est le fruit de l'analyse théorique et de l'analyse de terrain réalisé dans ce mémoire. Nous présentons ici, une des démarches possibles pour cette problématique.

🌀 Cette procédure se prête parfaitement à la mission réalisée au sein du service IS à Schneider-Electric, elle peut également être adaptée à d'autres projets. Toutefois cette démarche n'est pas générique, elle pourra ne pas convenir selon les projets, c'est pourquoi il est primordial d'évaluer le besoin du projet en premier lieu, avant de partir dans une démarche décisionnelle.

Figure 7 : Une proposition de démarche à suivre



🌀 La démarche que nous préconisons est de partir de « l'élément moteur du projet » : le besoin du client. Puis d'établir une **grille de critères** : cette grille va permettre d'identifier les points importants à prendre en compte dans la décision finale pour le choix de la solution.

Figure 8 : Grille de critères

Critères	Besoin	Ressources				Adaptabilité			Portabilité		
		Informatiques	Humaines	Financières		Fonctionnalités indispensables	Réactivité support et délai	Dimension stratégique de la fonction supportée par la solution	Infrastructure existante et technologie		
				coûts de main d'œuvre	coûts de licence					coûts de matériels	coûts de maintena nce
Solution 1											
Solution 2											

🔄 Dans un premier temps, il faudra **remplir les critères** que l'on souhaite voir apparaître dans la future solution : quel est le besoin ? de quelles ressources mon projet dispose-t-il ? quelles fonctionnalités sont indispensables dans ma future solution ? de quel niveau de réactivité et de support ai-je besoin ? quelle est l'architecture existante ? autant de questions auquel il va falloir répondre avant d'envisager de mettre en place une quelconque solution<sup>34</sup>.

🔄 Ensuite il faudra **lister plusieurs solutions potentielles**. Vous devrez alors réaliser une maquette pour chacune des solutions exprimées, des maquettes qui vous permettront d'étudier ces outils, les fonctionnalités et interfaces possibles ou encore la faisabilité même de la solution.

🔄 Enfin grâce à la grille de critères vous pourrez réaliser une **étude détaillée** de chaque solution. Vous devrez pour cela, compléter **pour chaque critère les avantages ou contraintes** observés pour la solution.<sup>35</sup>

Enfin vous pourrez faire votre **choix** entre les solutions. Le choix devra reposer sur l'analyse effectuée, tout en se basant sur le besoin exprimé<sup>36</sup>.

<sup>34</sup> Les réponses à ces questions sont à remplir dans la grille des critères, dans la **ligne en rouge** (Figure 8 : Grille de critères)

<sup>35</sup> L'étude est réalisée grâce aux dernières lignes du tableau (Figure 8 : Grille de critères)

<sup>36</sup> Il ne s'agit pas ici de prendre la solution avec le plus d'avantages et le moins d'inconvénients, mais de choisir celle le mieux adaptée à vos besoins.

# CONCLUSION

« Conduire un projet, c'est passer d'un début où l'on peut tout faire, mais où on ne sait rien, à une fin où on n'a plus aucun degré de liberté pour exploiter toute la connaissance qu'on a acquise en cours de projet. » C. Midler

🕒 L'objectif de ce mémoire est de chercher à savoir quels sont les **critères de décision à prendre en compte dans le choix entre développement spécifique et progiciel ?**

🕒 Au travers de la réalisation d'un état de l'art et d'une analyse de terrain nous avons tenté d'apporter quelques éléments de réponses en proposant une **démarche à suivre** :

🕒 Nous avons vu qu'entre un progiciel non paramétrable et un développement 100% sur mesure se trouvait un vaste territoire de solutions.

Bien qu'il existe, plusieurs **critères de décision** face au choix entre une solution progiciel ou une solution spécifique, le critère indispensable à tout projet est le **besoin du client**. C'est à partir de ce critère que nous devons initier la démarche pour la conception et le développement d'une solution informatique adéquate.

🕒 La **démarche préconisée** dans le mémoire, n'est qu'une des différentes façons de procéder dans le but d'effectuer le choix entre développement spécifique et progiciel. Elle se base sur des critères recensés dans la littérature, des études de cas et s'appuie également sur mon expérience au sein de Schneider-Electric.

🕒 Au cours de cette mission nous avons pu aborder la **dimension « open source »** d'une solution informatique. Cette problématique s'ajoute au dilemme qui existe entre développement spécifique et progiciel, elle se doit d'être pris en compte dans le développement d'un outil, qu'il soit spécifique ou logiciel.

Il nous paraît ainsi cohérent, **en complément de notre démarche décisionnelle**, d'établir des règles, des critères de décision afin de nous guider dans ce choix entre développement spécifique et progiciel tout en intégrant la dimension des logiciels open source ou propriétaire.

# BIBLIOGRAPHIE

Wikipédia. 2014. Wikipédia – Logiciel spécifique [En ligne] Article de Wikipedia.org. Page consulté le 12/08/2014. Disponible sur :

[http://fr.wikipedia.org/wiki/Logiciel\\_sp%C3%A9cifique](http://fr.wikipedia.org/wiki/Logiciel_sp%C3%A9cifique)

Toutlocal.fr. 2014. Le développement de logiciel sur mesure [En ligne] Article de Toutlocal.fr. Page consulté le 12/08/2014. Disponible sur :

[http://www.toutlocal.fr/Le\\_developpement\\_de\\_logiciel\\_sur\\_mesure-a954121.html#8215147](http://www.toutlocal.fr/Le_developpement_de_logiciel_sur_mesure-a954121.html#8215147)

Olivier AUDAIRE. 2014. Séminaire Grenoble IAE 2014 SI et contrôle de gestion – Evaluation du SI. Cours d'évaluation des SI, plus particulièrement les diapos 38 à 48.

Document consulté le 10/07/2014. Disponible sur l'intranet de l'IAE :

33339\_1\_2\_Evaluation\_SI\_-\_Cours\_1er\_Journee.

Maxence HUBICHE. 2004. Faut-il développer soi-même en PME? [En ligne] Article de developpez.com. Page consulté le 12/08/2014. Disponible sur :

<http://mhubiche.developpez.com/journaux/mondeinformatique-mai2004/#LI>

Guillaume EHRET. 2012. NoSoftwarePackage (7/7) : développement spécifique ou progiciel ? [En ligne] Article de Java mind. Page consulté le 12/08/2014. Disponible sur :

<http://javamind-fr.blogspot.fr/2012/12/nosoftwarepackage-77-quelles-solutions.html>

Martin FOWLER. 2010. UtilityVsStrategicDichotomy [En ligne] Article de martinowler.com. Page consulté le 12/08/2014. Disponible sur :

<http://martinowler.com/bliki/UtilityVsStrategicDichotomy.html>

Wikipédia. 2014. Wikipédia – Retro-planification [En ligne] Article de Wikipedia.org. Page consulté le 12/08/2014. Disponible sur :

<http://fr.wikipedia.org/wiki/R%C3%A9troplanification>

# ANNEXES

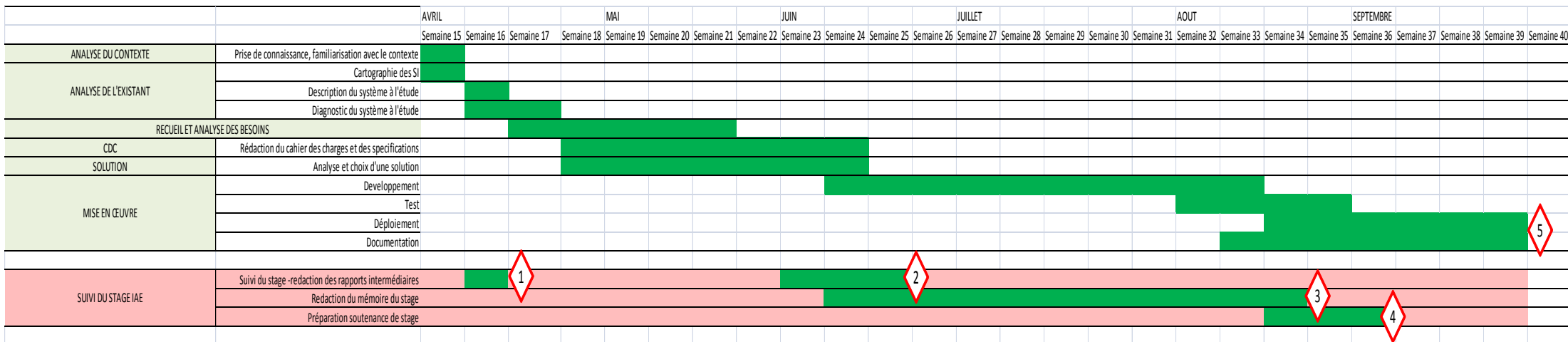
ANNEXE A : Planning prévisionnel .....	37
ANNEXE B : Les Systèmes d'Information .....	38
ANNEXE C : Les mécanismes de chargement .....	39
ANNEXE D : Les workflows.....	40

# ANNEXE A : Planning prévisionnel

## Planning prévisionnel avril 2014 :

Le planning intègre les **tâches** liées à la **mission au sein de Schneider-Electric** et également les **tâches** liées aux **livrables IAE** (rapports intermédiaires, mémoire et préparation de la soutenance).

Figure 9 : Planning prévisionnel



Légende :

- 1- Livrable n°1 suivi-IAE : Fiche de stage
- 2- Livrable n°2 suivi-IAE : Avancement du stage
- 3- Mémoire de stage
- 4- Soutenance de stage
- 5- Livrable Schneider Electric

## ANNEXE B : Les Systèmes d'Information

Présentation des SI du périmètre étudié :

Figure 10 : Tableau des SI

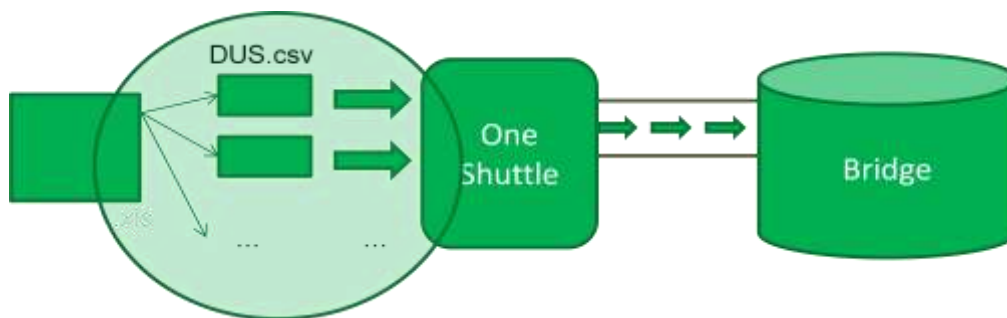
SI	Utilisation	Utilisateurs	Type de flux	Type d'information
<b>Bridge</b>	ERP : commande clients, gestion approvisionnement, SC, facturation, structuration projet	PM et BM, techniciens, logisticiens, IGFC, qualité, gestion, comptabilité, achats	Interne, Sortant, Entrant	Fonctionnement
<b>DocEmedia</b>	GED : Partage de documents, référentiel unique pour la gestion des docs, versioning et workflow	Tous les métiers PEC	Sortant, Entrant	Fonctionnement

## ANNEXE C : Les mécanismes de chargement

Afin de réaliser un diagnostic du système à l'étude, nous avons étudié les deux systèmes d'information au cœur de la solution que nous souhaitons développer.

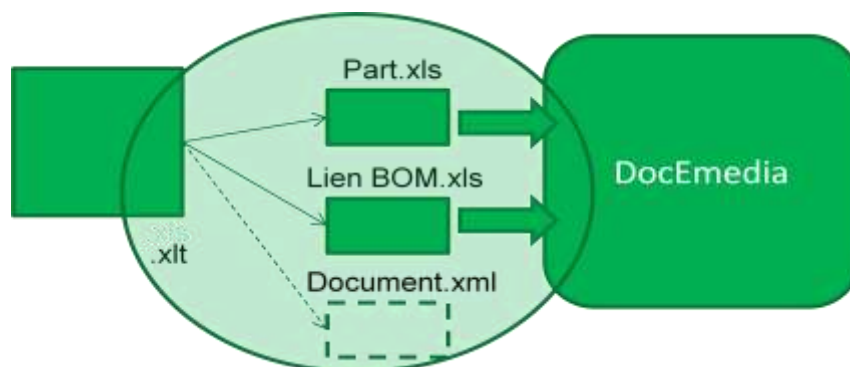
Le mécanisme de chargement existant pour Bridge est un système « corporate » nommé OneShuttle. Il est basé sur des Data Upload Sheet (DUS) des fichiers au format .csv (ou .xls) qui permettent, grâce à une requête d'import (DUS Upload Request) de charger des données sur Bridge.

Figure 11 : Mécanisme de chargement existant pour Bridge



Le mécanisme de chargement existant pour DocEmedia s'appuie sur la solution logicielle PTC Windchill qui organise le référentiel documentaire autour d'une architecture produit (BOM<sup>37</sup>). Le module de chargement de Windchill intègre une fonction update, qui permet d'établir une démarche de duplication de structure via import. Elle permet de préparer, puis importer la structure de l'architecture sur deux fichiers Excel : un pour les composants (Parts.xls) et un autre pour les liens entre les composants (Lien BOM.xls).

Figure 12 : Mécanisme de chargement existant pour DocEmedia

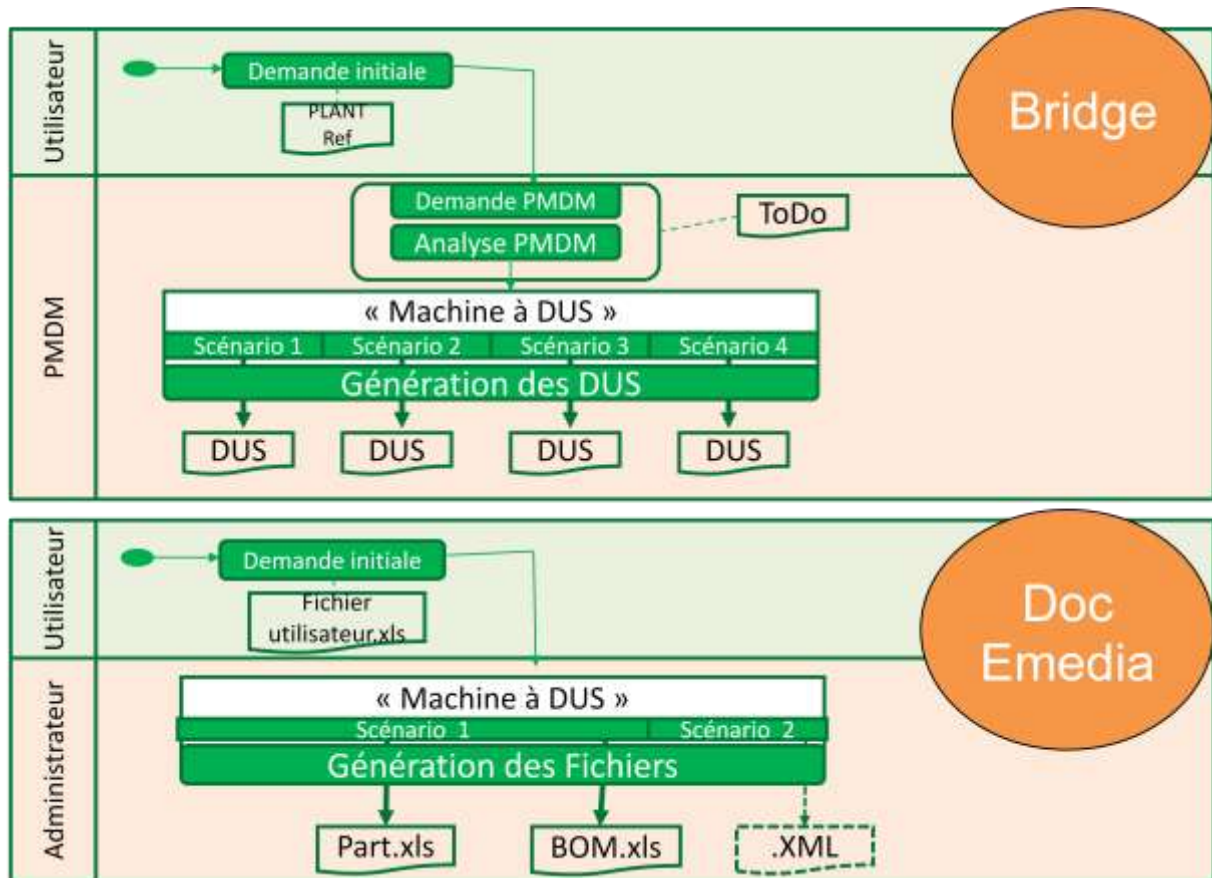


<sup>37</sup> BOM : Bill Of Material (liste des matériels)



## ANNEXE D : Les workflows

Figure 13 : Processus et scénarios identifiés



ToDo est une sheet Excel contenant les données d'entrée utile pour la génération des DUS.

PMDM : Plant Master Data Manager