**Travaille De :ayoubsaddi\_nouranesafi\_hamzabalty\_taissirsdiri\_faouziyahyahoui**

**Fiche descriptive Activité 2 / Processus Unifié**

* Remplir le tableau suivant tout en ajoutant les différences entre les 3 méthodes et pour certains critères, évoquer quand et comment ils sont appliqués au niveau de différentes méthodes :

|  |  |  |  |
| --- | --- | --- | --- |
| **Critères de comparaison** | **Scrum** | **RUP** | **2TUP** |
| **Famille de méthodologie** | Méthode agile | Méthode lourde | Méthode lourde |
| **Spécification des besoins** | Priorité aux personnes et aux interactions sur les procédures et les outils ;  • Priorité aux applications fonctionnelles sur une documentation pléthorique ;  • Priorité à la collaboration avec le client sur la négociation de contrat ;  • Priorité à l'acceptation du changement sur la planification. | Par cas de utilisation(tout va être fait addaptent le point de vue de utilisateur) | Besoins fonctionnels exprimés par les utilisateurs  Éléments :  Cas d'utilisation organisé en packages  Acteurs  Activités  Interactions entre objets  contraintes dynamiques |
| **Architecture système** | Architecture en sprints | Architecture en couches et vues. | Le développement d’un système en forme Y, est composé :  - Une branche fonctionnelle  - Une branche technique  - La phase de réalisation |
| **Itération (en nbr et en contenu)** | Des sprints  Max 4 sprints | 3 itérations  -initial  -élaboration 1  - élaboration 2  -construction 1  - construction 2  -transition 1  -transition 2 | N itérations  -Itération 1 : Développer les fonctions de validation du principe du système.  - Intégrer les outils prévus pour le développement.  - Itération 2 :  - Centrée sur l’architecture  –Prototype de réalisation technique.  - Itération 3 :  -Réaliser les fonctions les plus prioritaires de manière à présenter une première version de déploiement pour les utilisateurs.  - Améliorer et compléter l’architecture technique.  -Itération 4. N :  - Réaliser des fonctions jusqu’à l’obtention complète du système initialement envisagé. |
| **Incréments et/ou artéfacts** | Backlog  Burndown chart  Scrum board  produit | -Élément d’information  -produit ou utilisé lors d’une activité de développement logiciel(source, modelé) | -validation fonctionnelle : validation du principe du système  -validation technique : focalisation sur architecture : prototype  -réalisation des fonctions les plus propriétaire  -finition |
| **Les acteurs/ rôles** | Product owner  Scrum master  L’équipe | -Concepteur  -Rédacteur D utilisation  -Analyste systeme  -Développeur  -Architect | •le système : est une boite noire  • les acteurs l'entourent et sont reliés à lui |
| **Gestion des risques** | Ensuite à chaque sprint:  -les risques sont **identifiés** à tous les RDV du sprint (réunion de planification, Daily SCRUM, revue de sprint, rétrospective)  -les risques sont **analysés et gèrés** (recherche de diminution ou autre stratégie contenir, accepter, éviter) à tous les RDV du sprint  (réunion de planification, Daily SCRUM, revue de sprint, rétrospective)  -les risques sont **surveillés en permanence** grâce au RISK BOARD présent sur le radiateur d’informations, et grâce au Burndown Chart | •Les risques sont atténués dès le début du projet  •Identifier les risques.  • Maintenir une liste de risques tout au long du projet. | •Les risques d’inadéquation aux besoins ou d’inadaptation technique.  •les risques d’incapacité à intégrer les technologies |
| **Avantages** | - vise à satisfaire au mieux les besoins du client tout en maximisant les probabilités de réussite du projet.  - La méthode Scrum soutient la livraison rapide et régulière de fonctionnalités à haute valeur ajoutée. | - permet de s’assurer de toujours répondre aux besoins des clients en temps et en heure.  - permet d’utiliser une architecture qui est basée sur les composants  -contrôle le long du développement. | - facilite la répartition du travail  -la documentation apportée par les modèles facilite les textes, l'intégration, aide à identifier les sources d'erreurs. |
| **Inconvénients** | - les difficultés arrivent lorsque l'équipe de projet change ou lorsque la maintenance logiciel doit être faite par une autre équipe.  -adaptable seulement pour les projets de taille petite. | - difficilement possible de faire des mises à jour fréquentes avec la méthode RUP.  - moins rapide et adaptable que les méthodes Agile ou Scrum.  - RUP 4 fois plus lent que Scrum. | -Fait tout, mais lourd.  - Parfois difficile à mettre en œuvre de façon spécifique. |
| **Types de projets adéquats** | Petit projet | Adapter à tous les projets | Tous les types du projet. |
| **Outils utilisés sur le marché** | IceScrum, Xplanner  Le tableau blanc | UML, USPM |  |