

Rapport d'un BTS SIO

★ Stage : Evoliz
Semaine 6



BRUEL LUCAS

16/06/2023

Sommaire

I/ Le stage.....	2
A/ Définition du stage.....	2
B/ Présentation de l'entreprise.....	2
II/ Résumé de la semaine.....	3
A/ Ressentit et difficulté.....	3
B/ Les activités effectuées.....	3
III/ Sprint de la semaine.....	4
A/ L'objectif.....	4
B/ Synthèse des us.....	4
IV/ Vocabulaire.....	7
V/ Approfondissement : Les T.U en php.....	7
A/ Prérequis :.....	7
B/ Créer un test unitaire.....	8

I/ Le stage.

Dans cette partie, je souhaite rappeler toutes les informations concernant mon stage tel que le cadre qui définit mon stage et le but et une présentation de l'entreprise.

A/ Définition du stage.

Mon stage se déroule du 9 Mai au 2 juillet 2023, en tant que développeur dans une entreprise proposant une solution de facturation facile accompagnée par ces partenaires, nommée Evoliz.

Durant ce stage, avec l'aide de mon tuteur de stage Monsieur Julien Libert, je travaillerai dans le développement, la maintenance, l'amélioration, l'ajout de fonctionnalités, ainsi que la modification de l'existant de la solution de facturation et pré comptabilité Evoliz.



B/ Présentation de l'entreprise.

Crée en 2011, cette SAS (Société par Actions Simplifiée) réussit déjà à s'affirmer en tant qu'acteur innovant et dynamique dans le paysage des éditeurs en proposant des solutions facilitant le principe de facturation auprès d'auto-entreprise et désormais de grosses entreprises. Cette entreprise ne comptait, au départ, que seulement 2 employés, leurs fondateurs Olivier et François¹, afin d'arriver à 32 employés avec un recrutement actif de nos jours.

¹ [Site d'Evoliz](#)

II/ Résumé de la semaine.

A/ Ressenti et difficulté.

Durant cette semaine, je vis cette même routine que l'entreprise a mise en place, une routine agréable et qui respecte les employés !

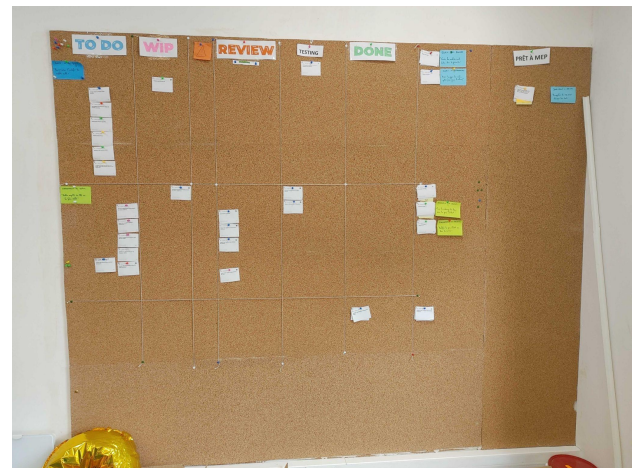
De mon côté, comme chaque semaine, j'ai pu participer au sein de l'équipe de développeurs afin de maintenir et d'améliorer la solution Evoliz.

Côté difficulté, ce qui nous auras pris le plus de temps fut l'installation de la pop-in dans la liste des factures impayée, nous pensions qu'au départ, il faudrait seulement mettre l'affichage de la pop-in, mais il s'avère qu'il y avait toute l'interaction que nous avons à refactoriser et à exporter afin d'y parvenir, cependant cela a provoqué un retard considérable avec le sprint de cette semaine mais avec un code réutilisable.

B/ Les activités effectuées.

Au niveau des activités, celles-ci restent en lien avec la routine ayant lieu chaque semaine, celle du SPRINT qui reste axé à l'objectif principal qui est de générer une relance depuis la liste des factures impayées. Je traiterai cela dans la partie suivante.

Tout comme chaque semaine, à partir de la méthode OKR, un **SPRINT** à lieux et se compose en 5 cérémonies : **Planification, Daily Scrum Meeting, L'affinage, Review et Rétrospective.**



III/ Sprint de la semaine.

A/ L'objectif.

Goku vs Lapins

Générer une relance depuis la liste des factures impayées

Search this board

Only My Issues Recently Updated

Goku vs Lapins Clear all

Insights

TO DO	IN PROGRESS	REVIEW	TESTING	DONE
<p>SUPPRESSION D UN MODELE UTILISE</p> <p>BUG-2831</p> <p>Gestion des succès de création (des 3 cas)</p> <p>RELANCES V2</p> <p>SCRUM-10621</p> <p>Gestion des erreurs sur création (des 3 cas)</p> <p>RELANCES V2</p> <p>SCRUM-10622</p> <p>Gérer la signature dans la relance</p> <p>RELANCES V2</p> <p>SCRUM-10655</p> <p>Désactiver l'envoi par courrier</p> <p>RELANCES V2</p> <p>SCRUM-10380</p> <p>Gestion des champs destinataires (affichage)</p> <p>RELANCES V2</p> <p>SCRUM-10656</p> <p>Pré- peupler les champs destinataires</p>	<p>Refacto paramètres</p> <p>RELANCES V2</p> <p>SCRUM-10614</p> <p>Affichage popin de création sur utilisation du CTA</p> <p>RELANCES V2</p> <p>SCRUM-10620</p>			<p>Générer la relance et préparer l'envoi par courrier</p> <p>RELANCES V2</p> <p>SCRUM-10352</p> <p>Retour de review</p> <p>RELANCES V2</p> <p>SCRUM-10648</p> <p>Affichage du CTA de création (enveloppe)</p> <p>RELANCES V2</p> <p>SCRUM-10377</p>

Afin de commencer le sprint de cette semaine, nous allons d'abord entrer dans la phase de planification, dans laquelle nous allons définir des **"user story"** en lien avec l'objectif du sprint qui est : "Générer une relance depuis la liste des factures impayées".

Afin de réaliser cet objectif, plusieurs "user story" ont été créés afin de permettre la réalisation de celui-ci.

B/ Synthèse des us.

Affichage popin de création sur utilisation du CTA

Attach Create subtask Link issue

Description

Objectif du ticket :

Depuis les factures en retard, je dois être capable de relancer une facture depuis la liste sans rentrer dedans.

- En cliquant sur l'enveloppe, la popin de création d'une relance s'ouvre (identique à celle de la facture)
- Je dois avoir exactement les même possibilités que pour la création d'une relance depuis une facture
- Attention : Les boutons d'interaction sont fonctionnel dans les us suivantes

Lors de la planification, les us ont été analysés, potentiellement divisés en plusieurs us et ont reçu une note de difficulté afin de définir la charge de travail, que cela donnerait, comparé à la charge disponible par l'équipe. (Méthode Agile)

Concernant les us, tous se sont axés sur la possibilité de créer une relance à partir de la page listant les factures impayées.

Dans les us effectués, ont à tout d'abord commencer par la suite de la **refactorisation** du code, plus précisément l'ajout de T.U (**Test Unitaire**) avec phpunit dans le but de vérifier le bon fonctionnement de notre code ainsi qu'une protection face aux **effets de bords**, ce que nous verrons plus tard.

Par la suite, nous nous sommes occupées de la modification de la page concernant la liste des factures impayées, pour que dans le futur la page ressemble à la maquette qu'il y a, ci-dessous.

Liste des factures en retard de paiement

Créer une relance

Gérer les modèles de relance

Recherche

Recherche avancée

Client

Retard de paiement

1 à 15 jours

Tout

Relancées

Non relancées

<input type="checkbox"/> N° facture	Date facture	Client	Retard	Total TTC	Restant dû	Nb relances	Dernière relance
F-202000000023	12-11-2021	Alan média	102 jours	152,56 €	152,56 €	2	12-11-2022
F-202000000029	16-11-2021	Agence Dupond	152 jours	36,00 €	10,00 €	-	12-11-2022
<input type="checkbox"/> F-202200000025	12-11-2021	Alan média	101 jours	152,56 €	152,56 €	-	-
F-202000000029	16-11-2021	Agence Dupond	99 jours	36,00 €	10,00 €	-	-
F-202000000023	12-11-2021	Alan média	97 jours	152,56 €	152,56 €	2	12-11-2022
F-202200000058	16-11-2021	Agence Dupond	90 jours	36,00 €	10,00 €	2	12-11-2022
F-202000000023	12-11-2021	Alan média	78 jours	152,56 €	152,56 €	1	12-11-2022
F-202000000029	16-11-2021	Agence Dupond	60 jours	36,00 €	10,00 €	-	-
F-202000000023	12-11-2021	Alan média	59 jours	152,56 €	152,56 €	1	12-11-2022
F-202000000029	16-11-2021	Agence Dupond	44 jours	36,00 €	10,00 €	-	-
Total à percevoir				4995,10 €			

Ont a donc commencer par ajouter une icône d'enveloppe à la fin de la ligne, celle-ci ne devait apparaître que quand, l'utilisateur, passait la souris dessus.

F-20230000001	19-05-2023	Brue	+28 jours	30,00 €	30,00 €	
---------------	------------	------	-----------	---------	---------	--

Ensuite, nous avons ajouté les interactions précédentes que possède déjà le pop-in (Pour rappel : enregistrer, envoyer par mail et par courriel) ainsi que l'affichage d'un message de succès selon l'action définie. Cette partie fût compliquée étant donné que nous avons dû refactoriser le code de la pop-in afin de pouvoir la réutiliser et ainsi éviter la duplication du code.

Après, nous nous sommes occupée de la gestion des erreurs, pour être plus précis, grâce à la refactorisation de la pop-in, nous avons toutes les erreurs concernant la

gestion de la relance, cependant nous avons ajouté des erreurs dans le cas où la pop-in n'apparaît pas, cela est fait dans le but de prévenir l'entreprise en cas de bug.

Et pour finir l'objectif de ce sprint, nous nous sommes occupés d'un oubli, remonté lors du dernier sprint, concernant la signature de la relance, en effet, un utilisateur peut poser sa signature afin d'accompagner le mail/courriel, cependant cette fonctionnalité avait été oublié lors de la mise à jour de l'interface des relances.

Note : Par soucis de confidentialité, je ne pourrais pas vous montrer de code

IV/ Vocabulaire.

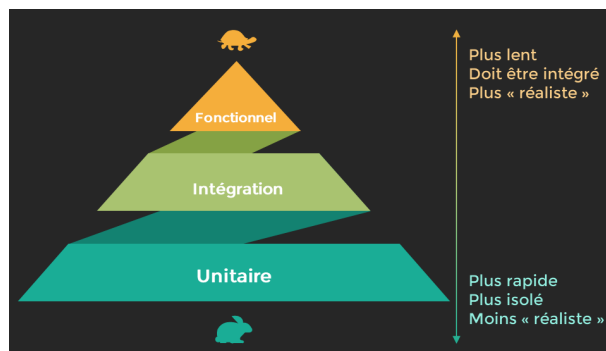
Refactorisation : Amélioration et optimisation du code source d'un projet sans y ajouter de fonctionnalités et corriger des bugs.

Test Unitaire : procédure permettant de vérifier le bon fonctionnement d'une partie précise du code, l'addition de toute les T.U donne ce qu'on appelle le "code coverage" qui représente la couverture de tout le code en pourcentage.

Effets de bords : Effet négatif inattendu suite à la modification du code dans un endroit ayant "aucun rapport" avec.

V/ Approfondissement : Les T.U en php.

Comme expliqué dans la partie Vocabulaire, Les T.U sont des procédure permettant de vérifier le bon fonctionnement d'une partie précise du code, l'addition de toute les T.U donne ce qu'on appelle le "code coverage" qui représente la couverture de tout le code en pourcentage. (Image de geraudwilling.com)



Les T.U font partie de la base de la pyramide des tests, son but est détecter rapidement, lors du développement de la solution, les bugs, oublis ou tout autre type d'erreurs techniques, cependant ceci ne se rapproche pas de la réalité, c'est à dire ce que les utilisateurs finaux rencontreront.

A/ Prérequis :

Il faudra avant tout les choses suivantes :

- PHP
- Phpunit (qui doit être dans une version supportée par la version de php choisie)

Par la suite il faudra configurer phpunit dans le fichier phpunit.xml, dans mes exemples, je me servirai de la configuration faite par l'entreprise.

B/ Créer un test unitaire

Le but d'une T.U est de vérifier la fonction que nous avons créée, il serait donc logique de se dire qu'il faut créer x T.U pour x fonctions, cependant il faudra prendre en compte la valeur, son importance, sa testabilité de la fonction dans le code.

Une fois cela définit nous allons les regrouper par type de T.U (le groupe concerné tel que l'interaction avec la base de donnée) puis nous avons créé notre classe php qui étendra de la classe "Base Test" (Classe améliorée de l'entreprise qui reprend la classe "Test Case" de phpunit).

Le cas que je vais vous présenter concerne les T.U que nous avons fait, cette semaine, concernant l'interaction avec les modèles de relances, c'est-à-dire dans la page de paramétrage et dans les pop-in que nous avons créés.

La T.U suivante permet donc que notre modèle passent bien en favori quand on le lui demande :

```
<?php
```

```
// Namespace -> Espace de nom du fichier (On peut donc l'appeler avec "use Tests\Recovery")  
namespace Tests\Recovery;
```

```
// Classe dont on a besoin.
```

```
use Evoliz\Orm\Models\TemplateRecovery;  
use Evoliz\Orm\Models\TemplateRecoveryFavorite;  
use Evoliz\Recovery\CreateTemplateActionResult;  
use Evoliz\Recovery\MakeFavoriteTemplateActionResult;  
use Evoliz\Recovery\RecoveryTemplateActionResult;  
use Evoliz\Recovery\Services\RecoveryTemplateService;  
use Evoliz\Recovery\UnfavoriteTemplateActionResult;  
use Evoliz\Recovery\UpdateTemplateActionResult;  
use Tests\BaseTest;
```

Julien LIBERT, 4 days ago • SCRUM-1

```
/**
 * Ont met notre T.U dans un groupe de T.U
 * @group recovery
 */
... | 0 references | 0 implementations
class RecoveryTemplateServiceTest extends BaseTest
{
    /**
     * Ont met notre T.U dans un autre groupe de T.U
     * @group db
     */
    0 references | 0 overrides
    public function testFavoriteTemplateSuccess()
    {
        // Ajout de valeurs fictive dans la DB
        $templateRecovery = $this->evzFactory
            ->factory(TemplateRecovery::class)
            ->create();

        // Instanciation de l'objet service de la classe RecoveryTemplateService
        $service = new RecoveryTemplateService($templateRecovery->COMPANYID);
        // Met le template en favoris en appelant la methode et récupere le résultat.
        $result = $service->favoriteTemplate(new MakeFavoriteTemplateActionResult(), $templateRecovery->TEMPLATEID);

        // Ici ont vérifie tout les résultat attendu. assert"valeurattendu"(valeurattendu, variable)
        // Est-ce que mon objet et bien un objet de la class RecoveryTemplateActionResult
        $this->assertInstanceOf(RecoveryTemplateActionResult::class, $result);
        // Verifie si le résultat contion la property "actionSuccess" qui doit être à True
        $this->assertTrue($result->actionSuccess);
        // Met templateId corresepond-t-il ?
        $this->assertEquals($templateRecovery->TEMPLATEID, $result->templateRecovery->TEMPLATEID);
        // Est-ce que met valeurs sont bien bien dans la DB?
        $this->assertDatabaseHas('TemplateRecoveryFavorite', [
            'TEMPLATEID' => $templateRecovery->TEMPLATEID,
            'COMPANYID' => $templateRecovery->COMPANYID,
        ]);
    }
}
```

Une fois le test fini, on peut le tester avec la commande "docker exec -w //app dockerconf-evz-1 vendor/bin/phpunit testing/Orm/Models/UserTest", cette command permet d'exécuter phpunit qui est situé dans un conteneur docker.