CRR Point Projet

25 février 2013

Planning de la réunion :

20 minutes :

- Revue des tâches
- Décisions
- Next steps

15 minutes :

- Code review

30 minutes :

- Point entre membre sur points spécifiques à partager
  - Feedback V1
  - Configuration VPN

Contenu :

Revue des tâches :

- Antoine
  - Ajout plugins jenkins
  - Cahier des tests fait à la place d'Eric
  - Pres préparée
- Eric
  - Dev commentaires
  - Services + tests workflow
- Tarik
  - workflow v1 : à check ce qui manque / faire
  - Commencé workflow v2 : écrans + controllers
  - test selenium workflow à faire : ceux qui ne bougeront pas
- Rudy
  - services v1 + v2 ceux dans interfaces + unit test
- Mélody
  - création de compte
  - vacances !

Informations / Contraintes à venir:

- Antoine déplacement Portugal cette semaine, peu de dispo (pas internet le soir)

Décisions :

- Projet
    - Utilisation onglet V2 uniquement à partir de maintenant
    - Chaque tache de dev mettez vos estimations
- Spec
    - ajout jeux de données + appels quels DAOs
- Dev
    - Cf excel de tâches

Feedback :

\+ Investissement pour la V1

\- Manque de pilotage du code review =>. Eric à prendre en charge.

\- Pas assez de résultats / temps passé partie workflow. => Eric/Tarik se coordonne aujourd'hui

\- Pas assez de temps sur affichage pour rendu client => Tarik sur le workflow, Antoine a une tache de CSS

\- Pas se fier uniquement à maven, checker l'appli déployée en développement de services, prendre plus de recul => Une personne par semaine qui fait des tests utilisateurs, Mélody

Prochain point : lundi 4 mars - 10h

Remarque développement qualité :

UTILISER JSTL :

- utiliser JSTL, si vous faites des if else etc dans vos JSP c'est que vous avez mal implémenté/pensé la logique de vos services en amont, une jsp affiche des données, elle fait pas de traitement, du coup en vous forçant à utiliser JSTL, cela vous forcera à ne pas faire du java pur dans les JSP via des scriptlet <%%>. De plus, comme vue en NTSI, ça gère les NPE automatiquement, c'est plus lisible et ça gère la portée des variables context > page > session > request automatiquement aussi.
Merci de corriger vos devs et de prendre cette considération pour les prochains devs.

conséquence :
votre logique métier étant externaliser dans les vues, on est donc liée fortement à celle ci et on se retrouve dans l'incapacité de switcher de mécanique de vue, ou de transformer vos Services en WebServices par exemple pour une éventuelle compatibilité mobiles ou otre.

Solution :
Si vous vous retrouver au sein de votre Jsp et que vous vous dites mince il me faut A et B aussi, la solution n'est pas de faire un booleen qui vous donne le choix entre A et B, car le jour où il vous faudra C, vous devez tout (en partie) refaire...
Vous devez plus penser en terme d'ensemble, votre jsp n'affiche que des element
Si j'ai A je l'affiche, si j'ai B je l'affiche aussi et du coup votre service renvoi du A,B,C ...etc mais c'est lui qui fait le tri, tester en amont dans votre service au lieu de le faire dans votre JSP.

Si vous souhaitez juste afficher cacher des élements HTML, votre service ne doit pas faire ce travail, utiliser plutot du Javascript via Jquery hide() ou Show()

La bonne pratique est de se dire que lorsque je charge ma JSP, j'ai tout les éléments à disposition que je souhaite afficher et la seule chose sur laquelle je dois penser c'est comment mettre en forme ces données via HTML et CSS.

UTILISER DES CONVENTIONS DE NOMMAGE :

- Lorsque vous faite du HTML/CSS/JS, pour assurer que vos id ou vos class soit propres à votre code et ne vient pas impacter le reste, préfixer vos identifiants par le nom de votre composant.jsp

exemple : login.jsp aura un des id qui commence par login_XXX pareil pour les class et le javascript

explication :
- dans le workflow de reservation, on fait une dataTable sur une classe .dataTable
- dans le rating, on fait une dataTable sur une classe .dataTable
cela genere un bug qui cree une popup affichant que l'on tente de creer 2 fois la datatable sur le meme element html.

Consequence :
cela empeche des tests selenium à s'executer car la popup bloque les actions du webdriver, du coup le mvn verify ne passe plus, donc mvn install ne genere plus le war, et on perd du temps à corriger.
Cet exemple illustre comment on peut à partir d'une simple convention de nommage faire echouer des tests d'integration et faire gagner du temps de developpement.

Solution 1 : on met la construction des dataTable dans le footer pour toutes les construire en même temps, cela implique une convention de nommage global des dataTable et d'éventuel problème de génération de dataTable si celle si apparait une fois que la ligne de code est déjà jouée, donc c'est 1 ligne de code mais plus complexe à gerer (cf CRM365 les appel jquery dans footer)

Solution 2 : on préfixe chaque propriété par <composant>_<propriete> pour garantir l'unicité des éléments

Code review
> Cf sonar VPN