

Compte-Rendu Réunion

19 Octobre 2016 / 10:30-12:00

Rencontre avec le Responsable Pédagogique

PARTICIPANTS : TOUTE L'ÉQUIPE DE PFA ET JULIEN ALLALI

SECRÉTAIRE : ELODIE FENG

PLAN :

1. Discussion autour de notre sujet de PFA
2. Présentation de méthodologies de développement logiciel
3. Présentation de la machine de découpe laser et du cas général d'utilisation

1. Discussion autour de notre sujet de PFA

- Le **projet Plater**, trouvé sur le net, est très proche de celui du client, même s'il prend des objets en 3D en paramètre : il faut en discuter avec le client pour savoir s'il souhaite qu'on parte de ça, qu'on s'en inspire, ou rien du tout. M. Allali nous recommande de « ne pas réinventer la roue », et donc de piocher des éléments.
- Le « **Texture Packing** » n'est pas vraiment le cadre de notre projet contrairement à ce qui est écrit et demandé sur le sujet. En effet ce problème concerne uniquement des **rectangles** et ne permet **pas la rotation autre que 90°** de ces derniers (d'après M. Allali)
- M. Allali a discuté du sujet avec Grégoire PASSOT (?) : les termes du sujet leur semblent mal choisis (trop orientés vers la 3D et non la 2D), cela peut indiquer que le client connaît mal le domaine de son projet et il faut donc l'éclairer sur l'existant et **être force de proposition**.
- Attention tout de même lors de **propositions** : il est conseillé de traduire les différentes options en **temps de travail** (\$).
- Nous a été présentée une **planche d'exemple** ayant servi à la découpe laser de nombreuses pièces : dans ce cas, l'agencement des pièces a été fait à la main par M. Allali (fastidieux et non optimal). Mise en avant de la nécessité de respecter une **distance de sécurité** entre deux pièces.

2. Méthodologies de développement logiciel

• Cycle en cascade

Étude du besoin

Cahier des charges

Architecte

Développeur

Recette

Livraison

- Communication de proche en proche entre les différents intervenants
- Grosse contractualisation avec le client
- (+) Responsabilités bien délimitées entre les différents rôles, rassurant
- (-) Personne n'assume plus aucune responsabilité en cas de problème

● Cycle en V



- (+) Cela donne un projet très bien testé, en THÉORIE
- (-) Ça coûte cher, donc souvent on finit dans une méthodologie Cascade
- (-) Demande énormément de rigueur en amont, pas évident de se poser toutes les bonnes questions au début
- Problème avec ce type de méthode : le projet est souvent long, et pendant ce délai les attentes du client changent forcément. Or on ne s'en rend compte qu'à la livraison ??!
- Autre problème : La communication écrite est toujours de mauvaise qualité comparée à des échanges oraux. C'est toujours flou, sujet à interprétation... À l'oral on voit mieux les intentions, on peut interagir, et préciser les points compliqués.

● Méthodes Agiles

- Il en existe plein, toutes reposent sur un petit document appelé « Manifeste Agile » spécifiant les principes de base.

/! à partir de là c'est parfois difficile de savoir si M.Allali parlait de TOUTES les méthodes agiles ou uniquement de la méthode SCRUM, la plus « connue ».

- S'appuient sur des cycles courts, « sprints », et des réunions de l'équipe 15min/jour !
- Compliqué à mettre en place au début, demande de la discipline.
- Principe de la méthode **SCRUM** :
 - Le Product Owner (en relation avec le client) décroît le travail sous forme de **UserStories**, qui sont des fonctionnalités orientées client. On détermine la **Business Value** de chacune. (c'est relatif aux autres US, pour prioriser)
 - Ensuite avec l'équipe de dev, on détermine la **Difficulté Technique**. Ces notations prennent des valeurs dans la suite de Fibonacci : plus on augmente, moins on est précis.
 - On priorise ensemble les tâches en s'intéressant au rapport **BV / DT**, à maximiser.
 - On fixe également un seuil de difficulté technique à ne pas dépasser.
- Après quelques Sprints on évalue la **Vélocité** de l'équipe de dev. C'est sa capacité en points de difficulté de travail en un sprint. Cela sert à mieux calibrer les prochains sprints.
- On fait régulièrement des **démos utilisateur**, qui sont réelles, quitte à utiliser des MOCKS.
- Norme **INVEST (Indépendance, Négociabilité, Valuabilité, Testabilité, Small ...)** pour les US.
- EPIC = gros chantier futur : c'est une idée, mais qu'on n'a pas encore l'intention de développer.
- Importance du rôle de **SCRUM MASTER**, qui assure la bonne application de la méthode.
- Méthode SCRUM souvent utilisée sur des petites équipes de ~10 personnes.

- Principe de la méthode « **CANBAN** » (KANBAN? KHANBAN ? ...) :
 - Colonnes désignant différents états d'une UserStory, représentée par un Post-It :
- Product Backlog // US INVEST // US ToDo // US en cours // Développement(...) //Validation**
- Ces colonnes possèdent des seuils de capacité, en nb de US ou en pts de DT
 - Chaque US précise sa modalité de test, pour la dernière phase.
 - (+) On travaille en flux tendu
 - (-) Il est facile de s' « endormir » sur le travail car moins de réunions (« Évènements ») sont prévus.
 - (-) Certaines US peuvent en **bloquer** d'autres pendant longtemps ...
- Extrem Coding : évoqué mais non détailler : à aller voir ...

3. Machine découpe Laser

- Important de s'intéresser au MÉTIER CLIENT, pour voir donner son avis sur son besoin, l'aider.
 - Utilisation de InkSkape pour dessiner les formes, également pour les agencer.
 - Étape de calibrage : on vérifie manuellement l'endroit où on souhaite découper la pièce
 - On peut découper (lent: 1% de la vitesse de la machine) et graver (rapide). On indique ça grâce à niveau de couleur dans Inkskape. C'est le logiciel d'impression (propre à l'imprimante) qui spécifie ça.
- [PROPOSITIONS EXTENSION SUJET]
- Existence d'une caméra normalement placée au dessus de la découpeuse laser : idée de scanner les plaques (chutes) pour déterminer les trous et calculer le placement des pièces en fonction
 - Toujours plus : numéroter les plaques, et faire une BDD pour proposer à l'utilisateur d'utiliser telle ou telle plaque.