

---

# PLASTICO : une illustration de démarche de développement conciliant agilité et urbanisation du système d'information

Jacques Simonin\* — Philippe Tanguy\* — Julie Gourmelen\*\*

\* Institut Télécom/Télécom Bretagne – Lab-STICC UMR CNRS 6285 - UEB  
Technopôle Brest Iroise  
F-29238 Brest cedex

{jacques.simonin, philippe.tanguy}@telecom-bretagne.eu

\*\* INSERM, U1018, Université Versailles Saint Quentin, Plateforme de Recherche  
"Cohortes en Population" - Centre de recherche en Épidémiologie et Santé des  
Populations, Hôpital Paul Brousse  
16 avenue Paul Vaillant-Couturier  
F-94800, Villejuif

julie.gourmelen@inserm.fr

---

*RÉSUMÉ. Le processus de développement de la plate-forme épidémiologique PLASTICO doit satisfaire deux objectifs à concilier : la durabilité de la plate-forme et la réactivité de son évolution par rapport aux exigences du client. Le premier objectif a nécessité la prise en compte de l'architecture, ou urbanisation, du système d'information supportant les processus d'étude épidémiologique de l'INSERM, client de la plateforme. Le second objectif a été atteint grâce à une approche agile limitée à quelques activités du développement considérées comme pertinentes par le client. Un bilan positif de l'adéquation de la démarche de développement à des exigences évoluant rapidement et à la durabilité de la plateforme PLASTICO a été dressé par l'INSERM.*

*ABSTRACT. The development process applied to the PLASTICO epidemiological platform should achieve two goals to be conflicting: the sustainability of the platform and reactivity to its evolution in relation to the customer requirements. The first goal has required taking into account the information system architecture, or urbanisation, which supports the epidemiological study processes of the INSERM that is the customer of the platform. The second goal has been achieved thanks to an agile approach limited to some development activities considered as relevant by the customer. A positive assessment of the appropriateness of the development approach to a requirements quick evolving and to the sustainability of PLASTICO has been achieved by the INSERM.*

*MOTS-CLÉS : processus de développement, système, agilité, système d'information, urbanisation.*

*KEYWORDS: development process, system, agility, information system, urbanisation.*

---

## 1. Introduction

Le contexte de l'article est le choix et l'expérimentation d'une démarche de développement choisie pour la plateforme PLASTICO (PLAteforme Scientifique et TechnIque d'aide à la gestion de COhortes et de grandes enquêtes de santé publique) de l'INSERM (Institut National de Santé Et de la Recherche Médicale) (Goldberg *et al.*, 2008). Cette démarche de développement de type Unified Process (Jacobson *et al.*, 1999) est itérative. L'architecture applicative est une activité centrale de la démarche de développement.

### 1.1. Architecture applicative d'un système

L'activité d'architecture applicative (ISO, 1995) (ou organique dans (Simonin *et al.*, 2011)) consiste dans la définition de couches logicielles et dans la définition des composants de chacune de ces couches. Les relations entre composants appartenant à une même couche logicielle ou à deux couches logicielles distinctes sont aussi conçues lors de cette activité. Une architecture applicative de type 3-tier peut être définie de façon classique (Steiert, 1998) à partir des couches suivantes :

- la couche logicielle « Présentation » associée à l'IHM (Interface Homme-Machine) de la plateforme,
- la couche logicielle « Application » comprenant les services offerts par la plateforme,
- la couche logicielle « Données » responsable de l'accès aux données de la base de données associée.

Ces couches logicielles sont complétées par la dépendance de la couche « Présentation » vis-à-vis de la couche « Application », qui elle-même dépend de la couche « Données ». Les relations de dépendance entre couches logicielles signifient que deux composants logiciels de deux couches différentes peuvent dépendre l'un de l'autre si les couches auxquelles ils appartiennent sont dépendantes. De plus, l'orientation de la dépendance des composants logiciels est cohérente avec l'orientation de la dépendance de leurs couches. Par exemple, un composant logiciel de la couche « Application » peut dépendre d'un composant logiciel de la couche « Données », mais l'inverse est impossible.

L'activité de codage succédant à l'activité d'architecture applicative correspond à l'implémentation des éléments structurant chaque composant logiciel. Ces éléments structurant un composant logiciel sont définis par le *framework* de la couche logicielle à laquelle il appartient (par exemple, dans le cas de PLASTICO, la structure interne d'un composant de la couche logicielle « Données » est définie par le *framework* Hibernate).

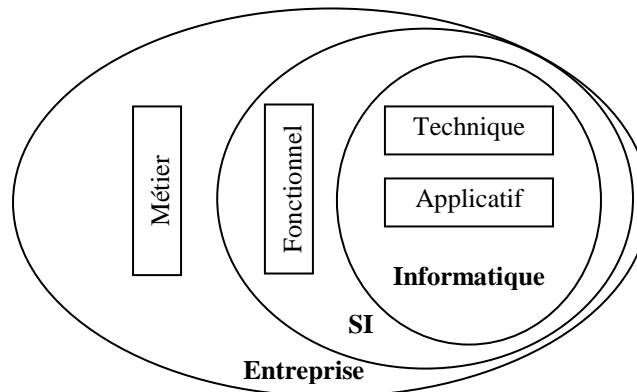
L'architecture applicative de la couche logicielle « Application » d'un système doit satisfaire les cas d'utilisation puisque les services sont définis par les scénarios nominaux qui illustrent ces cas d'utilisation lorsque leurs déroulements sont sans erreurs. Cette architecture peut devoir aussi être conforme à des préconisations

propres au SI (Système d'Information) qui contient le système. Cette conformité permet d'assurer une durabilité du développement si ces préconisations sont alignées avec la stratégie de l'entreprise. Cet alignement des systèmes d'un SI avec les objectifs de l'entreprise offre en effet des systèmes durables et donc un SI durable.

### 1.2. Architecture urbanisée d'un système d'information

L'architecture de l'entreprise ou EA (Enterprise Architecture) définie par Zachman (Zachman, 1987) est une approche qui permet de s'assurer de la conformité d'un SI avec la stratégie de l'entreprise. Le SI est défini dans la Figure 1 à partir des vues de l'entreprise proposées par l'urbanisme (Club URBA-SI, 2003) qui est le cadre de l'EA choisi en raison de la modélisation hiérarchique associée :

- la vue métier ou le « pourquoi ? » (Pourquoi un système est dans le SI ?),
- la vue fonctionnelle ou le « quoi ? » (Que réalise le système, qu'il soit informatisé, voire automatisé, ou non ?),
- la vue technique ou le « avec quoi ? » (Avec quoi est réalisé ce système, si il est informatisé, voire automatisé ?),
- la vue applicative ou le « comment ? » (Comment est réalisé ce système si il est informatisé, voire automatisé ?).



**Figure 1.** Vues d'urbanisme d'une entreprise, d'un SI et d'un système informatique

Les activités de développement décrites dans cet article sont celles de la démarche EA4UP (Enterprise Architecture for Unified Process) (Simonin *et al.*, 2008).

L'architecture fonctionnelle d'un système est contrainte dans la démarche EA4UP par la vue fonctionnelle du SI qui spécifie les fonctions du SI alignées avec la stratégie de l'entreprise. L'architecture applicative du système est impactée par

cette contrainte. En effet, les composants conçus lors de l'architecture applicative réalisent des fonctions du SI architecturant le système. L'architecture fonctionnelle du SI doit de plus supporter les processus métier de l'entreprise afin de renforcer la durabilité des vues de l'entreprise.

Les composants logiciels de la vue applicative du système vont bénéficier de l'architecture fonctionnelle du SI lors de la réutilisation des fonctions du SI réalisées par les composants logiciels. Néanmoins, ce développement urbanisé tenant compte de l'architecture fonctionnelle du SI ne prend pas en compte *a priori* les exigences du client lors du développement d'un système. L'approche agile permet cette prise en compte.

### **1.3. Agilité du développement d'un système**

Un développement agile permet au client d'exprimer son besoin lors du développement du système afin que le développeur le prenne en compte lors de courtes itérations.

Pour rappel, les quatre valeurs fondamentales du manifeste pour le développement Agile (Beck *et al.*, 2007) sont de valoriser :

- les individus et leurs interactions plus que les processus et les outils,
- des logiciels opérationnels plus qu'une documentation exhaustive,
- la collaboration avec les clients plus que la négociation contractuelle,
- l'adaptation au changement plus que le suivi d'un plan.

Une activité par nature agile est l'analyse des exigences fonctionnelles d'un système. La reformulation du besoin qui y est spécifiée permet en effet au client de stabiliser son besoin et au développeur de s'assurer de la compréhension de ce besoin. Cette compréhension est la première garantie d'un développement satisfaisant les exigences fonctionnelles du client. L'aspect agile de cette activité est en particulier souligné lors de la validation des exigences fonctionnelles faite lors de réunions mettant en présence le client et les développeurs (Paetsch *et al.*, 2003). Cet aspect est fondamental lors du développement de la plate-forme PLASTICO.

L'agilité lors des activités d'architecture et de codage est plus délicate, à l'exception de l'architecture applicative de la couche logicielle « Présentation ». Cette couche permet au client de visualiser le développement de ses exigences et donc de juger de leur pertinence par rapport à son besoin. C'est le principe de la conception centrée utilisateurs intégrée aux méthodes agiles (Schwartz *et al.*, 2009) telles que SCRUM (Schwaber *et al.*, 2012). L'activité d'architecture fonctionnelle et les activités d'architecture et de codage des autres couches logicielles sont généralement réservées aux développeurs du fait de la difficulté d'échanger avec le client à partir des livrables de ces activités. Ceci est en particulier significatif pour les activités d'architecture fonctionnelle et d'architecture applicative dirigées par les modèles. Il est intéressant de noter à ce propos l'émergence d'outils fondés sur une

vue intégrée de la modélisation comme, par exemple, pour les lignes de produit (Stoiber, 2009). Cette intégration de l'ensemble des diagrammes définissant les modèles du système à développer reste néanmoins réservée aux développeurs.

La recette des exigences fonctionnelles a aussi vocation à être agile. Elle consiste ici à vérifier la satisfaction des exigences fonctionnelles du client lors de la phase de test. Cette vérification s'accompagne généralement d'une qualification des cas d'utilisation, c'est-à-dire d'une évaluation par le client de l'importance de l'effort de test à faire pour chaque cas d'utilisation. La recette est double dans un développement agile car elle nécessite d'être réalisée par les développeurs et d'être conforme à la vision du client du système (Lindval *et al.*, 2004).

#### **1.4. Contribution**

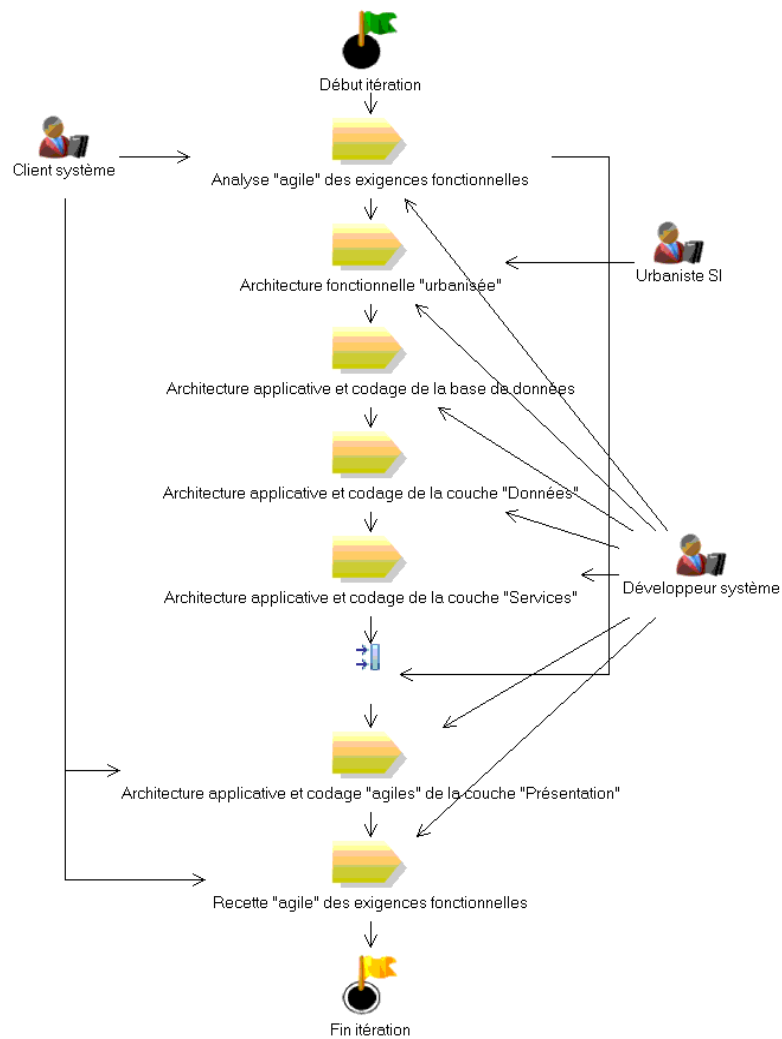
Une méthode de développement contraint par l'architecture cible d'un SI et bénéficiant d'une approche agile n'existe pas à notre connaissance. L'architecture d'entreprise est généralement considérée de façon limitée lorsqu'elle est un complément à une approche agile (Madison, 2010). Ces apports ciblent essentiellement la réutilisation de composants existants des différentes vues, en particulier ceux de la vue technique.

La contribution consiste en une utilisation conciliée de deux approches *a priori* contradictoires : la prise en compte de l'urbanisme lors du développement d'un système et la mise en œuvre d'une approche agile restreinte à quelques activités du développement. La motivation est d'obtenir la durabilité du système grâce à la prise en compte de l'architecture fonctionnelle urbanisée du système d'information et d'impliquer le client lors de certaines activités du développement de manière agile afin d'être le plus réactif possible à ses demandes d'évolution.

La partie suivante présente les principes de notre démarche de développement recouvrant à la fois l'approche agile mais aussi les contraintes de l'urbanisation du SI. La démarche et ses principes sont illustrés par le développement de la plateforme PLASTICO. Dans la partie 3, un bilan quantitatif et qualitatif de la mise en œuvre réelle de notre démarche de développement pour la plateforme PLASTICO est présenté. Une conclusion et des perspectives sont proposées dans la partie 4.

## 2. Démarche de développement agile et urbanisé d'un système

### 2.1. Présentation générale de la démarche de développement



**Figure 2.** Démarche de développement itératif conciliant agilité et urbanisation

La démarche de développement (à l'exception des activités d'analyse des exigences non fonctionnelles et d'architecture technique traitées en amont),

conciliant agilité et contraintes d'urbanisation, est décrite dans la Figure 2 par un diagramme de tâches conforme au langage xSPeM (Bendraou *et al.*, 2007). Le choix de ce langage est justifié par l'exécutabilité des tâches. C'est particulièrement utile dans le cas de tâches du processus de développement fondées sur des transformations de modèles qui peuvent être codées et exécutées (Simonin *et al.*, 2011). Dans ce processus de développement, le rôle *Client système* caractérise l'agilité alors que celui d'*Urbaniste SI* souligne la prise en compte de l'urbanisme.

L'apport de l'agilité est classique pour les activités d'analyse des exigences fonctionnelles et de recette du système par le client. L'approche agile proposée dans notre démarche recouvre en complément l'activité d'architecture et de codage de la couche « Présentation ». La contrainte de l'urbanisation du SI impacte l'activité d'architecture fonctionnelle du système dans notre démarche. Cet impact a des conséquences sur les résultats de l'activité d'architecture applicative des couches « Application » et « Données » qui en découle. La durabilité de la plateforme ciblera donc les services conçus dans la couche « Application » et l'accès aux données de la couche « Données », sans oublier l'architecture de la base de données.

## **2.2. Activités agiles du développement**

L'objectif est d'impliquer le client durant le développement afin d'être le plus réactif possible à ses demandes d'évolution. Comme dans tous les projets, les exigences du client peuvent évoluer et il nous a semblé important de prendre en compte cette considération dès le début. La solution est une approche agile restreinte à certaines activités du développement décrit dans la Figure 2. La démarche choisie est fondée sur une approche agile mise en œuvre lors de l'architecture et du codage de la couche « Présentation » et d'intégrer la recette du système par le client à cette activité logicielle. Les anomalies relevées lors de la recette peuvent, si c'est utile, être reportées sur les couches « Application » et « Données » qui sont alors corrigées pour l'itération suivante. L'analyse des exigences fonctionnelles complètent l'ensemble des activités « agiles » de notre démarche.

### **2.2.1. Analyse « agile » des exigences fonctionnelles**

L'agilité est mise en œuvre à chaque début d'itération lors de l'analyse des exigences fonctionnelles. Cette mise en œuvre est fréquente lors d'un développement de système, y compris non agile, puisque l'activité d'analyse est une occasion privilégiée d'échanges entre les développeurs et le client afin de bien cerner le besoin du client.

Ces échanges se situent autour des cas d'utilisation du système et des scénarios qui illustrent, scénarios d'exception inclus, ou autour d'une maquette dite creuse. La maquette illustre les scénarios par des enchaînements d'écrans. L'aspect creux de cette maquette signifie l'absence de code permettant d'accéder aux données réelles. Le choix d'échanges entre le client et les développeurs autour d'une maquette creuse

nécessite cependant une mise à jour simultanée des scénarios correspondants. Ces scénarios sont en effet la source des activités d'architecture fonctionnelle et d'architecture applicative. Toujours lors de l'analyse des exigences fonctionnelles, la définition des entités participantes aux cas d'utilisation, fournis par le client du système, et de leurs relations est spécifiée par le développeur du système car elle est l'élément d'entrée pour la conception d'architecture.

L'agilité mise en œuvre lors de l'architecture et le codage de la couche « Présentation » incite à privilégier les échanges entre le client et les développeurs autour des cas d'utilisation et de leurs scénarios. Ces échanges ont été concrétisés lors de réunion de validation des cas d'utilisation et de leurs scénarios comme l'approche agile l'exige.

### 2.2.2. *Architecture et codage « agiles » de la couche « Présentation »*

L'originalité de notre démarche pour son aspect agile est de réaliser simultanément les activités d'architecture et de codage de la couche « Présentation » et l'activité de recette des exigences fonctionnelles du système.

La prise en compte de nouvelles exigences ou l'évolution d'exigences déjà exprimées donne lieu à la création ou la modification d'un cas d'utilisation qui permettra de concevoir et de développer la couche « Présentation » correspondante.

Les écrans de la couche « Présentation » architecturés et codés de manière agile impactent les couches « Application » et « Données » du système. Cet impact doit être pris en compte et peut aller jusqu'à l'évolution du schéma de la base de données. Dans la pratique, c'est le cas le plus courant induit par l'appréciation donnée par le client des écrans en fonction du critère de satisfaction de ses exigences fonctionnelles. En effet, chaque écran fait appel à un ou plusieurs services de la couche « Application ». Le résultat de l'appel du service est donc visualisé et permet ainsi un jugement rapide du client sur la satisfaction de ses exigences.

L'impact de l'agilité au niveau de la mise à niveau des couches « Application » et « Données » se fait dans la même itération. Ces itérations courtes (deux semaines pour un déploiement dans le cas de PLASTICO) qui succèdent à une première itération plus longue ciblant le développement d'un noyau logiciel stable composé des couches « Application » et « Données » et de la base de données permettent un rythme soutenu pour l'évolution du système.

### 2.2.3. *Recette « agile » des exigences fonctionnelles*

La recette est double dans notre démarche de développement, comme pour tout développement agile (cf. §1.3.), puisque :

- la recette du codage des couches « Application » et « Données » est effectuée par l'équipe de développement en ayant la charge,
- la recette du codage de la couche « Présentation » est faite directement par le client.



La première recette est fondée sur les scénarios d'utilisation du système. Chaque scénario est une illustration d'un cas d'utilisation spécifié en analyse des exigences fonctionnelles (hors exceptions). Cette recette faite par les développeurs des couches « Application » et « Données » consiste pour chaque service de la couche « Application » à exécuter le scénario correspondant, sans ou avec levée d'exception.

Une fois la couche « Présentation » architecturée et codée, avec son éventuel impact sur les deux autres couches, sa recette est réalisée par le client. La recette est aussi fondée sur les scénarios illustrant les cas d'utilisation spécifiés en analyse des exigences fonctionnelles. Toute modification d'un écran de la couche « Présentation » est répercutée en analyse lors de l'itération suivante. Cette répercussion cible d'abord les cas d'utilisation. Une conséquence est la facilitation de l'évolution des exigences fonctionnelles spécifiées par le client.

Il est utile de préciser que la seconde recette complète la première puisque les services utilisés par la couche « Présentation » peuvent être les sources des anomalies rencontrées. Le cas d'anomalie le plus fréquent est une compréhension partielle du cas d'utilisation spécifié en analyse des exigences fonctionnelles ou une spécification partielle de ce même cas d'utilisation.

### **2.3. Urbanisation fonctionnelle du SI**

Le système a pour objectif de s'intégrer dans un SI dédié aux études épidémiologiques. Cette intégration signifie le respect de la vue fonctionnelle de ce SI qui supporte une vue métier cible de l'entreprise conforme à sa stratégie (cf. Figure 1) (Sassoon, 1998) (Longépé, 2001).

#### **2.3.1. Vue métier cible supportée par le SI**

La vue fonctionnelle du SI est alignée avec la vue métier conçue par le client. Une vue métier définie comme cible, car alignée avec la stratégie, est en effet une solution efficace pour faciliter l'évolution du SI (Nurcan *et al.*, 1999).

Pour PLASTICO, les processus de la vue métier conçus par les chercheurs épidémiologistes de l'INSERM sont :

- le processus métier *Etudier la faisabilité de l'étude épidémiologique* traitant la faisabilité d'une étude épidémiologique suite à un pilote réalisé sur un ensemble réduit de données épidémiologiques ;
- le processus métier *Mettre en œuvre l'étude épidémiologique* permettant la livraison d'une base de données conçue à partir d'une étude épidémiologique. La livraison signifie un contrôle de conformité scientifique et technique de la base de données ;
- le processus métier *Exploiter la base de données* où les données extraites de l'étude épidémiologique sont transmises au chercheur pour ses travaux.

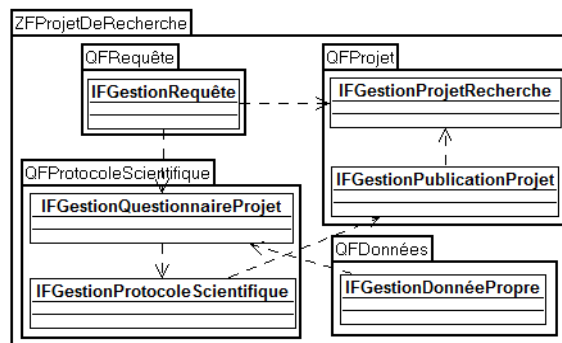
### 2.3.2. Vue fonctionnelle cible du SI

La vue fonctionnelle du SI qui doit contraindre l'architecture fonctionnelle de système est structurée de façon hiérarchique (Wegman *et al.*, 2007) suivant un PLU (Plan Local d'Urbanisme) fonctionnel (Simonin *et al.*, 2010) composé de façon exhaustive :

- des fonctions du SI et de leurs répartitions entre différents îlots fonctionnels qui les contiennent.
- de la répartition des îlots fonctionnels dans des quartiers fonctionnels, eux-mêmes répartis dans des zones fonctionnelles.

Les trois zones fonctionnelles de PLASTICO préconisées sont :

- la zone *ZFFonctionsTechniques* permettant à un administrateur de gérer la plateforme PLASTICO,
- la zone *ZFEtudeEpidémiologique* dédiée aux études épidémiologiques de cohortes faites par l'INSERM,
- la zone *ZFProjetDeRecherche* dédiée à la gestion des requêtes de données épidémiologiques faites par les projets de recherche auprès d'organismes de santé tels que la CNAMTS (Caisse Nationale d'Assurance Maladie des Travailleurs Salariés) et supervisées par l'INSERM.



**Figure 3.** Zone fonctionnelle ZFProjetDeRecherche

La zone *ZFProjetDeRecherche* qui illustre la vue fonctionnelle cible du SI dédié aux études épidémiologiques (cf. Figure 3) est composée des quartiers fonctionnels :

- *QFRequête* restreint à l'îlot fonctionnel *IFGestionRequête* qui contient les fonctions d'extraction et de transmission des données épidémiologiques ;
- *QFProtocoleScientifique* composé des îlots fonctionnels *IFGestionQuestionnaireProjet*, comprenant les fonctions de gestion des listes de

variables et d'individus qui sont les cibles de la requête, et *IFGestionProtocoleScientifique* qui cible les fonctions de gestion d'une demande d'un projet de recherche liée à une liste de variables et d'individus ;

– *QFProjet* qui contient les îlots fonctionnels *IFGestionProjetRecherche*, composé des fonctions de gestion d'un projet de recherche déposant une demande, et *IFGestionPublicationProjet* chargé de la publication, ou validation, d'un projet de recherche par l'INSERM ;

– *QFDonnées* défini par l'îlot fonctionnel *IFGestionDonnéePropre* qui cible la gestion d'une donnée propre à un projet de recherche.

### 2.3.3. Alignement des modèles de la vue fonctionnelle et de la vue métier cible

L'alignement cible dans cette section des modèles de la vue fonctionnelle et de la vue métier et non la stratégie comme précédemment. La vue fonctionnelle doit être alignée avec la vue métier. Cet alignement de modèles garantit que le PLU fonctionnel est durable puisque la vue métier est conforme à la stratégie de l'entreprise.

Par exemple, pour la vue fonctionnelle cible du SI des études épidémiologiques :

– la zone *Etude épidémiologique* est alignée avec les processus métier *Etudier la faisabilité de l'étude épidémiologique* et *Mettre en œuvre l'étude épidémiologique*,

– la zone *Projet de recherche* est alignée avec le processus métier *Exploiter la base de données*.

Un îlot fonctionnel est aligné dans notre approche avec un objet métier produit par une activité d'un processus métier. Par exemple, l'îlot fonctionnel *IFGestionRequête* est aligné avec l'objet métier *Requête* produit par l'activité *Extraire les données* du processus *Exploiter la base de données*.

L'alignement des dépendances entre îlots fonctionnels et des séquences d'activités métier complètent ce travail d'alignement du modèle de la vue fonctionnelle du SI par rapport au modèle de la vue métier de l'entreprise. Le développement de système dans le SI des études épidémiologiques est fondé sur cette vue fonctionnelle stabilisée du SI.

## 3. Bilan du développement agile et urbanisé de PLASTICO

Le schéma de développement itératif de PLASTICO comprend une première itération longue ciblant l'initialisation des couches « Application » et « Données », fondée sur la vue fonctionnelle du SI puis un nombre d'itérations rapides liées à l'évolution des exigences fonctionnelles dans une approche agile ciblant l'architecture applicative et le codage de la couche Présentation doublés d'une recette du système.

### 3.1. Le développement de la plateforme PLASTICO en chiffres

La première version de la plateforme PLASTICO comprend :

- l’architecture technique de la plateforme, qui est considérée comme complexe du fait des *frameworks* mis en œuvre,
- les fonctionnalités de gestion du catalogue, de gestion d’un projet de recherche demandeur de données épidémiologiques, de gestion de la demande de données par ce projet ainsi que l’extraction des données épidémiologiques et la transmission de ces données au projet de recherche.

Le Tableau 1 permet de quantifier l’activité d’analyse des exigences fonctionnelles réalisée de manière agile avec l’INSERM. Cette quantification est fondée sur la quantité de concepts instanciés lors du développement de la première version de PLASTICO.

Concept	Quantité
Cas d’utilisation	8
Scénario illustrant les cas d’utilisation	64
Entité participante aux cas d’utilisation	23
Attribut d’entité participante aux cas d’utilisation	74
Associations d’entités participantes aux cas d’utilisation	33

**Tableau 1.** *Eléments quantitatifs de l’analyse « agile » des exigences fonctionnelles réalisée pour la première version de la plateforme PLASTICO*

Le Tableau 2 ci-après permet de quantifier l’activité d’architecture applicative des couches « Application » et « Données » ainsi que de la base de données.

Concept	Quantité
Composants de la couche « Application »	8
Services offerts par les composants de la couche « Application »	102
Composants de la couche « Données »	39
Données physiques	45
Attributs des données physiques (sans les clés étrangères)	120
Relations entre données physiques (clés étrangères)	48

**Tableau 2.** *Eléments quantitatifs de l’architecture applicative des couches « Application » et « Données » ainsi que de la base de données relationnelle de la première version de la plateforme PLASTICO contrainte par l’urbanisme du SI épidémiologique.*

Les 102 services conçus dans la couche « Application » correspondent aux 64 scénarios spécifiés en analyse fonctionnelle et à des consultations de données utiles à la couche « Présentation » (consultation des projets de recherche, etc.). Les 39 composants de la couche « Données » sont de type DAO (Data Access Object).

Activité	Nombre de jours
Analyse des exigences fonctionnelles	12,5
Architecture fonctionnelle	11
Architecture applicative des couches « Application » et « Données »	7,5
Codage des couches « Application » et « Données » et de la base de données	19,5
Tests des couches « Application » et « Données »	16,5
Architecture applicative et codage de la couche « Présentation » + recette des exigences fonctionnelles	207

**Tableau 3.** Coût de développement de la plate-forme PLASTICO en jours.

L'équipe de développement pour l'architecture applicative, le codage et la recette qui nous intéressent particulièrement dans notre démarche est constituée :

- d'un architecte fonctionnel et applicatif expert qui est aussi un codeur débutant pour les couches « Application » et « Données », ainsi que pour la base de données relationnelle,
- d'un architecte et codeur senior pour la couche « Présentation », mais néanmoins débutant sur le *framework* SmartGWT (Jaber, 2010) choisi pour cette couche par le client,
- d'une spécialiste métier de l'INSERM incontournable dans l'approche agile mise en œuvre.

Le coût à ajouter est celui de la spécialiste métier de l'INSERM. Il a été de 12 jours pour l'analyse des exigences fonctionnelles (presque identique au coût côté équipe de développement) et de 42 jours pour la recette.

Il est d'ores et déjà possible de conclure que le coût du développement (architecture applicative + codage + tests) de la couche «Présentation» couplé avec la recette est environ quatre fois plus important que celui des couches « Application » et « Données ». Cette comparaison souligne l'apport de l'architecture fonctionnelle du SI épidémiologique pour ces deux couches et l'impact de l'approche agile sur le coût du développement de la couche « Présentation ».

### **3.2. Bilan qualitatif de la plateforme PLASTICO**

Pour l'INSERM, le bilan qualitatif de cette approche combinant les contraintes de l'urbanisation et l'approche agile est pour l'analyse des exigences fonctionnelles :

- un travail en commun (INSERM et développeurs) sur le vocabulaire nécessaire pour les cas d'utilisation spécifiés de façon agile ;
- un besoin de l'INSERM précisé tout au long du développement des services de la plate-forme du fait de l'agilité. La stabilité du besoin de l'INSERM apparaît ainsi dans notre démarche plus longue à obtenir en comparaison de la réactivité des analystes de l'équipe de développement de PLASTICO.

Concernant l'architecture applicative et le codage :

- pour les couches « Application » et « Données » dont un noyau dur est déjà architecturé et codé conformément à l'urbanisme du SI dans la première itération longue, les évolutions incluses dans les itérations courtes sont jugées par l'INSERM comme très rapides ;
- le rythme de travail choisi pour la couche « Présentation » fondée sur l'approche agile est assez efficace côté INSERM (au moins une réunion par semaine avec le développeur pour répondre à ses questions, faire la recette des nouveaux écrans, discuter des évolutions possibles, modifier les cas d'utilisation, modifier les écrans) ;
- le rythme de déploiement pour la couche « Présentation » est en moyenne bimensuel ;
- un travail régulier entre l'INSERM et le développeur de la couche « Présentation » est nécessaire dans notre démarche partiellement agile. Cela permet en effet d'expliquer au développeur les souhaits de l'INSERM concernant le visuel, mais aussi le but des diverses fonctionnalités et de réfléchir ensemble aux différentes solutions pour les problèmes rencontrés.

Les services de la plate-forme PLASTICO ont été présentés lors de démonstrations (fin 2012 et début 2013) à des institutions détentrices de données épidémiologiques. L'objectif était de leur montrer l'avancement du projet afin de poursuivre le développement de PLASTICO en intégrant des services de transmission des données de ces institutions vers la plateforme PLASTICO, puis aux équipes de recherche.

### **4. Conclusion et Perspectives**

Dans l'aspect agile de notre démarche, il est d'abord utile de souligner la réactivité due à la participation du client lors de l'analyse des exigences. C'est en effet lui qui gère la complexité de son besoin fonctionnel. L'analyste n'a ainsi plus

qu'à reformuler ces spécifications clarifiées. Le développement de la couche « Présentation » et la recette offrent de plus au client des possibilités de faire évoluer ses exigences fonctionnelles. La stabilisation de ses exigences est ainsi facilitée.

L'utilisation de l'urbanisation fonctionnelle du SI permet de plus un gain de temps et une réactivité lors du développement de la couche « Données » et de la base de données relationnelle. Ceci nécessite cependant une cohérence de l'architecture du SI avec les processus métiers conformes à la stratégie de l'entreprise qui apportent au SI une durabilité.

La démarche de développement proposée a permis enfin de montrer la possibilité de concilier un développement réactif aux évolutions des exigences du client grâce à l'aspect agile de certaines de ses activités et un développement durable suite à la prise en compte de l'urbanisme fonctionnel du SI.

Une première perspective à court terme est l'étude de l'alignement de modèle de la vue applicative d'un système par rapport à un modèle de sa vue fonctionnelle. Cette étude devrait cibler l'alignement des services de la couche « Application » d'un système par rapport aux îlots fonctionnels de ce système. Il est intéressant de souligner que cet alignement de modèles permettrait dans un deuxième temps de vérifier l'adéquation de la vue applicative d'un système avec la vue fonctionnelle cible du SI.

Une seconde perspective est une étude d'alignement permettant de réaliser un bilan de l'aspect agile du développement d'un système. L'alignement du modèle de la couche « Présentation » de la vue logicielle du système par rapport à un modèle des exigences fonctionnelles du client formulées dans les cas d'utilisation pourrait permettre d'évaluer la qualité de la mise en œuvre de l'approche agile.

## 5. Bibliographie

Beck K. *et al.*, « The Agile Manifesto », <http://agilemanifesto.org/>, 2001.

Bendraou R., Combemale B., Crégut X., and Gervais. M.-P., « Definition of an eXecutable SPEM2.0 », *14th APSEC*, Japan, IEEE Computer Society, 2007.

Club URBA-SI, *Pratiques de l'urbanisme des systèmes d'information en entreprises*, Publibook, 2003.

Goldberg M., Quantin C., Guéguen A., Zins, M., « Bases de données médicoadministratives et épidémiologie : intérêts et limites », *Courrier des statistiques*, n° 124, 2008.

International Organization of Standardization, « Software Life Cycle Processes », *ISO/IEC 12207*, 1995.

Jaber S., *Programmation GWT 2: Développer des applications RIA et Ajax avec Google Web Toolkit*, Eyrolles, 2010.

- Jacobson I., Booch G., Rumbaugh J., *The Unified Software Development Process*, Addison-Wesley, 1999.
- Lindvall M., Muthig D., Dagnino A., Wallin C., Stupperich M., Kiefer D., May J., Kahkonen T., « Agile software development in large organizations », *Computer*, n° 37, 2004, p. 26-34.
- Longépé C., *Projet d'urbanisation du système d'information – Démarche pratique avec cas concret*, Dunod/01 Informatique, 2001.
- Madison J., « Agile-Architecture Interactions », *IEEE Software*, 2010.
- Nurcan S., Barrios J., Grosz G., Rolland C., « Change process modelling using the EKD - Change Management Method », *7th European Conference on Information Systems*, Copenhagen, Denmark, 1999.
- Paetsch F., Eberlein A., Maurer F., « Requirements Engineering and Agile Software Development », *12th IEEE international Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises*, Linz, Austria, 2003, p. 308-313.
- Sassoon J., *Urbanisation des systèmes d'information*, Hermès, 1998.
- Schwaber K., Sutherland J., « Scrum Guide », SCRUM alliance, <http://www.scrum.org/>, 2010.
- Schwartz L., Vergnol L., Gronier G., Vagner A., Altenburger T., Battisti S., « Comment concilier agilité et conception centrée utilisateurs dans un projet de développement ? », *21st Conference on Association Francophone d'Interaction Homme-Machine (IHM)*, Grenoble, France, 2009, p. 237-240.
- Simonin J., Alizon F., Deschrevel J.-P., Le Traon Y., Jézéquel J.-M., Nicolas B., « EA4UP: an Enterprise Architecture-Assisted Telecom Service Development Method », *12th IEEE International EDOC Conference*, München, Germany, 2008.
- Simonin J., Picouet P., Jézéquel J.-M., « Conception fonctionnelle de services d'entreprise fondée sur l'alignement entre cœur de métier et système d'information », *Ingénierie des systèmes d'information*, 2010, p. 37-61.
- Simonin J., Beugnard A., Nédélec R., « Processus de développement de système contraint par l'urbanisation d'un système d'information », *INformatique des ORganisations et Systèmes d'Information et de Décision (INFORSID)*, Lille, France, 2011.
- Steiert, H.-P., « Towards a Component based n-Tier C/S-Architecture », *International Workshop on Software Architecture (ISAW)*, Orlando, Florida USA 1998.
- Stoiber R., Glinz M., « Modeling and Managing Tacit Product Line Requirements Knowledge », *2nd International Workshop on Managing Requirements Knowledge (MaRK)*, Atlanta, USA, 2009.
- Wegmann A., Regev G., Rychkova I., Lë L.S., De La Cruz J.D., Julia P., « Business and IT Alignment with SEAM and Enterprise Architecture », *11th IEEE International EDOC Conference*, Annapolis, USA, 2007.
- Zachman J.A., « A Framework for Information Systems Architecture », *IBM Systems Journal* 26, n° 3, 1987, p. 276-292.