Compte rendu réunion méthodologique avec Mr. Brenner le 15/11/2012

Auteur : Rafik.B

Bon comme vous le savez il a commencé avec la partie de Giraud ensuite il a parlé de ses documents a lui, donc je vais respecter ca dans le compte rendu

Ensuite, il a fait des remarques pour chaque document, en plus de remarques générales, donc je vais vous faire les remarques une par une ensuite les remarques générales !

Partie Giraud :

Benchmarking :

- Est-il vraiment utile d'avoir ne base de données ? est ce que c'est interessant d'n utiliser une ?
- Il faut faire un diagramme UML de l'architecture technique (meme si on est sur qu'elle va changer plus tard)
- Diagramme déploiement / composant

Architecture :

- Il faut que le 25 Novembre on ait une architecture technique complete et prete, que des truc issus du Benchmark
- Il faut travailler etroitement avec celui qui fait le Benchmark

Plateforme d'intégration :

- Documenter tout ce que tu fais, et formes les autres à l'utiliser
- Exemple avec du code !
-

Partie Brenner

RoadMap :

- C'est à toi de faire l'outil de plannification (Rally / Version One) et Puisque c'est Hamza.L qui a beaucoup avancé sur cette partie il faudra que tu l'aides sur la plateforme d'intégration
- Il faut faire une simulation de tout un projet (un faux) et le mener jusqu'a la fin avec Rally ou Version One (normalement ca a été fait par Hamza.L) avec estimation des charges (modification / ajoute etc)

- - Documenter pourquoi le choix s'est porté sur Version One et non Rally Dev
- Il faut faire la liste complete UC / contrainte / taches transverses pour le Vision Do

Vision Doc :

- Il doit etre avancé par rapport au UC Model
- L'equipe ne fait pas partie des acteurs
- Product owner doit faire partie des acteurs (il represente la société de transport)
- Détails sur le client (société)
- Si on ne trouve pas un product owner, il faut voir quelqu'un d'autre qui s'y connait au moins un peu ou qui est a peu pres dans le meme domaine (un domaine ou il y'a de la capture d'information => transmission d'information a une appli)
- Il ne faut pas se baser sur les besoin d'un metro du style Paris, New York .. etc mais plutot sur un petit metro, d'une ville qui « commence » a faire un metro par exemple (penser a Creteil, .. etc) => il faut voir petit, pas trop ambitieux
- Bien connaitre le Sujet ISIAD et la « taille » de leur application parcequ'elle a un réel impart sur la performance
- Volume d'information qui remonte => ca doit etre concis dans le Vision Doc
- Réduire le champs de possibilités (Volumetrie)
- Consulter les ISIAD parcequ'ils reflechissent déja sur le besoin métier et ils sont fortement interessés par notre produit
- Le schéma des liens entre applications est important
- La replication n'est pas une fonctionnalité principale

UC Model

- RTDRS [un ESB n'a pas de boisoin ni d'objectifs]
- EFIC n'est pas LE SEUL composant qui collecte les information (ouverture des portes, pilotage automatique, ... TOUS les composants terrains (le SI qu'il y'a derriere les equipements) donc il faut piser = contrainteas les citer tous dans le UC mais trouver un moyen de généraliser en UML
- Il faut mettre un admin RTDG
- HQMR (je sais pas exactement ce qu'il a dit a prpos de ca)
- Il ne faut pas mettre les Mock dans le UC model
- Il faut briefer les use case, parceque on ne comprends pas ce que fait chacun d'eux dans le diagramme
- Il faut sequencer les use cases dans le diagramme UML, il faut pas directement mettre « prioriser les messages » par exemple mais : collecter info => traiter info => prioriser .. etc
- Il faut bien comprendre ce qui fait partie du systeme (dedans) et ce qui est exterieur (acteur) on ne peut pas avoir des use case qui soient en meme temps dedans/dehors
- L'administrateur ne fait pas la priorisation des messages, il fait autre chose, il y'a des composants informatiques qui font ca, donc il faut faire apparaitre QUI fait la collection des infos, QUI remonte ces info, QUI lance la procedure , qu'est ce qui fait qu'il y'a une detection,

comment ca demarre .. il ne suffit pas de dire un SI (d'apres ce que j'ai compris il faut vraiment entrer dans le détail)
- Nommer les fleches extends/ include
- On fait quoi avec un message du centre ? EHQMR (il y'a surement un truc entre les deux dans le UC)
- Il faut reflechir a la nature des messages envoyés par l'ISIAD
- Les messages qui ne passent pas par le centre doivent apparaitre dans le diagramme

Analyses de risques :

- Il faut determiner un risque principal
- Chaque risque doit avoir un plan d'action PRECIS en indiquant précisemment les mesures à prendre
- Le risque « mauvaise estimation taille projet et reccueuil du besoin« est un vrai risque mais qui a une probabilité tres faible (explication dans les remarques generales)
- Si on dit qu'on ne maitrise pas tel ou tel outil, il faut que le plan d'action soit tres precis, pas « aller se documenter a la BU » mais plutot indiquer precisement quel livre .. etc
- Par exemple ne pas maitriser les algo de réplication est un risque interessant
- Il faut revoir les contraintes (travailler en partenariat avec Khaoula)
- Pour chaque risque il faut voir si c'est important pour le projet, à quel point, la gravités, acceptabilité et comment éliminer ces risques

Plan projet

- Identifier les lacunes individuelles, ne pas faire de généralités (pareil que soukaina, ca doit etre precis)
- Qu'est ce que l'integration continues ? (1 build chaque combien ? 5 minutes ? chaque jour ? chaque semaine ?)
- Il faut controler les livrables de chacun, et faciliter la synchronisation
- - mettre a jour les taches
- Pendant les réunions ne pas généraliser, il faut du concret : réunion à telle date, l'ordre du jour, la durée .. etc) et il fait mettre des infos en relation directe avec notre projet seulement
- Il ne faut citer que les réunions importantes.
- Ce que j'ai mis entre accolades je pense que c'est ton role en tant que Scrum master, pas concernant le document

Remarques générales sur le Projet

- Dans le projet il ne faut ecrire que des choses qui ont un rapport direct avec notre projet , pas dé généralités

- Une mauvaise estimation des besoins ou de l'objectif du projet n'est pas possible parceque si c'est le cas les profs nous diront ce qu'ils attendaient vraiment pour ne pas etre hors sujet, mais dans ce cas la note sera pourrie ! parcontre si on reccueille bien le besoin et si on a tout bon, la note sera meilleure !
- Tout le monde doit savoir faire un Build à la fin de la R0
- Il faut toujours montrer nos livrables au profs (Agile : rassurer le client)
- Dialgouer avec les ISIAD
- Dans la R0 il ne faut pas trop aller dans les details du besoin (débordement sur les autres releases) mais plutot comprendre « juste ce qu'il faut « pour finir la R0
- Il faut penser a l'evolutivité de l'application, par exemple penser a un jeu de parametres pour adapter l'application a un autre client par ex
- Une mauvaise application ISIAD ou pas finie => mauvaise note pour les deux MAIS le projet n'est pas un echec
- Le coef heure / SP ne doit pas etre choisi / défini, pour l'instant 3H / SP c'est peu,
  Explication :
- Si 3H / SP alors une reunion par exemple devrait faire 7 heures, et les grosses US (complexes) vont etre de l'ordre de 1000

Remarque : j'ai tapé ce compte-rendu en cours d'ILOO donc il se peut que j'ai dit des conneries, dans ce cas n'hesitez pas a remettre en cause ce que je dis.