
Mission critical solutions
services
security
safety

Retour sur expérience du projet DH2

XP Days 02/05/2007

Pierre Bufferne (SWCS)
Marc-André Filion-Pilon (Thales Belgique)

Etape 1:
Contexte du projet DH2

Contexte

- Client : Société Wallonne du Crédit Social (SWCS)
  - 350 M€ d'activité, prêts hypothécaires revenus précaires
  - 65 personnes – 6 personnes au service informatique
  - Tutelle sur 23 guichets locaux employant 95 personnes
- DH2 : Outil de gestion de prêts hypothécaires
  - Nouveaux modules et remplacement d'applications existantes
- Démarrage du projet, octobre 2005 -1ère mise en ligne, mai 2006

Contexte

- **Volonté du client d'utiliser XP, et tout XP**
  - souci d'autonomie de maintenance : binômes
  - faible maîtrise métier des équipes de production : développement itératif
  - volonté de créer de l'adhésion : réactivité par des livraisons rapides – écoute des demandes des utilisateurs
  - solidité face aux changements législatifs sur 30 ans : tests
- Besoin d'ergonomie au sein du projet car utilisateurs sur plusieurs sites de production

Pilotage du projet

- Projet piloté par le périmètre (dont la qualité), souplesse coûts et délais

Choix technologique : .NET, C#

- Un seul outil (Visual Studio, TFS), avec intégration forte, correspondant mieux à la petite taille de l'équipe de maintenance SWCS
- Client léger pour les 23 guichets locaux + ouverture au grand public

Équipe choisie

L'équipe de développement est choisie sur base d'un marché public.

C'est la société THALES (Belgique) qui remporte le marché en septembre 2005.

L'organisation mise en place :

- 1 chef de projet MOA SWCS à temps partiel, 1 chef de projet MOE
- 1 assistant à maîtrise d'ouvrage (tiers indépendant sélectionné par marché public)
- 1-2 analystes THALES
- 6 développeurs THALES
- 3 développeurs SWCS à temps partiel, dont 1 testeur + mise à jour bdd

Etape 2 :
Solution mise en place

Quelques métriques

Analyse	
Nb Modules	6+
Nb CUs	200+
Nb SCs	500+

Base de données	
Nb Tables	53
Nb Vues	10
Nb Fonctions	20
Nb SPs	147

Code	
Nb Projets	24
Nb Fichiers	540
Nb Classes	720
Nb Lignes	170.000

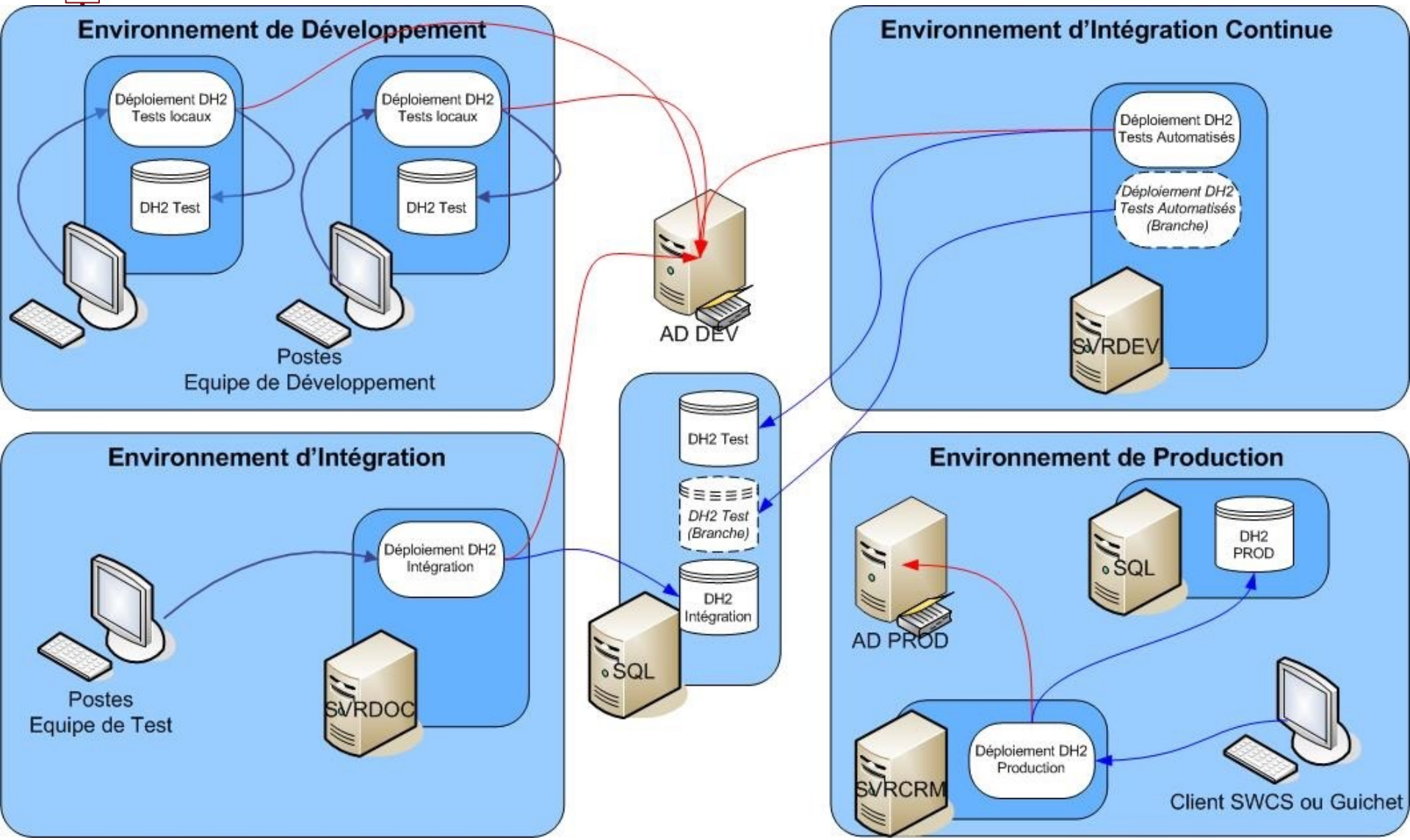
Tests	
Nb Tests Recette	1.300
Nb Test Unitaires	400
Nb Total	1.700
% Code Test	~ 50%
Temps d'exécution	2 h 45

Description technique de l'application

- Application Web en ASP.NET (C#)
  - VisualStudio .NET 2005
  - Utilisation du Framework .NET 2.0 (Beta) avant sa sortie officielle
  - Migration au Framework 3.5 prévue prochainement

- Bases de données : SQLServer 2005 et Progress (système existant)
- Utilisation d'ActiveDirectory pour l'authentification et les rôles
- Team Foundation Server utilisé pour les tests et l'Intégration Continue
- JIRA (Atlassian) comme outil de suivi des tâches et des bugs
- Confluence (Atlassian) comme Wiki de partage d'information, documentation des Cas d'Utilisation, de l'architecture, et du contenu des livraisons

Solution mise en place – 3 / 3



Etape 3 :
Utilisation de XP dans DH2

Utilisation de tout XP ?

Parcours des 13 pratiques dans le projet

1. Livraisons courtes et rapides
   - Itération de 2 semaines
   - 2 itérations par livraison en production = 4 semaines
   - Livraisons : au Début 6 sem, Puis 2 sem et Actuellement 4 semaines
2. Intégration Continue (+)
   - Build et tests lancés chaque nuit
   - Avantages de la solution intégrée TFS Server de Microsoft
3. Développement piloté par les tests (TDD) (+)
   - Surtout du développement piloté par les Tests de Recette
   - Création d'un framework de tests de recette

Utilisation de XP – 2 / 3

4. Implication du client (+)
   - L'équipe THALES travaille sur site depuis le début du projet
   - Le client est impliqué dans l'analyse, la planification, le développement et la validation
   - Organisation du retour d'infos des utilisateurs internes et externes

5. Travail en binôme
   - Trop tôt pour déterminer si rentable
   - Tentation de scinder les binômes pour aller 'plus vite'

6. Rythme soutenable
   - A nécessité l'adaptation du rythme des livraisons

7. Jeu du planning (-)
   - Pas pris tel quel à ce jour
   - Travail en cours pour utiliser la vélocité

Utilisation de XP – 3 / 3

8. Refactoring en continu
9. Appropriation collective du code
10. Conception simple (mais non simpliste)
11. Utilisation de métaphores
12. Convention de nommage
    - Utilisation du “franglais”
13. Stand-up meetings

Etape 4 :
Développement piloté par les tests (TDD)

Besoins du client

- Permettre de tester l'application à partir de l'interface web
  - Test de la couche de présentation, pas seulement de la couche « Métier »
- Permettre de simuler les actions des utilisateurs
- Permettre de coder les tests avant de coder l'implémentation
  - Pas un robot
  - Ne doit pas être impacté par un déplacement des contrôles de l'interface
- Le code de test de recette doit être lisible par le client

Solution

- Nécessite la création de nouveaux outils pour automatiser les TRs

1. Création d'un Simulateur web pour simuler les actions des utilisateurs de l'application via des requêtes HTTP

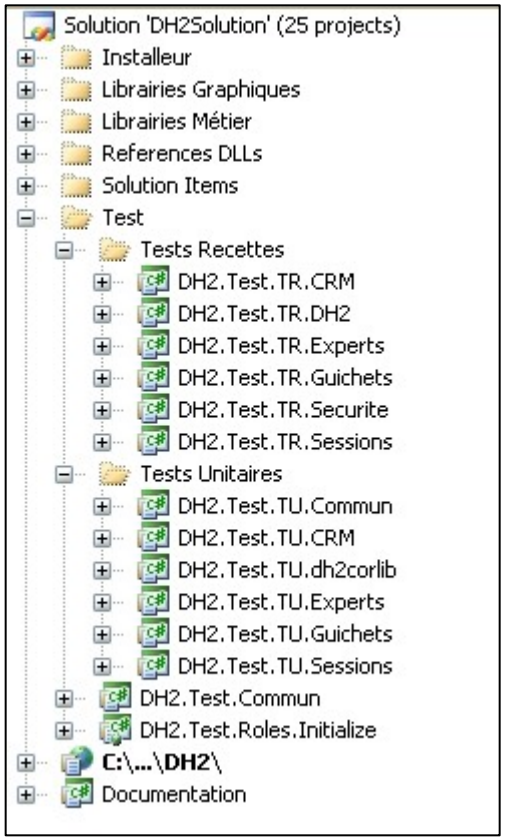
2. Création d'un Framework de tests de recette pour
   - Ajouter des conditions initiales uniques à un test donné
   - exposer une liste de commandes de haut niveau pour effectuer les actions et la validation du site

Métrique des tests dans DH2

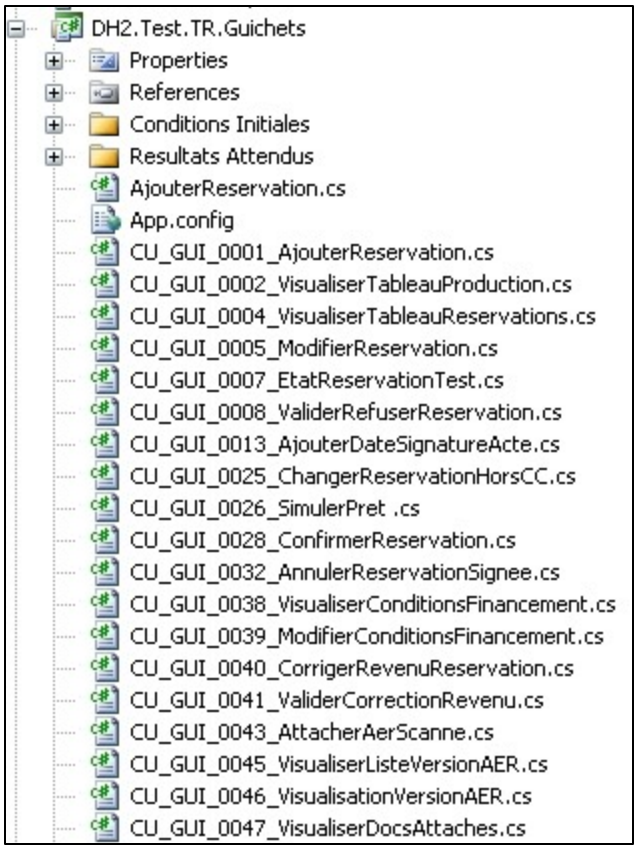
Tests	
Nb Tests Recette	1.300
Nb Test Unitaires	400
Nb Total	1.700
% Code Test	~ 50%
Temps d'exécution	2 h 45

Arborescence des TRs dans la solution DH2

Projets de tests



Classes de tests



Composition des fichiers de CUs dans la solution DH2

```
[TestClass]
[Interception]
public class CU_GUI_0068_ValiderReservationApresSignatureActe : ContextBoundObject
{
    private static readonly string urlVisualiserReservation = "~/Controles/VisualiserR

    SC_GUI_0068_01: Validation d'une réservation après signature de l'acte

    SC_GUI_0068_02: Refus d'une réservation après signature de l'acte

    SC_GUI_0068_03: Validation impossible d'une réservation après signature de l'acte

    SC_GUI_0068_04:
}
```

```
[TestClass]
[Interception]
public class CU_GUI_0068_ValiderReservationApresSignatureActe : ContextBoundObject
{
    private static readonly string urlVisualiserReservation = "~/Controles/VisualiserR

    #region SC_GUI_0068_01: Validation d'une réservation après signature de l'acte

    SC_GUI_0068_01_TR1: Validation d'une réservation après signature de l'acte

    SC_GUI_0068_01_TR2: Sauvegarde d'une ligne dans l'historique de la réservation lor

    SC_GUI_0068_01_TR3: Validation d'une réservation après signature de l'acte sans at

    #endregion
```

Exemple de test de recette codé dans DH2

```
/// <summary>
/// Acteur : dh2_dcg_user
/// Date d'exécution: 16/03/2006
/// Instructions:
/// - Aller sur la page du tableau d'affichage des réservations
/// - Chercher les réservations au statut "Acte Signé"
/// - Cliquer sur l'hyperlien correspondant à la réservation DR-1003-00001
/// - Cliquer sur le bouton "Valider la réservation"
/// - Vérifier que la page de visualisation de la réservation est affichée
/// - Vérifier que le statut de la réservation est: "Attestation Conforme"
/// </summary>
[TestMethod]
[SpecifieGuichetEnRegle(true, true)]
[ExecuterADateDe(16, 3, 2006)]
[NecessiteEffacementReservationEtSession]
[ExecuterEnTantQue(UtilisateurTest.Dh2_DCG_User)]
[InsereReservation(TypeReservationTest.GUI1003_Avec_Attestation, ValideActeSigne = true,
   Statut = StatutReservation.ActeSigne,
   DateConfirmation = "15/03/2006")]
public void SC_GUI_0068_01_TR1()
{
   SimulateurWeb sw = new SimulateurWeb();
   sw.VaSurReservationEnTantQueDRFouDCG("DR-1003-00001", "Acte Signé");
   sw.CliqueSurBouton("Valider la réservation");
   sw.ValidePage(urlVisualiserReservation);
   sw.ValideLabel("txtStatut", "Attestation Conforme");
}
```

1. Description du test

2. Conditions Initiales

3. Corps du test

1. Description du test

- L'analyste décrit les étapes du test
- Utilisation du commentaire XML (« summary ») en haut de chaque méthode de test
  - Possibilité d'extraire le commentaire XML pour avoir la description de tous les tests d'un module donné

```
/// <summary>
/// Acteur : dh2_dcg_user
/// Date d'exécution: 16/03/2006
/// Instructions:
/// - Aller sur la page du tableau d'affichage des réservations
/// - Chercher les réservations au statut "Acte Signé"
/// - Cliquer sur l'hyperlien correspondant à la réservation DR-1003-00001
/// - Cliquer sur le bouton "Valider la réservation"
/// - Vérifier que la page de visualisation de la réservation est affichée
/// - Vérifier que le statut de la réservation est: "Attestation Conforme"
/// </summary>
```

2. Conditions initiales

- Permet d'initialiser le système dans les conditions uniques pré-requises au test
- Ajoutées par l'analyste lors de la rédaction du test
- Chaque test :
  - a ses propres conditions initiales
  - ne dépend pas des conditions des autres tests
- Permet d'isoler le code du test
  - le corps de la méthode = le test et rien d'autre

```
[TestMethod]
[SpecifieGuichetEnRegle(true, true)]
[ExecuterADateDe(16, 3, 2006)]
[NecessiteEffacementReservationEtSession]
[ExecuterEnTantQue(UtilisateurTest.Dh2_DCG_User)]
[InsereReservation(TypeReservationTest.GUI1003_Avec_Attestation, ValideActeSigne = true,
   Statut = StatutReservation.ActeSigne,
   DateConfirmation = "15/03/2006")]
```

Conditions initiales: Types

- Date d'exécution du test :
  - Ex: [ExecuteEnDateDe(2007,04,19)]
- Utilisateur connecté :
  - Ex: [ExecuteEnTantQue(Utilisateur.dh2_expert)]
- Insertion d'objets métiers en base de données
  - Objet métiers : Réservations, Dossiers de prêt, Rapport d'expertise
  - Ex: [InsereReservation(Type.Reservation)]
- Attachement d'un document à un objet métier
  - Ex: [InsereDocumentAttache(« MonFichier.pdf », « DR-1003-012541 »)
- Effacement de table en base de données
  - Ex: [NecessiteEffacementTable(« GUI_Reservation »]
- Modification de la configuration de l'application

Conditions initiales : Objets métier réutilisables

- Lors de l'insertion d'un objet métier, permet d'instancier un objet précis parmi une liste pré-définie d'objets «types »
  - Exemples :
    - TestReservationType.Categorie1
    - TestReservationType.Categorie1_Avec_DocumentAttache
    - TestDossierPretType.UnEmprunteur
- Possibilité de modifier un ou plusieurs champ de l'objet « type » lors de l'insertion
  - Utilisation de propriétés « nommées » de l'attribut
  - Permet de mettre en évidence le champ qui sera testé par le test
  - Exemple:
    - [InsereReservation(Type.Categorie1, MontantFinance=15000, DateConfirmation=« 12/04/2007 »)]

3. Corps du test

- Le développeur code les étapes du test
- Utilisation des commandes de haut niveau du simulateur Web

```
public void SC_GUI_0068_01_TR1()
{
    SimulateurWeb sw = new SimulateurWeb();
    sw.VaSurReservationEnTantQueDRFouDCG("DR-1003-00001", "Acte Signé");
    sw.CliqueSurBouton("Valider la réservation");
    sw.ValidePage(urlVisualiserReservation);
    sw.ValideLabel("txtStatut", "Attestation Conforme");
}
```

Commandes du simulateur web

- Pour naviguer :
  - VaSurPage(« Accueil.aspx »), VaDansTableau()
- Pour remplir un formulaire :
  - RemplitChamp(« txtMontant », 800)
  - CocheCase(« EstMajeur », true)
- Pour valider un formulaire :
  - ValideLabel(), ValideChamp(), ValideTableau()
- Pour effectuer des actions simples sur la page :
  - CliqueSurBouton(« Soumettre »), CliqueSurLien(« Suivant »)
- Pour effectuer des actions complexes (Macros)
  - VaSurReservation(« DR-1003-02456 »)
  - CorrigerRevenuReservation(« DR-1206-00541 », 15000)

Améliorations possibles

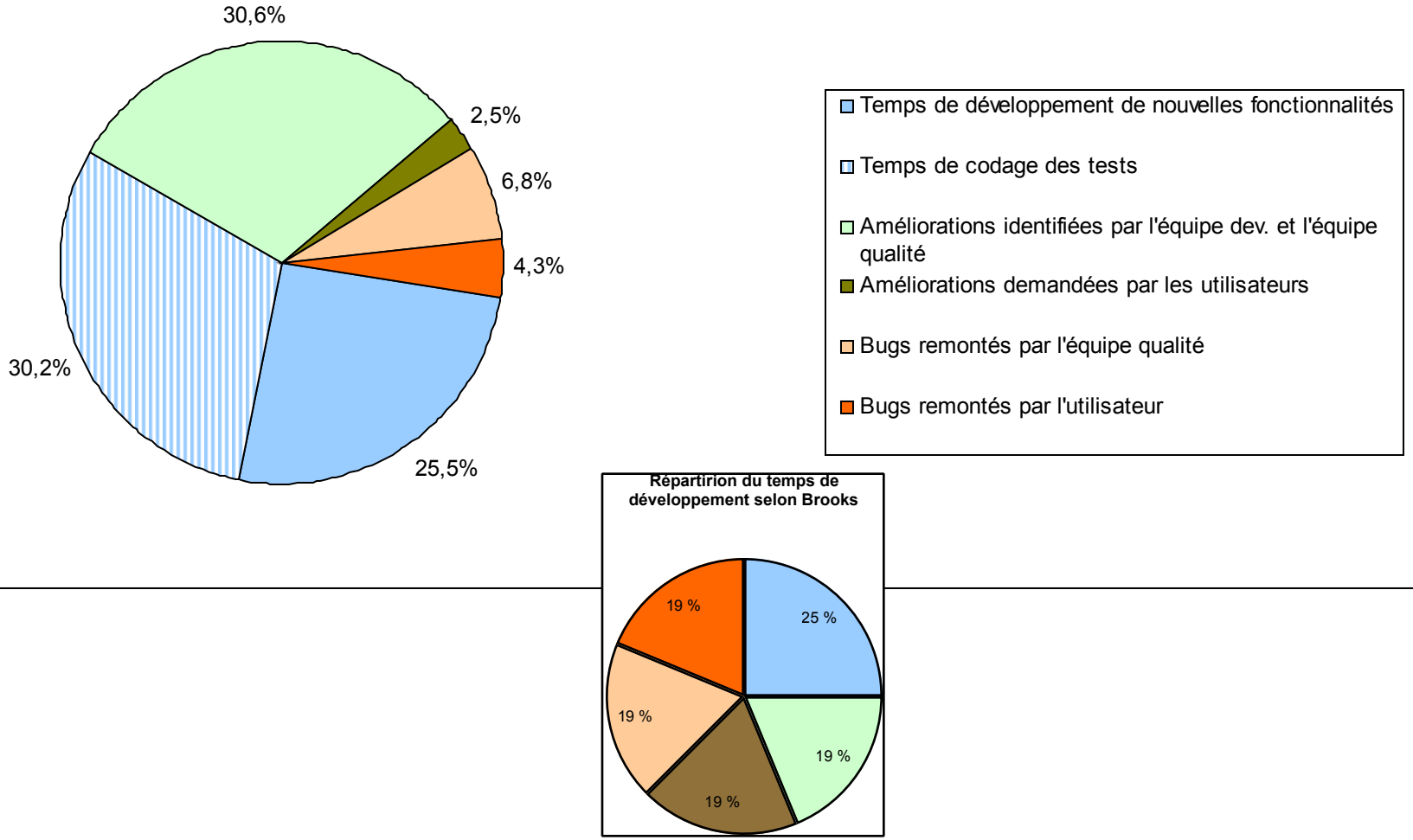
- Le framework de test supporte les fonctionnalités les plus importantes que nous avons rencontrées
  - Plus de 95% des fonctionnalités peuvent actuellement être testées automatiquement
- D'autres améliorations sont possibles :
  - Tests de clients web riches
    - Javascript
    - AJAX *(Asynchronous Javascript and XML)*
  - Vérification de l'envoi des mails et leur contenu
  - Vérification du contenu des exports Excel / PDF

Étape 5 : Bilan de l'expérience

Bilan de l'expérience – 1 / 3

- Le développement itératif et les livraisons rapides ont permis de créer de l'adhésion et une appropriation de l'outil par les utilisateurs
  - Contact renoué avec les guichets externes
  - Refonte des processus proposée par les utilisateurs
- La souplesse du système et sa capacité à permettre des changements rapides reste à évaluer ainsi que la rentabilité du travail en binômes.
- La bonne qualité perçue de l'application crée le contraste par rapport à d'autres expériences de développement vécues dans l'administration

Bilan de l'expérience – 2 / 3



Bilan de l'expérience – 3 / 3

- Les tests automatisés assurent une non-régression essentielle dans le métier
- Après un an et demi, nous ressentons le besoin de croiser les expériences car la méthodologie est encore peu représentée en Wallonie.

Étape 6: Des questions ?