Équipe 10

Poly Paint Pro
Plan de projet

Version 1.08

Historique des révisions

Date	Version	Description	Auteur
2019-01-29	1.00	Première ébauche	Jérémie Huppé
2019-01-30	1.01	Rédaction des sections 1, 2 et 3.1	Jérémie Huppé
2019-01-31	1.02	Première ébauche de la section 4	Jérémie Huppé
2019-02-01	1.03	Révision des sections 1, 2 et 3.1	Geneviève Bock
2019-02-02	1.04	Rédaction des section 5 et 6	Geneviève Bock
2019-02-02	1.05	Rédaction de la section 4	Jérémie Huppé
2019-02-05	1.06	Révision de l'ensemble du document.	Geneviève Bock, Jérémie Huppé
2019-02-06	1.07	Corrections des éléments révisés. Rédaction des sections 3.2, 3.3 et 3.4. Révision de la section 6.	Geneviève Bock, Jérémie Huppé
2019-02-07	1.08	Révision finale du document	Geneviève Bock, Jérémie Huppé

Table des matières

Plan de projet

1. Introduction

Le présent document définit le plan de projet pour la réalisation de l'application PolyPaint Pro. Il spécifie également l'entente contractuelle proposée pour la réalisation du projet.

La section 2 consiste en un aperçu du projet. On y décrit la solution proposée, les hypothèses et contraintes sur lesquelles repose le plan de projet ainsi que les biens livrables du projet. La section 3 décrit le processus de gestion utilisé pour le projet. On y décrit les mécanismes mis en place afin de gérer les changements d'exigences, la qualité du produit, les risques et la configuration des artefacts. La section 4 détaille l'ensemble des échéanciers pour les lots de travail à réaliser. La section 5 présente les responsabilités des membres de l'équipe ainsi que l'expertise de chacun. Finalement, la section 6 développe l'entente contractuelle proposée.

2. Énoncé des travaux

2.1. Solution proposée

Ce projet consiste en la réalisation de l'application PolyPaint Pro pour la PME PolyApps.

L'objectif du projet est de créer un logiciel, PolyPaint Pro, permettant l'édition de formes sur canevas en mode collaboratif ou en mode individuel et la création de diagrammes UML et de diagrammes de processus. Il permet également aux usagers de communiquer entre eux à l'aide d'une interface de clavardage. Le système communique via un réseau afin de connecter les usagers entre eux. Le système devra assurer une bonne fluidité au niveau de l'édition de formes et la communication entre usagers. L'application PolyPaint Pro devra offrir une interface conviviale aux usagers. Le système devra offrir un profil utilisateur à chaque usager. Des galeries d'images devront être disponibles aux usagers. Le code qui sera rédigé pour implémenter les fonctionnalités de l'application PolyPaint Pro devra se soumettre à des normes de programmation. L'application PolyPaint Pro devra être compatible avec un ordinateur Windows 10 et une tablette Nexus 9 compatible avec la version Android 7.1.1.

Un ensemble d'artefacts accompagnera l'application PolyPaint Pro. Il sera possible de retrouver parmi ces artefacts:

- Document de spécifications des requis du système (SRS)
- Document présentant la liste des exigences
- Document détaillant l'architecture logicielle de Poly Paint Pro
- Document décrivant le protocole de communication client-serveur
- Plan de tests logiciels
- Document énumérant les résultats des tests
- Plan de projet

De plus, un prototype de communication sera développé. Ce prototype consistera en deux interfaces de clavardage: une implémentée sur le client lourd et une sur le client léger. Il incluera également le serveur nécessaire à la communication entre clients.

Il est important de préciser que, du côté du client lourd, l'application PolyPaint Pro devra être construite à partir de l'application PolyPaint fournie par PolyApps. L'application PolyPaint devra donc être adaptée afin de respecter les exigences de l'application PolyPaint Pro. Du côté du client léger, PolyPaint Pro est une toute nouvelle application.

2.2. Hypothèses et contraintes

Le présent plan de projet assume que les contraintes énumérées ci-dessous sont respectées.

2.2.1. Contraintes au niveau du calendrier

Le projet PolyPaint Pro sera réalisé entre le 15 janvier 2019 et le 8 avril 2019 ce qui correspond à une durée totale de 84 jours. Le projet sera divisé en trois jalons. Ces trois jalons seront constitués de 4 itérations à raison d'une itération par semaine. Le premier jalon consiste en la planification du projet et sera réalisé entre le 15 janvier 2019 et le 8 février 2019, soit 25 jours. Le deuxième jalon consiste en la réalisation des fonctionnalités de base de l'application PolyPaint Pro et sera réalisé entre le 9 février et le 8 mars 2019, soit 28 jours. Le troisième jalon consiste en la réalisation des fonctionnalités avancées et sera réalisé entre le 9 mars 2019 et le 8 avril 2019, soit 31 jours.

2.2.2. Contraintes au niveau des ressources

Le projet PolyPaint Pro sera réalisé avec un effectif constitué de cinq développeurs et d'un gestionnaire de projet. Les membres de l'équipe de développement devront consacrer 14 heures de travail par semaine à la réalisation du projet.

2.2.3. Contraintes au niveau de la qualité du logiciel

L'application PolyPaint Pro possède une interface utilisateur simple, facile à utiliser et conviviale. Elle possède un éditeur de diagrammes fluide et cohérent.

2.2.4. Contraintes au niveau des logiciels à utiliser

Le client lourd de l'application PolyPaint Pro devra être basé sur le logiciel PolyPaint fourni par PolyApps.

2.2.5. Contraintes au niveau des technologies à utiliser

Le client lourd de l'application PolyPaint Pro devra être développé avec les technologies C# et WPF afin de respecter les technologies initiales de PolyPaint.

2.3. Biens livrables du projet

Tableau 1. Biens livrables du projet

Livrable	Artefacts	Code	Date
Réponse à l'appel d'offres	● Plan de projet ● SRS ● Liste d'exigences ● Document d'architecture logicielle ● Protocole de communication	● Prototype de communication client lourd-serveur ● Prototype de communication client léger-serveur	8 février 2019
Produit final	● Mise à jour des artefacts remis précédemment ● Plan de tests ● Résultats de tests		8 avril 2019

3. Gestion et suivi de l'avancement

3.1. Gestion des exigences

Pour gérer les exigences notre équipe utilise l'outil Redmine. Cet outil permet à notre équipe de mesurer, rapporter et contrôler l'ensemble des fonctionnalités à développer. Ces dernières sont regroupées par itération et chaque itération correspond à une version dans l'outil Redmine.

Dans Redmine, les fonctionnalités à développer sont représentées sous forme de demandes. La procédure pour créer une demande est la suivante :

1. Attribuer un sujet descriptif à la demande.
2. Assigner la demande à un membre de l'équipe.
3. Assigner à la demande la nouvelle version créée pour l'itération.
4. Indiquer les dates de début et de fin pour la réalisation de cette demande.
5. Attribuer un temps estimé pour la réalisation de cette demande.

L'attribution d'un temps estimé pour chaque demande nous permet d'évaluer les demandes à prioriser durant une itération donnée. Saisir le temps passé pour chaque demande nous permet de mesurer l'avancement de celle-ci, ce qui nous permet d'évaluer si cette demande est en avance, en retard ou à jour avec la planification initiale.

3.1.1. Gestion initiale des exigences

L'ensemble des exigences qui sont initialement connues sont regroupées en lots de travail. Un ensemble de lots de travail à réaliser est assigné à chaque itération.

Au cours d'une itération, notre équipe équipe utilisera la procédure suivante:

1. Saisir le temps passé sur chaque demande dans Redmine.
2. Vérifier si les travaux à réaliser pour l'itération en cours sont en avance par rapport à l'échéancier:
    a. Si c'est bien le cas, un lot de travail, qui n'a préalablement pas été terminé durant l'itération qui le contenait, sera ajouté à l'itération en cours.

À la fin de chaque itération, notre équipe utilisera la procédure ci-dessous afin de clore l'itération en cours:

1. S'assurer que le temps passé sur chaque demande a été saisi pour toutes les demandes de l'itération en cours.
2. Changer le statut des demandes terminées de l'itération en cours pour « Fermé ».
3. Évaluer les lots de travail qui ne sont pas terminés:
    a. Si un lot de travail non terminé est prérequis à un lot de travail de l'itération à venir, alors ajouter ce lot de travail à la semaine à venir.
    b. Si un lot de travail non terminé n'est pas prérequis à un lot de travail de l'itération à venir, alors reporter la réalisation de ce lot de travail à plus tard.

À la fin de chaque itération, notre équipe utilisera la procédure ci-dessous afin planifier l'itération suivante:

1. Créer une nouvelle version.
2. Créer une demande pour chaque fonctionnalité des lots de travail prévus pour l'itération.

3.1.2. Gestion des exigences lors du changement d'une exigence existante

Lorsqu'une exigence existante change notre équipe utilisera la procédure suivante:

1. Mettre à jour l'exigence en question dans le document SRS avec les nouvelles spécifications.
2. Communiquer les changements à l'ensemble de l'équipe à l'aide de notre outil de communication, Slack.
3. Évaluer l'impact des changements apportés à l'exigence, en calculant le nombre d'exigences à modifier, lors du prochain SCRUM.
4. Ajouter ou enlever des demandes dans Redmine pour que celles-ci tiennent compte des changements apportés à l'exigence.
5. Ajuste le calendrier du projet dans la section 4 de ce document.

3.1.3. Gestion des exigences lors de l'ajout d'une nouvelle exigence

Lorsqu'une nouvelle exigence est ajoutée notre équipe utilisera la procédure suivante:

1. Ajouter la nouvelle exigence dans le document SRS.
2. Communiquer la nouvelle exigence à l'ensemble de l'équipe à l'aide de notre outil de communication, Slack.
3. Évaluer l'impact de l'ajout de la nouvelle exigence, en calculant le nombre d'exigences à modifier, lors du prochain SCRUM.
4. Ajouter ou enlever des demandes dans Redmine pour que celles-ci tiennent compte de la nouvelle exigence ajoutée.
5. Ajuste le calendrier du projet dans la section 4 de ce document pour que celui-ci considère cette nouvelle exigence..

3.2. Contrôle de la qualité

La qualité des biens livrables du projet, que ce soit les artefacts, le prototype de communication ou le produit final PolyPaint Pro, sera assurée grâce aux activités de validation avant livraison.

Dans le cas des artefacts, leur qualité est tout d'abord assurée par une vérification avec un expert après leur première rédaction. Ensuite, lorsqu'un artefact contient toutes les sections et les informations nécessaires, deux personnes sont responsables de la relecture de l'artefact et de discuter des améliorations à apporter. Ces mêmes personnes apportent les modifications nécessaires.

Pour ce qui est des biens livrables logiciels, l'assurance qualité est tout d'abord assurée par des tests logiciels. Ces derniers sont décrits dans le document « Plan de tests logiciels ». Celui-ci spécifie la rigueur et les types de tests que nous effectuerons. Il spécifie également que notre équipe fera appel à un groupe d'alpha-testeurs afin de tester le produit final avant sa livraison. Ce test sera réalisé par quelques usagers potentiels de l'application PolyPaint Pro qui n'ont pas développé le produit. Ces derniers pourront cibler des bogues non découverts par les développeurs et pourront pointer certains aspects moins conviviaux du produit. Les commentaires des usagers seront pris en compte dans la planification des itérations à suivre.

L'assurance qualité des biens livrables logiciels est aussi assurée par notre méthode de revue de code. Un développeur est responsable d'une fonctionnalité; il est en pair avec un autre développeur qui est responsable de la revue de code de ladite fonctionnalité. Lorsque la fonctionnalité et ses tests sont implémentés, le critique lit les changements apportés au code et propose des améliorations au développeur. Le développeur suit les conseils pertinents du critique, et ajoute sa nouvelle fonctionnalité à la version stable de l'application.

Finalement, pour assurer la maintenabilité du code, notre équipe de développeurs suit les principes de programmation suivants:

- DRY (*Do not Repeat Yourself)*
- YAGNI (*You ain't gonna need it*)

3.3. Gestion de risque

La description des risques suit la convention suivante :

- Ampleur : sur une échelle de 1 à 10, 10 étant le risque le plus élevé. Cette analyse est basée sur la probabilité d'occurrence du risque, ainsi que ses impacts.
- Description : une description textuelle du risque ainsi que les problèmes attendus.
- Impact : échelle définissant la portée du risque
    - C – critique (affecte le projet en entier)
    - E – élevé (affecte les fonctionnalités principales du système)
    - M – moyen (devrait être maîtrisable en appliquant une stratégie d'atténuation adéquate)
    - F – faible (l'acceptation du risque est une stratégie envisageable)
- Facteurs : aspects du système pouvant être compromis.
- Stratégie de gestion : mesures à prendre afin de gérer le risque.

R1 - Utilisation de librairies externes sujettes à changement				
Ampleur	**Description**	**Impact**	**Facteurs**	**Stratégie de gestion**
6	- Définition: la sortie d'une nouvelle version de librairie utilisée dans le projet n'est pas compatible avec le code source de notre application. - Problèmes potentiels: 1. Incompatible avec d'autres composantes du système. 2. Des fonctionnalités déjà implémentées ne fonctionnent plus. 3. Incompatibilité des tests avec la nouvelle librairie.	M	Maintenabilité Réutilisabilité	Mitiger le risque: Ne pas prendre la nouvelle version disponible, garder la version actuelle de la librairie.

R2 - Limitations de l'interface matérielle du client léger.				
Ampleur	**Description**	**Impact**	**Facteurs**	**Stratégie de gestion**
8	- Définition: l'interface matérielle du client léger offre seulement quelques interactions possibles avec l'application (touchstart, touchend, touchmove, etc.) - Problèmes potentiels. 1. Certaines fonctionnalités seront difficiles à implémenter. 2. Retards sur l'échéancier dus au temps supplémentaire passé à développer les fonctionnalités compliquées. 3. Impossibilité d'implémenter une fonctionnalité.	M	Utilisabilité Convivialité	Éviter le risque: Ne pas développer une fonctionnalité impossible à implémenter ou très difficile à implémenter à cause de l'interface matérielle limitée du client léger.

R3 - Manque d'expertise dans le développement d'applications WPF				
Ampleur	**Description**	**Impact**	**Facteurs**	**Stratégie de gestion**
5	- Définition: les développeurs n'ont pas les connaissances nécessaires pour réaliser l'application. - Problèmes potentiels: 4. Retards sur l'échéancier dus au temps supplémentaire à faire de l'essai-erreur. 5. Complexification des revues de code.	F	Extensibilité Utilisabilité Convivialité	Mitiger le risque: Suivre des tutoriels rapides ou demander l'avis d'experts avant de commencer le développement d'une nouvelle fonctionnalité.

R4 - La source d'un bogue de communication avec Socket.IO entre le client et le serveur est introuvable.				
Ampleur	**Description**	**Impact**	**Facteurs**	**Stratégie de gestion**
7	- Définition: un échange entre le client et le serveur à l'aide d'un protocole de communication provoque un comportement inattendu et ce dernier ne peut être expliqué. - Problèmes potentiels: 1. Retards sur l'échéancier dus au temps passé à déboguer. 2. Correction superficielle qui contourne le bogue sans le régler. 3. Fonctionnalité impossible à avancer ou réaliser à cause du bogue. 4. Impossibilité de corriger le bogue.	E	Fiabilité Stabilité	Contourner le risque: Utiliser un autre protocole de communication, tel que https, pour effectuer la communication dans le cas du bogue.

3.4. Gestion de configuration

Le tableau 2 ci-dessous énumère les artefacts du projet et comment ils sont nommés.

Tableau 2. Artefacts du projet

Artefact	**Nom de l'artefact**
Document de spécifications des requis du système	SRS
Document présentant la liste des exigences	Liste d'exigences
Document détaillant l'architecture logicielle de PolyPaint Pro	Document d'architecture logicielle
Document décrivant le protocole de communication client-serveur	Protocole de communication
Document spécifiant le plan des tests à effectuer sur PolyPaint Pro	Plan de tests logiciels
Document énumérant les résultats des tests de PolyPaint Pro	Résultats de tests logiciels

Chacun des artefacts présentés ci-dessus contient un historique des révisions. Cet historique contient la date, la version, la description et l'auteur des changements effectués. Le format des versions est **x.y**, où **x** débute à 1 et est incrémenté après une remise et **y** débute à 00 et est incrémenté à chaque révision.

Dans Redmine, la gestion de configuration correspond aux versions. Une version est créée pour chaque itération. Les demandes d'une itération y sont liées. Les versions sont nommées de la sorte: **vx.0 - Semaine x**, où **x** représente l'itération, commence à 0 et est incrémenté à chaque nouvelle semaine/itération.

Une demande est associée à chaque exigence d'un lot de travail. Les demandes pour une nouvelle fonctionnalité sont nommées de la sorte: **[NF] x: y**, où **NF** est un acronyme pour **N**ouvelle **F**onctionnalité, **x** représente le numéro de l'exigence spécifiée dans le document « SRS » et **y** représente le nom de l'exigence.

En cas d'un bogue pour une exigence donnée, une nouvelle demande sera créée afin de résoudre ce bogue. Une demande pour la résolution d'un bogue sera créée et liée à la demande de fonctionnalité d'origine. Les demandes pour la résolution d'un bogue sont nommées de la sorte: **[B] x: y**, où **B** est un acronyme pour **B**ogue, **x** représente le numéro de l'exigence spécifiée dans le document « SRS » et **y** représente un nom descriptif du bogue en question.

4. Échéancier du projet

Tableau 3. Échéancier du premier jalon du projet

Planification du jalon 1 réalisé entre le 15 janvier 2019 et le 8 février 2019							
# Itération	Date de début	Date de fin	Lots de travail	Temps estimé (h)	Coûts estimé ($)	Totaux temps estimé (h)	Totaux coûts estimés ($)
0	2019-01-15	2019-01-21	Rédaction artéfact SRS	40	5000	105	13 125
			Rédaction artéfact plan de projet	40	5000		
			Rédaction artéfact liste d'exigences	25	3125		
1	2019-01-22	2019-01-28	Rédaction artéfact d'architecture logicielle	40	4160	80	8 320
			Rédaction artéfact protocole de communication	40	4160		
2	2019-01-29	2019-02-04	Prototype de communication client lourd-serveur	40	4160	80	8 320
			Prototype de communication client léger-serveur	40	4160		
3	2019-02-05	2019-02-08	Assurance qualité, phase de tests et marge de manoeuvre en cas de retard.	15	1560	15	1 560
Fin du jalon 1						280	31 325
Livrable: Réponse à l'appel d'offres dû pour le 8 février 2019							

Tableau 4. Échéancier du deuxième jalon du projet

Planification du jalon 2 réalisé entre le 9 février 2019 et le 8 mars 2019							
# Itération	Date de début	Date de fin	Lots de travail	Temps estimé (h)	Coûts estimé ($)	Totaux temps estimé (h)	Totaux coûts estimés ($)
4	2019-02-09	2019-02-18	Client lourd: Comptes utilisateurs	6	624	131	13 624
			Client lourd: Ajout des formes de bases	25	2600		
			Client lourd: Sélection et déplacement d'éléments	15	1560		
			Client lourd: Synchronisation d'une modification avec le canevas collaboratif	25	2600		
			Client léger: Comptes utilisateurs	6	624		
			Client léger: Réinitialisation de canevas	4	416		
			Client léger: Ajout des formes de bases	25	2600		
			Client léger: Synchronisation d'une modification avec le canevas collaboratif	25	2600		
5	2019-02-19	2019-02-25	Client lourd: Réinitialisation de canevas	4	416	114	11 856
			Client lourd: Gestion de la pile de modifications d'un usager	15	1560		
			Client lourd: Duplication d'éléments et couper et coller des éléments	15	1560		
			Client lourd: Redimension d'éléments et rotation	20	2080		
			Sélection et déplacement, et effacement d'éléments	25	2600		
			Client léger: Chargement et sauvegarde de canevas à distance	20	2080		
			Client léger: Effets visuels et sonores	15	1560		
6	2019-02-26	2019-03-04	Client lourd: Chargement et sauvegarde de canevas à distance	15	1560	95	9 880

			Client lourd: Ajout d'une forme de connexion	25	2600		
			Client léger: Gestion de la pile de modifications d'un usager	15	1560		
			Client léger: Duplication d'éléments et couper et coller des éléments	15	1560		
			Client léger: Redimension d'éléments et rotation	25	2600		
7	2019-03-05	2019-03-08	Client lourd: Ajout des différents types de formes de connexions	15	1560	80	8 320
			Client Lourd: Clavardage avancé (Ajout des canaux de discussion)	15	1560		
			Client léger: Galerie d'images publiques et privées	20	2080		
			Assurance qualité, phase de tests et marge de manoeuvre en cas de retard.	30	3120		
Fin du jalon 2						420	43680

Tableau 5. Échéancier du troisième et dernier jalon du projet

Planification du jalon 3 réalisé entre le 9 mars 2019 et le 8 avril 2019							
# Itération	Date de début	Date de fin	Lots de travail	Temps estimé (h)	Coûts estimé ($)	Totaux temps estimé (h)	Totaux coûts estimés ($)
8	2019-03-09	2019-03-18	Client lourd: Modification du style des formes	20	2080	85	8 840
			Client lourd: Galerie d'images publiques et privées	15	1560		
			Client léger: Protection de base des images	20	2080		
			Client léger: Ajout d'une forme de connexion	30	3120		
9	2019-03-19	2019-03-25	Client lourd: Protection de base des images	20	2080	80	8 320
			Client lourd: Protection avancée des images	20	2080		

			Client léger: Ajout des différents types de formes de connexions	20	2080		
			Client léger: Clavardage avancé (Ajout des canaux de discussion)	20	2080		
10	2019-03-26	2019-04-01	Client lourd: Historique de modifications	25	2600	100	10 400
			Client lourd: Génération automatique de classes	30	3120		
			Client léger: Ajout de notifications pour la réception de messages, d'effets visuels, d'effets sonores et de gestures	25	2600		
			Client léger: Protection avancée des images	20	2080		
11	2019-04-02	2019-04-08	Client lourd: Ajout conditions sur les formes de connexions	10	1040	115	11 960
			Client lourd: Boutons d'alignement d'éléments	15	1560		
			Client lourd: Tutoriel de base	10	1040		
			Client léger: Ajout outils de rotation et de redimensionnement	20	2080		
			Client lourd: Tutoriel de base	10	1040		
			Assurance qualité, phase de tests et marge de manoeuvre en cas de retard.	50	5200		
Fin du jalon 3						380	39 520
Livrable: Produit final dû pour le 8 avril 2019							

5. Équipe de développement

L'équipe de développement du produit est composée de 6 développeurs: Antoine Daigneault-Demers, Jérémie Huppé, Maxime Gosselin, Mathieu Giroux-Huppé, Alexandre Rault et Geneviève Bock. Le tableau 6 ci-dessous précise les expertises et les responsabilités de chacun.

Tableau 6. Énumération des membres l'équipe de développement, description de leur expertise et de leurs responsabilités

Membre de l'équipe	Expertise	Responsabilités
Geneviève Bock	Socket.IO Expérience utilisateur	- Communication client lourd serveur. - Assurance qualité de l'expérience utilisateur sur le client lourd.
Antoine Daigneault-Demers	C# Interface utilisateur	- Conception du back-end du client lourd. - Conception de l'interface utilisateur du client lourd. - Assurance qualité du code du client lourd.
Mathieu Giroux-Huppé	Android Studio Expérience utilisateur	- Développement du client léger. - Assurance qualité de l'expérience utilisateur sur le client léger.
Maxime Gosselin	Java Interface utilisateur	- Conception du back-end du client léger. - Conception de l'interface utilisateur du client léger. - Assurance qualité du code du client léger.
Jérémie Huppé	Planification Socket.IO	- Gestion des livrables. - Communication du serveur. - Cohérence entre les interfaces des clients lourd et léger.
Alexandre Rault	Google Cloud Datastore Node.js	- Conception et administration de la base de données sur Google Cloud Datastore. - Développement du serveur. - Assurance qualité du code du serveur.

6. Entente contractuelle proposée

Notre équipe propose une entente contractuelle clé en main.

Ce type d'entente correspond aux critères de l'appel d'offres puisqu'il stipule que les soumissionnaires retenus devront livrer un produit final à une date précise. Également, les requis du système définissent clairement et précisément le produit à livrer. Ces derniers sont spécifiés dans le document « SRS » rédigé par notre équipe et obligatoire à la soumission. L'incertitude et la complexité des spécifications étant moindres, une assurance des coûts finaux et une garantie de livraison sont offertes grâce à un contrat clé en main.

Par ailleurs, il n'y a présentement pas d'urgence de livraison favorisant un contrat à terme. Effectivement, l'appel d'offres ne stipule qu'une livraison à effectuer, soit celle du produit final, le 8 avril 2019. Également, une mise en chantier plus rapide permise par ce type de contrat n'est pas requise.

Somme toute, un contrat de livraison clé en main est la meilleure option considérant qu'il produit la meilleure garantie de résultat et de coûts finaux, lorsque les requis sont clairs et précis. Cette entente implique que notre équipe devra livrer le produit final le 8 avril 2019. Le paiement final devra être émis à la livraison du produit final et à son acceptation par PolyApps. Elle implique également que tout changement aux spécifications après l'allocation de la soumission requerra une négociation et que les coûts pourraient être augmentés suite à des imprévus liés aux travaux à haut risque.

Le prix demandé pour le produit final est de 114 525$. Ce prix est basé sur les lots de travail présentés à la section 4 ainsi que sur les taux suivant :

Tableau 7. Taux horaire des employés selon leur rôle

Taux horaire des employés ($/h)	
Gestionnaire de projet	125
Développeur	100

Pour la rédaction du « SRS », du « Plan de projet » et de la « Liste d'exigences », le taux horaire utilisé est celui du gestionnaire de projet. Pour tous les autres lots de travail, nous avons utilisé un taux horaire d'environ 104$, correspondant à la moyenne pondérée du taux du gestionnaire de projet à 16% et d'un développeur à 84%. Ces taux horaires sont multipliés par les temps estimés pour chaque lot de travail. Leur somme constitue le prix demandé pour le produit final.

Le budget pour chaque jalon est détaillé dans le tableau ci-dessous :

Tableau 8. Nombre d'heures et coûts estimés pour chaque jalon du projet

Jalon	Nombre d'heures estimé (h)	Coûts estimés ($)
1	280	31 125
2	420	43 680
3	380	39 520