

---

**Équipe 12**

---

**Poly Paint  
Plan de projet**

**Version 2.2**

## Historique des révisions

Date	Version	Description	Auteur
2019-01-28	1.0	Rédaction initiale	Chelsy Binet, Olivier Lauzon, Alexis Loïselle, Sébastien Cadorette, Sébastien Labine et William Sévigny
2019-03-14	2.0	Correction initiale suite à la première remise	Chelsy Binet
2019-04-02	2.1	Ajustements pour la remise finale	Chelsy Binet
2019-04-07	2.2	Relecture et approbation finale	Chelsy Binet, Olivier Lauzon, Alexis Loïselle, Sébastien Cadorette, Sébastien Labine et William Sévigny

# Table des matières

<b>1. Introduction</b>	<b>4</b>
<b>2. Énoncé des travaux</b>	<b>4</b>
2.1. Solution proposée	4
2.2. Hypothèses et contraintes	4
2.3. Biens livrables du projet	4
<b>3. Gestion et suivi de l'avancement</b>	<b>5</b>
3.1. Gestion des exigences	5
3.2. Contrôle de la qualité	5
3.3. Gestion de risque	5
3.4. Gestion de configuration	7
<b>4. Échéancier du projet</b>	<b>7</b>
<b>5. Équipe de développement</b>	<b>9</b>
<b>6. Entente contractuelle proposée</b>	<b>10</b>

# Plan de projet

## 1. Introduction

Le présent document fait office de plan de projet et contient 6 sous-sections. La prochaine section, la section 2, fournit un énoncé des travaux en détaillant la solution proposée, les hypothèses et contraintes du projet ainsi que les livrables. La section 3 porte sur la gestion et le suivi de l'avancement alors que la section 4 détaille l'échéancier du projet. La section 5 présente notre équipe de développement et la section 6 est notre suggestion concernant l'entente contractuelle proposée.

## 2. Énoncé des travaux

### 2.1. Solution proposée

Nous proposons de répondre à l'appel d'offres pour implémenter Poly Paint Pro, une extension au logiciel existant Poly Paint. Pour ce faire nous proposons une adaptation ainsi qu'une amélioration aux fonctionnalités de base du produit. En effet, le logiciel à développer comporte trois parties : un client léger sur tablette IOS, un client lourd sur PC ainsi qu'un serveur. Nous ajoutons plusieurs exigences tel un mode collaboratif en réseau qui emploie une communication entre les différents clients en passant par le serveur. En effet, un système de clavardage est mis sur pied afin de permettre aux utilisateurs de communiquer entre eux. De plus, il est possible de s'enregistrer et de s'authentifier sur le système afin d'avoir accès à un profil utilisateur personnalisé. Dans le même ordre d'idée, une galerie d'images privées est disponible sur le profil de chaque utilisateur ainsi qu'une galerie d'images publiques visibles pas tous les utilisateurs. En ce qui concerne l'édition du canevas, nous proposons l'ajout de plusieurs fonctionnalités telles que *Sélectionner tout*, *Inverser les couleurs*, *Inverser la sélection* ainsi qu'un *Transformer tout* qui seront accessibles dans une barre de tâche. Une édition collaborative de canevas sera également ajoutée. En effet, l'édition de base collaborative ainsi que l'édition de formes en mode collaboratif seront une option. Un tutoriel non interactif sera présent pour le mode d'édition et une connexion à Facebook sera proposée.

### 2.2. Hypothèses et contraintes

Le plan de projet repose sur l'hypothèse selon laquelle le projet se déroule tel que prévu, c'est-à-dire autant au niveau des compétences de chacun des membres de l'équipe qu'au temps consacré au projet par ceux-ci. Nous posons l'hypothèse qu'aucun problème majeur ne survienne et que l'équipe reste inchangée durant toute la durée du projet. Notre estimation de temps repose sur une charge de travail réparti uniformément sur l'ensemble des semaines, sauf pour les semaines où nous prévoyons des remises de travaux ou des examens dans d'autres cours. Les contraintes de ce projet se situent autant au niveau des ressources humaines, de l'équipement que de l'échéancier. En effet, nous sommes contraints à une équipe de 6 développeurs, nous devons donc ajuster l'effort en conséquence. De plus, trois développeurs possèdent des macs, ce qui permet de bien répartir les tâches des clients lourd et léger au sein de l'équipe. La plus grande contrainte se situe au niveau du temps alloué au projet. En effet, puisque le projet est réalisé dans un cadre pédagogique, il faut garder en tête que les développeurs sont inscrits à d'autres cours et ont beaucoup d'autres implications et doivent donc répartir leur temps judicieusement entre le projet et leurs autres obligations.

### 2.3. Biens livrables du projet

Dans un premier temps, nous remettons une première soumission sous forme de réponse à l'appel d'offre. Cette première version contiendra plusieurs artéfacts, dont ce plan de projet. Il contiendra également un document de spécification des exigences logicielles (SRS), une liste des exigences, un document d'architecture logicielle ainsi qu'un document détaillant le protocole de communication. En ajout à cela, nous remettons un prototype fonctionnel démontrant le protocole de communication du client lourd ainsi que du client léger. Pour prouver notre capacité à réaliser le système demandé et pour démontrer notre intérêt envers le projet offert, nous ajoutons en avance à ce prototype plusieurs fonctionnalités qui seront présentes sur le produit final. La date d'échéance de l'appel d'offre est

le **8 février 2019 à 23h59**. C'est donc cette journée que sera livrée cette première soumission.

Dans un deuxième temps, suite à l'acceptation de la proposition à l'appel d'offre, les livrables du produit final seront poursuivis. Pour cette livraison finale, des artéfacts seront également nécessaires. Tous les artéfacts remis lors de la réponse à l'appel d'offres seront retravaillés et peaufinés. À cela nous ajoutons un document de plan de tests ainsi qu'un document détaillant les résultats de ces tests. Nous remettrons également le code source et les exécutables du produit final pour les clients lourd et léger ainsi que pour le serveur. Le produit final sera conforme aux exigences et d'excellente qualité. La date de livraison de la totalité du projet est le **8 avril 2019**.

### **3. Gestion et suivi de l'avancement**

#### **3.1. Gestion des exigences**

D'abord, chaque exigence est traduite sous forme de demande sur Redmine et est assignée à un développeur. On lui attribue une date de début et de fin pour ainsi pouvoir vérifier son état d'avancement et pouvoir prévoir et corriger les retards d'échéancier. Dans le cas où un retard survient ou qu'un changement doit être apporté au calendrier, une rencontre d'équipe est de mise pour discuter de la meilleure façon de surmonter le changement. Souvent, on attribuera à cette tâche un développeur supplémentaire pour pallier au retard encouru. Si possible, nous tentons de redoubler d'efforts dans la semaine courante pour régler le problème, afin que celui-ci ne vienne pas influencer le travail des semaines subséquentes.

#### **3.2. Contrôle de la qualité**

En ce qui concerne le contrôle de la qualité du code et des exécutables, chaque fonctionnalité codée par un développeur est vérifiée attentivement et testée par un autre membre de l'équipe. En effet, les mardis nous nous rencontrons toute l'équipe et nous assurons que le code qui a été fait pendant le reste de la semaine est conforme aux exigences, respecte les normes de présentation fixée et est sans faille. Nous nous assurons qu'au moins deux membres de l'équipe aient vu et approuvé chaque fonctionnalité ce qui garantit un code performant et de qualité. Si une partie du travail ne respecte pas les normes que nous nous sommes fixées ou est en deçà du seuil de qualité exigé, cette partie ne sera pas fusionnée tout de suite avec le reste du travail approuvé et devra être retravaillée. Pour entreprendre une action corrective, une discussion d'équipe est faite pour rectifier la situation et ajuster les effectifs nécessaires à la bonne complétion du travail. Une nouvelle branche est créée sur git pour faire la correction et lorsque la fonctionnalité respecte les normes fixées, une fusion est faite avec le reste du code.

En ce qui concerne le contrôle de la qualité des artéfacts, tous les documents produits sont faits de manière conjointe avec toute l'équipe. Les documents sont produits sur google document afin que chacun puisse travailler en même temps et qu'une correction en temps réel puisse se faire. Chaque partie du travail est relue plusieurs fois et sans cesse améliorée. Nous utilisons le logiciel Antidote pour déceler les erreurs de grammaire et d'orthographe. Une fois un document considéré comme terminé, tous les membres de l'équipe se doivent d'effectuer une relecture finale et approuver le document. Nous nous assurons ainsi d'une uniformité du travail et d'une excellente qualité de celui-ci. Pour entreprendre une action corrective, si quelqu'un a un changement à apporter au document, celui-ci la fait sous forme de commentaire sur Google doc et une autre personne doit approuver la modification.

#### **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 (métriques) du système pouvant être compromis.
- Stratégie de gestion : mesures à prendre afin de gérer le risque.

R1 - Panne de la base de données				
Ampleur	Description	Impact	Facteurs	Stratégie de gestion
2	La base de données contenant les informations de l’application tombe en panne temporairement. Les utilisateurs ne peuvent donc pas accéder au service.	E	Performance et accessibilité à l’application	Mitiger. Tester très souvent la base de donnée afin de réduire au minimum la probabilité que le risque survienne.

R2 - Communication défectueuse entre les clients				
Ampleur	Description	Impact	Facteurs	Stratégie de gestion
3	La communication entre deux clients différents échoue.	E	Performance de l’application	Contourner. Revoir le protocole de communication

R3 - Attaque par déni de service (DDoS)				
Ampleur	Description	Impact	Facteurs	Stratégie de gestion
3	Indisponibilité du service suite à l’inondation du réseau. L’accès à l’application devient donc impossible.	E	Performance et accessibilité à l’application	Éviter. S’assurer qu’une inondation du réseau soit impossible ou très peu probable.

R4 - Bogue non détecté				
Ampleur	Description	Impact	Facteurs	Stratégie de gestion
3	Un problème de code non détecté survient après l’acceptation de la fonctionnalité. Par exemple une différence entre le client lourd et léger en collaboration	M	Qualité du code	Mitiger. Tester toutes les fonctionnalités avant de les accepter afin de minimiser la probabilité d’un bogue.

R5 - Les bibliothèques choisies ne fonctionnent pas comme prévu.				
--	--	--	--	--

Ampleur	Description	Impact	Facteurs	Stratégie de gestion
5	La librairie ne fonctionne pas aussi bien que le montre la documentation. Du temps supplémentaire est donc nécessaire.	E	Performance de l'application	Contourner. Changer de librairie en cours de projet.

R6 - Problème lors de l'intégration des différentes composantes				
Ampleur	Description	Impact	Facteurs	Stratégie de gestion
5	L'intégration des composantes rend des fonctionnalités inutilisables.	E	Qualité du code	Mitiger. Faire des itérations plus courtes pour intégrer de plus petites composantes et ainsi minimiser l'impact du risque.

### 3.4. Gestion de configuration

Lorsqu'un problème est décelé dans le fonctionnement du code ou qu'un changement survient en cours de production, nous utilisons les outils Redmine et Git. Pour disposer d'un problème, nous créons d'abord une nouvelle demande sur Redmine et lui attribuons le tracker d'anomalie. Par la suite, une nouvelle branche nommée hotfix/nom-de-la-fonctionnalité est créée sur le répertoire et un développeur est affecté à la résolution de cette tâche. Une fois le changement fait et que la fonctionnalité a été testée et approuvée par au moins deux développeurs sur cette branche, une fusion est faite avec la branche dev. Si le tout est entièrement fonctionnel, la demande dans Redmine peut ensuite être fermée et la branche supprimée.

En ce qui concerne la façon dont sont nommées et numérotés les artefacts, nous nous sommes donné des conventions de nommage afin de bien nous comprendre et de rester constant dans le temps. Par exemple, lorsque nous créons des demandes sur Redmine, chaque sujet commence par un code de trois lettres suivi du nom de la fonctionnalité par exemple "AAA - Fonctionnalité". Les demandes parents débutent par CLO, CLE ou DOC pour représenter le client lourd, le client léger ou les documents respectivement. Ensuite, pour les demandes enfants, celles-ci débutent par soit FRT pour le frontend ou par BCK pour le backend. Pour ce qui est des artefacts, la première version de chaque document porte le numéro 1.0 et sera incrémentée de 0.1 à chaque itération importante.

## 4. Échéancier du projet

	Lot de travail	Effort estimé (heures-personne)	Date de début et de fin
	Interface du prototype du client lourd	40	8 janvier 2019 - 5 février 2019
	Envoi et réception de donnée pour le prototype du client lourd	40	15 janvier 2019 - 5 février 2019
	Interface du prototype du client léger	40	8 janvier 2019 - 5 février 2019

Réponse à l'appel d'offre	Envoi et réception de donnée pour le prototype du client léger	40	15 janvier 2019 - 5 février 2019
	Écriture du SRS pour la première remise en LOG3000	30	14 janvier 2019 - 25 janvier 2019
	Ajustement du SRS suite à la correction	15	4 février 2019 - 7 février 2019
	Écriture du plan de projet	15	29 janvier 2019 - 7 février 2019
	Écriture du protocole de communication	15	29 janvier 2019 - 7 février 2019
	Écriture du document d'architecture logicielle	15	29 janvier 2019 - 7 février 2019
Produit final	Gestion et amélioration de la base de données	20	11 février 2019 - 18 février 2019
	Amélioration du clavardage en mode intégré et fenêtré	20	11 février 2019 - 18 février 2019
	Amélioration des interfaces visuelles	30	11 février 2019 - 18 février 2019
	Ajout des éléments UML	50	11 février 2019 - 25 février 2019
	Gestion de l'authentification d'un nouvel utilisateur	30	18 février 2019 - 25 février 2019
	Création d'un profil utilisateur	30	18 février 2019 - 25 février 2019
	Création de l'interface pour la galerie d'image	30	18 février 2019 - 4 mars 2019
	Intégration de la galerie d'image	30	18 février 2019 - 4 mars 2019
	Gestion des images (sauvegarde, chargement, accessibilité)	50	25 février 2019 - 11 mars 2019
	Ajout des fonctionnalités de bases (Select all et Transform all)	35	25 février 2019 - 11 mars 2019
	Ajout des fonctionnalités de bases ( Invert colors et Invert selection)	35	25 février 2019 - 11 mars 2019
	Création de l'interface pour les canaux de discussion	30	25 février 2019 - 11 mars 2019



Intégration Clavardage avec plusieurs canaux de discussion	40	25 février 2019 - 11 mars 2019
Édition de base collaborative	50	4 mars 2019 - 18 mars 2019
Édition de forme collaborative	50	11 mars 2019 - 25 mars 2019
création d'un tutoriel	30	11 mars 2019 - 18 mars 2019
Effets visuels et sonore	30	11 mars 2019 - 18 mars 2019
Intégration de gestes	30	18 mars 2019 - 25 mars 2019
Ajouts des exigences souhaitables (facebook login et barre de tâche)	40	18 mars 2019 - 25 mars 2019
Mise à jour des artéfacts suite à la correction	50	25 mars 2019 - 7 avril 2019
Écriture du document de plan de test logiciel	30	25 mars 2019 - 7 avril 2019
Écriture du document de résultat de test	30	25 mars 2019 - 7 avril 2019
Préparation de la présentation orale	40	1 avril 2019 - 8 avril 2019

Jalons	Effort estimé total (heures-personne)	Date de début et de fin
réponse à l'appel d'offre	250	8 janvier 2019 - 8 février 2019
produit final	810	8 février 2019 - 8 avril 2019
projet total	1015	8 janvier 2019 - 8 avril 2019

Au terme de 45h/crédit/personne, il est attendu de l'équipe de faire un total de 1080 heures. Notre estimation de 1015 heures a donc du sens considérant que l'échéancier contient uniquement les principaux lots de travail. Nous ajoutons à cela des tâches minimales de quelques heures ainsi qu'une marge d'erreur. De plus, en termes de temps, la réponse à l'appel d'offre prend 1 mois sur 3, donc environ le tiers du projet. Les heures sont donc représentatives de l'effort mis jusqu'à présent et du travail qu'il reste à faire.

## 5. Équipe de développement

Chelsy Binet

- Étudiante en 3e année à Polytechnique de Montréal.
- Membre de trois exécutifs à Polytechnique : Codirectrice pour les computer science games, trésorière pour les jeux de génie et vp aux installations étudiantes pour le comité des étudiants en génie logiciel.
- Maîtrise de plusieurs langages tels que C++, JavaScript et Java.
- Parmi les responsables du développement du client lourd.

#### Sébastien Cadorette

- Étudiant en 3e année à Polytechnique Montréal.
- Membre d'un exécutif à Polytechnique : Trésorier de PolyPhoto.
- Grand intérêt pour le Web & développement mobile sous iOS.
- Parmi les responsables du développement de l'application mobile iOS.

#### Sébastien Labine

- Étudiant en 3e année à Polytechnique Montréal.
- Développeur .NET Core, C#, iOS.
- Entrepreneur technologique.
- Parmi les responsable du développement du serveur .NET ainsi que du client lourd.
- Responsable du déploiement.

#### Olivier Lauzon

- Étudiant en génie logiciel en 3e année à Polytechnique Montréal.
- Bonnes connaissances en programmation orientée objet.
- Spécialisé en C++ et en C#.
- Travaille plus particulièrement sur le développement du client lourd et du serveur.

#### Alexis Loisel

- Étudiant en génie logiciel en 3e année à Polytechnique Montréal.
- Expérience de 5 ans en C#.
- Maîtrise de plusieurs langages tels que C++, JavaScript, Ruby et Java.
- Parmi les responsables du développement du serveur .Net Core et du client lourd.

#### William Sévigny

- Étudiant en génie logiciel en 3e année à Polytechnique Montréal.
- Bonne maîtrise de l'architecture client-serveur et des patrons de conception..
- Enthousiaste du UI / UX.
- Travaille sur le développement du client léger en iOS.

## 6. Entente contractuelle proposée

Nous proposons une entente contractuelle de type clé en main à prix ferme. Même si une entente de partage des économies avait été possible avec la PME PolyApps, nous pensons qu'il ne s'agit pas là de la meilleure option. En effet, nous prenons sur nous tout le risque de dépassement des coûts et sommes prêts à fournir tout le temps nécessaire pour fournir un produit final qui sera extrêmement performant et peaufiné. Le projet sera livré avec toutes les exigences essentielles et souhaitables détaillées dans la liste d'exigences. Le temps total estimé pour le projet est de 1080 heures réparties sur l'équipe composée de 5 développeurs travaillant à un taux de 100\$ par heure et d'un gestionnaire de projet travaillant à 125\$ par heure. Le prix total, demandé lors de la livraison finale de l'application, est donc de 112 500\$.