

Scope document : Environnement de test représentatif

Note, relative à la BCSS et à ses partenaires, décrivant les principes à respecter, par les chefs de projets PPKB, dans le cadre de la réalisation de tests entre institutions.

Auteurs/Réviseurs

Acronyme	Prénom Nom	Institution	Fonction
R06	Mathieu Delehay	BCSS	Chef de projet

Revision history

Date	Version	Description	Auteur
14/03/17	0.1	Version initiale	R06
16/03/17	0.2	Mises à jour suite à retour TO, ajout procédure gestion projets et rapport audit processus mise en production	R06
16/03/17	0.3	Ajout validation formelle des tests en ACPT et vérification des traitements lors des premières exécutions en PROD	R06
28/03/17	0.4	Remplacement flux invalidité par assurabilité / invalidité, mise à jour section 4.3.2 concernant tests suite à correction	R06
24/04/17	0.5	Mise à jour suite à réunion du 7/04/17 avec TO	R06
22/05/17	0.6	Préparation GT relations informatiques du 1/06/17	R06

Documents référencés

	Description	Auteur
[1]	Fiche bedrijfsproces 304040: "Test, validatie en in productie stellen van nieuwe of aangepaste applicaties/diensten" (P-304040 - TestValiProdWip_integratie - 0.0 - NL.doc)	BCSS
[2]	Scope document : Environnement de test représentatif – Annexe (CBSS_Scope_environnement_test-Annexe.docx)	R06
[3]	Enquête de satisfaction des clients BCSS - Plan d'actions (Plan_action_enquete_04102016.docx)	Claudia Laerman
[4]	PRINCE2 2009 Edition Guide de Poche (2009, Van Haren Publishing)	Bert Hedeman, Ron Seegers
[5]	BCSS Gestion de projets et cycle de développement d'un logiciel (Projectbeheer en Softwareontwikkelingslevenscyclus KSZ v7.1_FR.doc)	BCSS
[6]	Rapport de réunion du jeudi 20/10/2016 Mise en production d'application (2016.10.20_PV_M Delehay_mise en production.doc)	Smals, BCSS
[7]	Foundations of Software Testing ISTQB Certification (2012, Cengage Learning, 3 rd edition)	Rex Black, Erik Van Veenendaal, Dorothy Graham

Table des matières

1.	Acronymes	2
2.	Objectif document.....	3
3.	Terminologie& Concepts	4
4.	Description des services existant	7
4.1.	Introduction.....	7
4.2.	Liste des services	7
4.3.	Observation des pratiques actuelles en termes de test.....	13
4.3.1.	Environnements de test	13
4.3.2.	Méthodes qualité	13
4.3.3.	Origine des données	14
4.3.4.	Communication avec les partenaires	15
5.	Pistes d'améliorations	15
5.1.	Introduction.....	15
5.2.	Environnements de test	15
5.3.	Méthodes qualité	15
5.4.	Origine des données.....	16
5.5.	Communication avec les partenaires	16
6.	Conclusion	16

1. Acronymes

ACPT	Environnement d' ACcePT ation
APA _{fr} , THAB _{nl}	A ide aux P ersonnes Â gées
BCSS _{fr} , KSZ _{nl}	B anque C arrefour de la S écurité S ociale
BUC	B usiness U nit C ases
Cl _{fr} , IC _{nl}	C ontrôle d' I ntégration
CIN _{fr} , NIC _{nl}	C ollège I ntermutualiste N ational
CPAS _{fr} , OCMW _{nl}	C entre P ublic d' A ction S ociale
DEV	Environnement de DE VeLoppement
DGPH	D irection G énérale des P ersonnes H andicapées
DmfA	D éclaration m ultifonctionnelle / multifunctionele A angifte
Dimona	D éclaration i mmédiate / o nmiddellijke a angifte
INASTI _{fr} , RSVZ _{nl}	I nstitut N ational d' A ssurances S ociales pour T ravailleurs I ndépendants
Famifed	Agence f édérale pour les allocations f amiliales
LDM	L ot D e M essages
NBIS	N uméro B IS
OA _{fr} , VI _{nl}	O rganisme A ssureur
ONVA _{fr} , RJV _{nl}	O ffice N ational des V acances A nnuelles
ONEM _{fr} , RVA _{nl}	O ffice N ational de l' E Mploi
ONSS _{fr} , RSZ _{nl}	O ffice N ational de S écurité S ociale
OP _{fr} , UI _{nl}	O rganisme de P aie M ent

PID	Project Initiation Document
PPKB	Programma Project Klant Beheer
PROD	Environnement de PRODUCTION
RBIS	Registre BIS
RGTI _{fr} , ARZA _{nl}	Registre Général des Travailleurs Indépendants
RN _{fr} , RR _{nl}	Registre National
RRAD	Registre des RADiés
SIGeDIS	Sociale Individuele Gegevens / Données Individuelles Sociales
SOA	Service Oriented Architecture
SFP _{fr} , FPD _{nl}	Service Fédéral des Pensions
SPF Sécurité Sociale _{fr} , FOD Sociale Zekerheid _{nl}	Service Public Fédéral Sécurité Sociale
TEST	Environnement de TEST
TSS	Technical Service Specification
XML	Extensible Markup Language
XSD	XML Schema Definition

2. Objectif document

L'objectif de ce scope document est de définir la manière dont la BCSS, en particulier l'équipe de chefs de projets PPKB¹, peut mettre en place un environnement de test représentatif qui permettra aux partenaires externes de disposer, par source authentique, d'environnements de test² configurés avec cas de test et scénarios de base.

En effet, dans le cadre du plan d'action rédigé suite à l'enquête de satisfaction 2016, le point suivant a été prévu ([3]) : « *En réponse à la demande de disposer par source authentique d'environnements de test avec des données de test de base qui pourraient être réinitialisées en vue de réappliquer les scénarios de test, une analyse est en cours. Un document scope avec planning sera présenté au GT 'Relations informatiques' lors du 1^{er} semestre 2017.* »

Lors de l'enquête de satisfaction 2016, il est apparu que, dans un certain nombre de cas (mise à disposition d'un nouveau service, abonnement d'un partenaire à un service existant, maintenance, correction...), aucun scénario de test représentatif ne pouvait être livré systématiquement aux utilisateurs, laissant ces derniers dans l'incapacité de vérifier si la demande avait été correctement effectuée, dans le cadre du service qu'ils utilisent, et si aucune erreur n'était à prévoir par la suite (étape de validation).

De manière générale, la validation d'une application par un partenaire externe a lieu dans l'environnement d'acceptation (ACPT). La méthodologie PRINCE2 prévoit que l'utilisateur principal « *fournit les exigences qualité du client* » et « *fournit les critères d'acceptation* », de même qu'il « *formalise l'acceptation du produit du projet* » ([4]). Dès lors, on peut considérer que la mise en place d'un environnement de test et data sets réutilisables, ainsi que la réalisation avec succès des tests par le client, permettent de vérifier ces critères d'acceptation.

De même, la procédure gestion de projets BCSS ([5]) prévoit les points suivants concernant les tests :

- Lors de la phase 3 (Analyse du projet), qui suit la mise à disposition du PID, il convient d'établir « *une première version du **plan de test**, dans lequel il (ndr : le test manager) donne un aperçu des types de tests qui seront exécutés – tests fonctionnels et non*

¹ Programma Project Klant Beheer

² Par « environnements de test », on comprend généralement les environnements TEST et ACPT (plus rarement l'environnement DEV).

fonctionnels tels que la performance, load, security... –. Il décrit à un haut niveau les scénarios de test y afférents ainsi que leur condition de réussite. Il se base à cet effet sur les use cases qui décrivent les besoins fonctionnels et sur les besoins non fonctionnels issus du PID (ndr : et validés par le client). ».

- La phase 4 (Développement du projet) prévoit la réalisation de tests unitaires, l'élaboration de test cases et test data, ainsi que le test déploiement (mise en DEV/TEST). Des tests d'intégration et vérifications sont également réalisés au cours de cette phase, qui se termine par la mise en acceptation de l'application.
- Enfin, la phase 5 (Réception du projet) prévoit des tests d'acceptations par les utilisateurs, sur base de scénarios de test fonctionnels (basés sur les use cases). Cette phase se termine par la mise en production de l'application, suivie d'une vérification des traitements lors des premières exécutions en production.

En ce qui concerne les environnements DEV et TEST, ceux-ci sont réservés aux tests en cours de développement (ou d'intégration), n'impliquant pas nécessairement les partenaires externes étant donné qu'il s'agit d'environnements par définition peu stables. L'environnement DEV, s'il est utilisé, est strictement à réserver aux tests internes³.

L'objectif du présent document est d'uniformiser les pratiques existantes et faire en sorte qu'un set minimum de données réutilisables soient disponibles pour la majorité des services BCSS (ou, du moins, les plus importants d'entre eux). En effet, dans ce cadre, la BCSS impose un certain nombre de bonnes pratiques (reprises dans les procédures [1] et [5]). La manière dont les services sont testés, concrètement, étant souvent non uniforme et inégale en termes de qualité, il convient de généraliser ces recommandations à l'ensemble des applications.

En termes de préparation, un ensemble d'interviews de chefs de projets, responsables des services, a été mené afin de recenser la situation « AS IS »⁴ en matière de testing. L'annexe [2] reprend un résumé, par service, des entretiens réalisés. A la section 4.2 du présent document, un tableau effectue la synthèse des pratiques existantes. Ce document a pour but de dégager des pistes d'améliorations, qui pourront constituer le « TO BE »⁵ des services en question.

Enfin, étant donné la coexistence, à la BCSS, de plusieurs technologies (IHFN « in-house », A1, SSDN, LDM, SOA...), il convient de déterminer quelles sont celles à privilégier à l'avenir, en matière de testing. En particulier, les technologies les plus récentes, basées sur le standard XML (LDM, SOA), seront à favoriser par rapport aux technologies plus anciennes (sauf dans le cas où aucun service XML équivalent n'est disponible).

Après un rappel des concepts et définition des termes usuels relatifs au testing, à la section 3, nous reprenons, à la section 4, la liste des principaux services existants ainsi que l'observation des pratiques actuelles. Dans la partie 5, nous donnons des pistes d'améliorations des pratiques de test mises en œuvre à la BCSS.

3. Terminologie & Concepts

A présent, nous définissons brièvement les principaux concepts qui seront utilisés dans ce document.

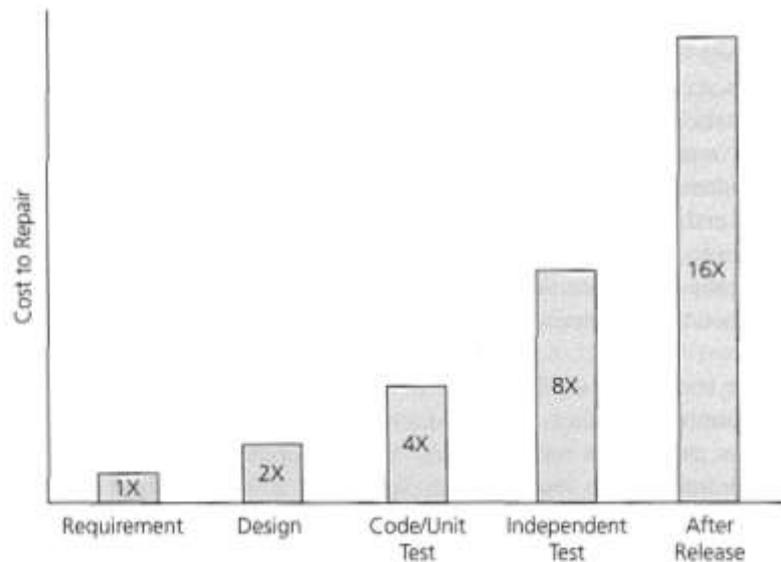
³ Bien souvent, DEV et TEST constituent un seul et unique environnement.

⁴ Avant amélioration

⁵ Situation après amélioration

D'après [7], un « acteur » (chef de projet, analyste, programmeur...) réalise une faute (error), comme l'oubli d'un requirement, contrôle logiciel... ce qui résulte en l'introduction d'un défaut (defect) dans l'application (en l'occurrence l'absence d'implémentation du requirement, contrôle, etc.) Lors de l'exécution, sous certaines conditions, le défaut pourra résulter en un comportement inattendu, en particulier un dysfonctionnement de l'application (failure).

Idéalement, les défauts doivent être supprimés dans la même phase du cycle de projet que celle où ils ont été introduits. On parle alors de confinement de phase (phase containment). En effet, le coût pour détecter et supprimer un défaut augmente à chaque fois que celui-ci est postposé à une phase ultérieure. Depuis les requirements jusqu'à l'après release, cet accroissement de coût peut varier de 1 à 5, dans le cas de projets simples, de 1 à 100, pour des systèmes complexes. Un exemple est donné ci-dessous :



D'une manière générale, on distingue différents types de tests :

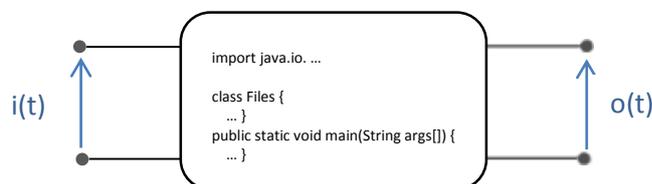
- Les tests « boîte noire » (ou « black-box » ou « specification-based » testing) : il s'agit des tests d'une application, au cours desquels on vérifie que les sorties obtenues sont bien celles prévues pour des entrées données (sans que les composants de l'application ne soient transparents pour l'utilisateur). L'application en question est alors connue uniquement via son interface d'entrées et de sorties.

Schématiquement parlant, cela peut être représenté comme suit : à une entrée $i(t)$, variant en fonction du temps, on associe une sortie $o(t)$. La forme du signal $o(t)$ est prévisible en fonction de celle du signal $i(t)$ ⁶.



- Les tests « boîte blanche » (ou « white-box » ou « structure-based » testing) : il s'agit des tests d'une application, dont on peut prévoir le fonctionnement interne car on connaît les caractéristiques des éléments qui la composent (par exemple, ses instructions).

Schématiquement parlant, cela peut être représenté comme suit : étant donné une entrée $i(t)$ et les instructions d'une application, il est possible de prévoir sa sortie $o(t)$.



- Les tests unitaires : il s'agit de tests au cours desquels on vérifie la fonction de chaque composant (ce sont des tests « boîte blanche » sur les instructions du code des composants).
- Les tests d'intégration : il s'agit de tests au cours desquels on vérifie les interfaces entre composants (il s'agit de tests « boîte blanche » sur les interfaces entre composants et « boîte noire » sur les instructions de chaque composant).
- Les tests système ou « end-to-end » : il s'agit de tests au cours desquels on vérifie un système intégré de différents composants, ainsi que s'il satisfait aux exigences spécifiées (de type fonctionnelles ou non). Il s'agit souvent de tests de type « boîte noire » (sur l'ensemble de l'application).
- Les tests d'acceptation : il s'agit de tests formels respectant les besoins utilisateurs, exigences et processus business, afin de déterminer si un système satisfait aux critères d'acceptation et permettant à l'utilisateur et aux clients de déterminer s'il(s) accepte(nt) le système ou non.
- Les tests fonctionnels : il s'agit de tests basés sur l'analyse de la spécification d'une fonctionnalité d'un composant ou système (afin de vérifier si celle-ci est respectée). Il s'agit de tests « boîte noire ».
- Les tests non fonctionnels : il s'agit de tests afin de vérifier les attributs d'un composant ou système non relatifs aux fonctionnalités (par exemple : fiabilité, maintenabilité, sécurité...).

⁶ Etant donné les spécifications du système.

- Les tests de performance : il s'agit d'un cas particulier de tests non fonctionnels, afin de vérifier l'efficacité (efficiency) d'un logiciel, à savoir sa capacité à fournir les performances appropriées, en termes de ressources utilisées dans des conditions données.
- Les tests de maintenance : il s'agit des tests d'une application, suite à une opération de maintenance (notamment : modification ou configuration), afin de s'assurer que cette dernière répond aux exigences de la demande de changement ou configuration.
- Les tests de régression : il s'agit de tests d'une application préalablement testée, suite à une modification, afin de s'assurer que des défauts n'ont pas été introduits, au cours de l'opération de maintenance ou dans des parties non modifiées. Notamment, on peut chercher à évaluer la rétro-compatibilité de l'application, à savoir sa capacité de gérer sans erreur des flux de données issus d'une version antérieure de l'application.
- Les tests techniques : il s'agit de tests ne cherchant pas à vérifier le comportement fonctionnel ou non fonctionnel de l'application (par exemple, des tests de connexion).

4. Description des services existant

4.1. Introduction

Afin de repartir de la situation « AS IS » des services existants, devant être testés de manière régulière ou occasionnelle, nous avons recensé la liste des flux, chacun étant géré par un chef de projet BCSS. Actuellement, un grand nombre de services et applications existent à la BCSS, ce en différentes technologies concurrentes. Notamment, concernant les flux de données « push » (notifications⁷) et « pull » (consultation), qui constituent le core business des applications BCSS, on peut recenser :

- 233 services de type A1 (notifications + consultation)
- 138 services de SSDN (consultation)
- 54 webservices SOA (consultation).

A cette liste, il faut ajouter un certain nombre de flux de données LDM (notifications), ainsi que plusieurs applications spécifiques. A la section suivante, nous détaillons davantage les services recensés.

4.2. Liste des services

Devant l'impossibilité de lister, de manière exhaustive, l'ensemble des services disponibles, nous avons choisi de nous limiter, en concertation avec les chefs de projets BCSS, à un nombre restreint mais représentatif de services (en terme de fréquence d'utilisation et de mise à jour, importance stratégique...). Parmi ceux-ci :

- Les flux cotisations sociales, dont l'ONSS et l'INASTI constituent la source authentique (DmfA, Dimona, Limosa, travailleurs indépendants)
- Les flux pension et carrière, dont le SFP et SIGeDIS constituent la source authentique
- Les flux registres, qui se composent des registres RN et BCSS

⁷ Anciennement appelées mutations.

- Les flux chômage, dont l'ONEM et les OP constituent la source authentique
- Les flux assurabilité / invalidité, dont le CIN et les OA constituent la source authentique
- Les flux fiscalité, dont le SPF Finances constitue la source authentique
- Les flux allocations familiales, dont Famifed constitue la source authentique
- Les flux vacances annuelles, dont l'ONVA constitue la source authentique
- Les flux régionalisés, dont les Régions constituent la source authentique
- Les flux et applications spécifiques.

Parmi les flux importants, ceux du CPAS et relatifs au handicap n'ont pu encore être ajoutés, faute de temps, mais suivent globalement les mêmes pratiques que les flux précédemment cités⁸.

Le tableau qui suit reprend les colonnes suivantes :

- Application ou flux de données concerné
- Source(s) authentique(s) de cette application ou ce flux de données
- Environnements disponibles
- Fréquence et origine des modifications
- Type de tests et origine des données.

⁸ Dans le cas des CPAS, les données sont généralement fournies par les firmes de soft, qui jouent ici le rôle d'intermédiaire avec les sources authentiques.

Application ou flux de données	Source(s) authent.	Environ. disponibles	Modifications		Tests	
			Fréquence	Origine	Type	Origine des données
DmfA	ONSS	ACPT, PROD	<ol style="list-style-type: none"> Trimestrielle Ajout d'un nouveau client 	<ol style="list-style-type: none"> Source authentique Destinataire 	<ol style="list-style-type: none"> Maintenance + régression Cas de tests représentatifs 	<ol style="list-style-type: none"> Copie de la production, données fictives Copie de la production
Dimona	ONSS	TEST, ACPT, PROD	<ol style="list-style-type: none"> Occasionnelle Ajout d'un nouveau client 	<ol style="list-style-type: none"> Source authentique, BCSS Destinataire 	<ol style="list-style-type: none"> Maintenance (pas de tests de régression) Pas tests systématiques 	<p>Test : données fictives ou copie à la demande de la production</p> <p>Acpt : copie toutes les 2 semaines de la production</p>
Limosa	ONSS	TEST, ACPT, PROD	<ol style="list-style-type: none"> Adaptations rares Nouveaux abonnements occasionnels 	<ol style="list-style-type: none"> Source authentique Destinataire 	<ol style="list-style-type: none"> Maintenance Cas de tests représentatifs 	Copie de la production
Travailleurs indépendants	INASTI	TEST, ACPT, PROD	<ol style="list-style-type: none"> Occasionnelle Ajout d'un nouveau client 	<ol style="list-style-type: none"> Source authentique, BCSS Destinataire 	Maintenance uniquement	Copie de la production ou données fictives
Flux pension et carrière	SFP, SIGeDIS	TEST, ACPT, PROD	<ol style="list-style-type: none"> Occasionnelle Ajout d'un nouveau client 	<ol style="list-style-type: none"> Source authentique, BCSS Destinataire 	Maintenance uniquement	Copie de la production ou données fictives

Application ou flux de données	Source(s) authent.	Environ. disponibles	Modifications		Tests	
			Fréquence	Origine	Type	Origine des données
Services RN	RN	ACPT, PROD	A préciser	Source authentique, BCSS	Maintenance uniquement	Copie de la production ou données fictives
Services registres BCSS	BCSS	TEST, ACPT, PROD	A préciser	Source authentique, BCSS	Maintenance uniquement	Copie de la production ou données fictives
Flux chômage	ONEM	ACPT, PROD	Occasionnelle	Source authentique, BCSS	Maintenance uniquement	Copie de la production à la demande
Flux assurabilité / invalidité	CIN	ACPT (ou « pilote »), PROD	Occasionnelle	Source authentique	Pas tests systématiques	Copie de la production (demande aide source authentique pour identifier cas)
Taxi-AS	SPF Finances	ACPT, PROD	1. Occasionnelle 2. Ajout d'un nouveau client	1. Source authentique, destinataire 2. Destinataire	Pas tests systématiques	Copie de la production (demande aide source authentique pour identifier cas)
PatrimonyService	SPF Finances	ACPT, PROD	1. Occasionnelle 2. Ajout d'un nouveau client	1. Source authentique, BCSS 2. Destinataire	Pas de distinction entre tests de maintenance et de régression	Copie de la production (demande aide source authentique pour identifier cas)
Flux allocations familiales	Famifed	TEST, ACPT, PROD	A définir	A définir	A définir	Copie de la production

Application ou flux de données	Source(s) authent.	Environ. disponibles	Modifications		Tests	
			Fréquence	Origine	Type	Origine des données
Flux vacances annuelles	ONVA	ACPT, PROD	1. Occasionnelle 2. Ajout d'un nouveau client	1. Source authentique, BCSS 2. Destinataire	Pas de distinction entre tests de maintenance et de régression	A définir
Flux régionalisés Flandre	VDI	TEST, ACPT, PROD	1. Occasionnelle 2. Ajout d'un nouveau client	1. Source authentique 2. Destinataire	Maintenance uniquement	Copie de la production à la demande
eDeductions	SPF Finances, Autres	ACPT, PROD	Pas nécessaire a priori	A définir	A définir si nécessaire	Copie de la production
Zacheus / IncomeInformatio ns	ONSS, ONVA, Autres	TEST, ACPT, PROD	Selon sources consultées (Dimona, A038...)	Source authentique	Pas de distinction entre tests de maintenance et de régression	Copie de la production, données fictives
Caliope	ONSS, Autres	TEST, ACPT, PROD	Selon sources consultées (DmfA, Dimona...)	Source authentique	Maintenance + régression	Copie de la production, données fictives
Dolsis	ONSS, CPAS, Autres	ACPT, PROD (TEST : tests connexion, pas accessible aux partenaires)	1. Semestrielle 2. Ajout d'un nouveau client	1. Source authentique, Smals 2. Destinataire	1. Maintenance + régression 2. Cas de tests représentatifs	Copie de la production ou données fictives : <ul style="list-style-type: none"> • DmfA : 20.000 cas (année 2012) • Dimona : extracts réguliers vers ACPT • CPAS : copies à la demande

Application ou flux de données	Source(s) authent.	Environ. disponibles	Modifications		Tests	
			Fréquence	Origine	Type	Origine des données
Projet européen BEESSI	Institutions nationales et européennes de sécurité sociale	DEV, TEST, ACPT, PROD	A définir	A définir	A priori : régression, cas de testreprésentatifs	A définir

4.3. Observation des pratiques actuelles en termes de test

Les observations qui peuvent être réalisées, par rapport aux services et pratiques actuels, peuvent être relatives à différents domaines :

- Environnements de test
- Méthodes qualité
- Origine des données
- Communication avec les partenaires.

Par la suite, nous détaillons chacun des points ci-dessus.

4.3.1. Environnements de test

D'une manière générale, les environnements suivants sont disponibles pour la plupart des projets BCSS : TEST, ACPT et PROD. A noter que l'environnement TEST n'est pas systématiquement utilisé par les partenaires extérieurs (ce qui est recommandé vu l'instabilité des applications qui s'y trouvent), sauf lorsque le partenaire souhaite réaliser des tests immédiatement, sans attendre la mise en ACPT (généralement planifiée à fréquence hebdomadaire). De plus, l'environnement TEST n'est parfois utilisé que pour des tests techniques.

4.3.2. Méthodes qualité

Suite au développement, il est possible de distinguer différents types d'adaptations des services existants :

- Les configurations (comme l'ajout d'un partenaire ou d'une donnée). La mise en production d'un nouveau service peut être considérée comme une configuration en vue d'ajouter l'ensemble des partenaires accédant au service.
- Les demandes de changement (qui visent à modifier les exigences fonctionnelles ou non fonctionnelles de l'application)
- Les corrections.

Pour chacune de ces adaptations, il convient de distinguer la fréquence, le partenaire à l'origine de celles-ci, ainsi que les méthodes de test mises en œuvre :

- En ce qui concerne les demandes de configuration, celles-ci ont généralement lieu à l'initiative d'un partenaire, qui dispose d'une délibération afin d'accéder à un service, aux données, etc. Le cas échéant, il sera intéressant de fournir au partenaire des scénarios de test fonctionnels, permettant de vérifier le fonctionnement du service ou l'accessibilité des données.
- Pour les demandes de changement, celles-ci peuvent avoir lieu à fréquence périodique (par exemple, adaptations trimestrielles) ou à la demande des partenaires (sources authentiques, BCSS, utilisateurs). Dans ce dernier cas, la fréquence des adaptations est impossible à prédire. Des scénarios de tests spécifiques au changement, ainsi qu'idéalement de régression, peuvent être fournis aux partenaires.

- Les corrections doivent être réalisées le plus rapidement possible et peuvent impliquer des tests spécifiques.

A noter que s'il existe un plan de test, comme prévu lors de la phase 3 de la procédure [5], ce dernier n'est (presque) jamais communiqué aux partenaires.

Hormis les tests techniques et d'intégration, les tests les plus souvent réalisés sont les tests système impliquant les partenaires externes. Parmi ces derniers, les tests fonctionnels ou de maintenance sont les plus souvent réalisés. De même, les cas de test réalisés sont en général de type « valides », les cas « invalides » (NISS pas intégré, RN ou service backend indisponibles, absence de certificat...) n'étant pas systématiquement testés par les partenaires.

Les tests de régression, qui permettent de vérifier la rétro-compatibilité d'une application, ne sont que rarement exécutés. Toutefois, pour certains flux comme la DmfA, la source authentique gère une banque de cas de test de régression, enregistrés depuis le début de l'application, pouvant être réutilisés lors d'une adaptation.

Les tests de performance ne sont pas systématiquement réalisés, étant donné qu'ils ne se justifient pas toujours (par exemple, lorsque le volume de données, mis à disposition des partenaires en production, ne dépasse pas un certain seuil). Cependant, lorsqu'ils le sont, c'est souvent en dernière minute par les partenaires (juste avant la release en PROD). De plus, ces tests nécessitent de disposer d'un grand nombre de NISS intégrés en TEST ou ACPT pour la source ou le destinataire (voir 4.3.3). La question des volumes consultés en test, afin d'être suffisamment représentatif⁹, est également un point d'attention, d'autant plus que le comportement peut parfois varier entre les environnements ACPT et PROD.

4.3.3. Origine des données

Comme signalé, des scénarios de tests ne sont pas systématiquement mis à disposition des partenaires, étant donné l'absence de réactivité de certaines sources authentiques. Quand disponibles, les données de tests sont soit fictives, soit copiées depuis la base de données en PROD :

- En cas de données de PROD, la copie a lieu à fréquence périodique ou à la demande. La fiabilité des données n'est alors pas toujours garantie. Notamment, en cas de copie à fréquence régulière (par exemple, tous les 15 jours), les scénarios de test sont difficilement pérennes (car de nouvelles données peuvent venir « écraser » les anciennes). De plus, certains retards ont été constatés, afin de dupliquer les données vers l'ACPT (en cas de copie quelques fois par an). Lorsqu'il s'agit de données de PROD, une autorisation devrait être disponible pour les partenaires qui testent (voir [1]).
- En cas de données fictives, les NISS utilisés dans les scénarios sont susceptibles d'être contrôlés au RN (généralement en environnement PROD). Cela implique donc une stratégie des partenaires dans le choix des NISS (qui doivent exister, être intégrés, etc.).

D'une manière générale, les profils RN étant susceptibles d'évoluer (par exemple, un NISS peut être remplacé, un BIS devenir un numéro RN, etc.), il faudra en tenir compte afin de mettre en place un environnement de test stabilisé. Dans le cas des profils BIS, ceux-ci peuvent être modifiés par les partenaires en ACPT, d'où une difficulté à garantir leur intégrité. Étant donné son rôle, un contrôle d'intégration (CI) est souvent réalisé par la BCSS, avant de communiquer les données au partenaire. Par conséquent, l'absence d'intégration d'un nombre de cas suffisant en

⁹ Étant donné certains process parallèles.

ACPT peut poser problème, étant donné la séparation des environnements ACPT et PROD (impossibilité d'effectuer de CI en PROD lors des tests). Ce problème se pose d'autant plus lors des tests de performance, où un grand nombre de dossiers (idéalement différents, afin d'être représentatifs) est consulté simultanément.

4.3.4. Communication avec les partenaires

Actuellement, les scénarios de test sont transmis par email dans la plupart des projets. Il n'y a donc pas d'espace centralisé, accessible à tous les partenaires, permettant de communiquer ces scénarios, leurs mises à jour, etc.

5. Pistes d'améliorations

5.1. Introduction

Suite à l'analyse « AS IS » des flux et données disponibles, réalisée à la section précédente, et les observations générales qui en découlent, nous décrivons à présent les pistes d'amélioration, qu'il est possible de dégager, en vue d'améliorer la qualité des tests réalisés et, de là, des services mis à la disposition des partenaires extérieurs.

A ce stade, les pistes d'améliorations suivantes peuvent être proposées :

- Environnements de test
- Méthodes qualité
- Origine des données
- Communication avec les partenaires.

Nous détaillons ci-après chacune de ces pistes.

5.2. Environnements de test

De manière générale, nous pouvons considérer que les 2 ou 3 environnements mis à disposition des partenaires externes (ACPT et PROD, parfois TEST) sont actuellement suffisants pour la plupart des demandes gérées par la BCSS.

Toutefois, dans le cas de projets spécifiques, comme le projet européen BeEESSI, la mise à disposition d'un environnement TEST (ou « pré-acceptation ») distinct de l'environnement DEV, afin de réaliser des tests d'intégration avec les partenaires, pourra s'avérer indispensable. L'environnement DEV servira aux développeurs internes, afin de réaliser des tests unitaires, tandis qu'ACPT sera réservé à la validation par les partenaires et aux tests système.

5.3. Méthodes qualité

Une amélioration de la procédure [5] peut prévoir de communiquer le plan de test aux partenaires. Ce plan de test devra reprendre :

- Les objectifs des tests
- Leurs types (fonctionnels, performance...)
- Eventuellement une référence aux use cases du PID

- Les scenarios de test avec prérequis et conditions de réussite.

Suite aux tests en ACPT, le plan de test pourra être mis à jour, ce qui constituera une validation formelle des tests réalisés par les partenaires, nécessaire à la mise en PROD. Suite à cette dernière, le chef de projet assume la vérification des traitements, lors des premières exécutions en PROD, puis donne son feu vert pour le transfert vers le service de gestion.

En plus des cas de test « valides », des cas « invalides » (NISS pas intégré, RN ou service backend indisponibles, absence de certificat...) pourront être proposés aux partenaires dans les plans de test. Le cas échéant, une réflexion devra être menée quant à la meilleure manière de simuler ces cas (par exemple, des mock tests avec messages d'erreur seront communiqués aux partenaires qui pourront ainsi vérifier leur application).

En ce qui concerne les tests de régression, il convient d'analyser dans quelle mesure ceux-ci peuvent être généralisés à un certain nombre de flux importants (Dimona, RGTI, flux pension et carrière...). Le cas échéant, il faudra disposer d'un environnement de test suffisamment stable, ainsi que d'une banque de cas de tests représentatifs.

Au niveau tests de performance, l'accent doit être mis auprès des partenaires, afin qu'ils réalisent ces tests suffisamment à l'avance avant la mise en PROD, sans quoi il est impossible à la BCSS de garantir que les exigences seront respectées.

5.4. Origine des données

En concertation avec la source authentique, des cas de tests devront être mis à disposition plus systématiquement. L'accent sera particulièrement mis sur les partenaires « mauvais élèves », qui ne communiquent pas toujours de scenarios de test représentatifs :

- En cas de données de PROD, l'accent sera mis sur la fiabilité et la stabilité de ces données. De même, ces cas devront être recopiés à temps par la source authentique.
- En cas de données fictives, il faut vérifier si les NISS utilisés existent bien, car certains services vérifient leur existence sur base d'une consultation RN.

De même, il pourra être intéressant de définir des profils « de référence », en concertation avec le RN et RBIS, qui seront utilisés lors des tests. En ce qui concerne le CI, les partenaires qui testent doivent être sensibilisés à intégrer régulièrement leurs cas en environnement PROD, mais également TEST et ACPT.

5.5. Communication avec les partenaires

Afin d'augmenter l'efficacité des échanges, notamment concernant la documentation technique (TSS, XSD) et les scenarios de test, la mise à disposition d'un espace centralisé, de type SharePoint Online, pour l'ensemble des partenaires internes et externes, sera envisagé pour toutes les demandes BCSS. Cela aura lieu via le nouvel espace « beConnected » de la BCSS (suite à migration vers SharePoint).

6. Conclusion

Comme exprimé dans ce document, un ensemble de bonnes pratiques relatives au testing est déjà appliqué aux différents services et applications disponibles au sein de la BCSS. Toutefois, certains services, notamment suivant la source authentique, souffrent d'un manque

d'uniformisation en termes de qualité, en comparaison avec les plus gros services disponibles (par exemple, DmfA ou Dolsis).

De même, les méthodes qualité en matière de test peuvent sans doute être améliorées, impliquant davantage le chef de projet BCSS, notamment afin de valider les tests d'acceptation en tant qu'intermédiaire du client (et mettre à jour le plan de test avec les résultats de test).

Il convient toutefois de signaler qu'une implication plus grande des partenaires et sources authentiques est demandée, afin d'assurer la fiabilité des cas de test, l'intégration des dossiers en ACPT, la réalisation des tests de performance...

A terme, il convient de mettre en place des mesures correctives, en vue de l'amélioration continue des process, afin de garantir à tous les clients qui consultent nos services un même niveau de qualité et de satisfaction.