Diapo planning :

Je vais maintenant vous présenter notre agenda du projet. Nous avons premièrement planifier les tâches à effectuer en fonction du nombre de jours de la mission à l’aide du rétro-planning que vous pouvez observer à l’écran. Nous avons une mission courte de 5 jours donc nous nous sommes attelés à respecter précisément ce planning. Nous avons commencé le jour 1 à installer les différents outils comme Apache et Postgres et préparer les environnements de test. Puis nous avons à l’aide d’une matrice sélectionner les TNR  à automatiser, commencé à  estimer le chiffrage de la rédaction des tests automatisés et à développer ces tests pendant 3j et demi. Les deux derniers jours donc le 11 et 12 nous avons parrallèlement préparé le POC (proof of concept) de la mission.

Diapo JDD 1 :

Dans le cadre de la réalisation du développement de nos tests nous avons fait appel à des fichiers properties.Un fichier properties est un fichier dans lequel on peut stocker des informations. Nous avons utilisé ce type de fichier pour variabiliser notre projet et surtout mutualiser les variables au sein d’un même fichier..Il a comme format clé=valeur : Ce que vous pouvez voir à l’écran est le contenu du fichier.

Diapo JDD 2 :

Nous avons créé une méthode d’initialisation et d’utilisation de notre fichier properties dans notre class Socle technique qui est une sorte de trousse à outils et qui est extends par toutes nos classes du projet pour avoir accès aux fonctions de cette cette classe dans nimporte quelle autre classe.

Diapo JDD 3 :

Nous avons créer une classe de type “interface” qui contient les variables de type String qui vient récupérer la valeur de la clé correspondante dans le fichier properties à l’aide de la fonction propertyJDD.getproepriete. Ttt ces variables String sont utilisables dans nos class de test qui sont implémentés par notre classe interface comme vous pouvez l’observer siur la derniere image par l’appel de la méthode creer participants.

Diapo PageObject1 :

Pour développer nos tests, nous nous sommes basés sur la méthode de PageObject. Le principe est de modéliser chaque écran sous forme d’une classe JAVA comme vous pouvez observer sur la première image nous avons une classe pour …..Et pour définir les pages dont nous avons besoin nous avons suivi la logique des tests cad par exemple pour le test créer des participants, nous accédons en premier par la pageconnexion puis la page accueil ensuite la page liste participants et pour finir la page de creation nous avons donc défini 4 pages voir 5 nous le verrons dans la slide suivante.

Diapo PageObject2 :

N Nous avons constaté que pour chaque page rencontrée dans le test le Bandeau Menu au dessus est toujours présent. Nous avons donc créer une classe java à part entière pour ce Bandeau appelé pageBandeau comme vous l’observer sur l’écran

Diapo PageObject3 :

Voici sur l'image de gauche, le code qui définit la pageBandeau. Nous avons un constructeur et des web élements qui compose le bandeau qui vont être utilisable sur tous les autres pages objects à l’aide de la notion d’héritage de java et le mot clé extends.

Diapo PageObject4 :

Comme nous l’avons énoncer précedemment nous avons écrit des méthodes dans la page bandeau comme par exemple pour accéder à la liste calendriers et liste des participants qui vont pouvoir être appelé dans nos classes de test par leur nom et  par nimporte quelle page qui hérite de la classe Pagebandeau.

Diapo Assertions1 :

JUnit contient en plus des instructions Java habituelles des instructions de validation appelées assertions donc pour vérifier que les éléments sont bien présents sur la page par exemple. Dans nos cas de tests, nous avons toujours besoin de vérifier le résultat attendu comme le montre l’image à gauche nous devons vérifier que la page contient bien un certain nombre d’éléments. Pour cela, nous utilisons la méthode d’explicit Wait donc attente explicite de java qui va permettre d’attendre que l’élément par exemple titre liste des participants soit visible avant de continuer le test

Diapo Assertions2 :

Voici un autre type d’assertion qui permet de s’assurer de l’état des éléments à l’écran. Par exemple pour vérifier que les champs prenom, nom et ID sont bien non renseignés on utilise la fonction “getattribute” sur l’attrubut valeur et on vérifie que cette valeur est nulle.Pour vérifier que ressource normale est sélectionné on utilise la fonction isselected qui permet de s’assurer que l’option est bien sélectionnée dans la liste comme pour le bouton radio “utilisateur lié.

Diapo Estimation ROI 1 :

Nous avons estimé un ROI retour sur investissement de nos 6 tests automatisés de la manières suivantes : nous avons découpé les coûts en deux catégories ; initial qui englobe rédaction des cas de tests et étude (faisabilité,roi,..) et les coûts d'exécution qui regroupe execution des cas de tests, l’analyse des erreurs et la maintenance de ces cas de tests. Puis nous avons estimé une charge pour chacune de ces étapes en jour/hommes que nous avons regroupé dans le tableau. Pour la redactiion…. Et pour estimer un Retour sur investissement nous avons appliqué la formule qui est affiché à savoir cout total = cout initial + nombre d’exécutions x cout exécution

Diapo Estimation ROI 2 :

Ce qui nous permet d’obtenir le bilan et le graphique suivant ; on peut observer que pour affirmer que l’automatisation soit rentable il faut 130 exécutions des 6 cas de tests chacun ce qui représente 170 jours/homme. On peut donc conclure que le retour sur investissement est très long. Et je vais laisser Mathis vous exposer les recommandations à ce sujet .