Équipe 5

Analyse de traces du démarrage de Chrome
Plan de développement logiciel

Version 1.4

Historique des révisions

Date	Version	Description	Auteur
2016-01-22	1.0	Début d'écriture du document	Maxime, Olivier
2016-01-28	1.1	Suite de l'écriture du document	Jonathan Rochon
2016-01-29	1.2	Correction du document	Jonathan Rochon
2016-02-11	1.3	Mise à jour des objectifs d'itération	Jonathan Rochon
2016-04-08	1.4	Mise à jour du document suite à l'évaluation de mi-session	Alexandre Thibault

Table des matières

Plan de développement logiciel

1. Introduction

Le présent document vise à détailler les étapes du développement logiciel. Il est divisé en trois grandes parties qui seront détaillées de manière consécutive: vue d'ensemble du projet, organisation du projet, processus de gestion du projet.

2. Vue d'ensemble du projet

2.1 But du projet, portée et objectifs

Le but du projet est de développer et d'améliorer des outils permettant de faciliter l'optimisation du démarrage de Google Chrome. À cette fin, nous développerons un outil permettant de convertir une trace ETW dans un format qui peut être lu par chrome://tracing/. Une fois cet outil développé, nous ajouterons des fonctionnalités à chrome://tracing/ permettant de présenter à l'utilisateur les nouvelles informations fournies par la trace ETW.

2.2 Hypothèses et contraintes

Nous avons pris pour hypothèse que chaque membre travaillera un total d'environ 270 heures au cours du projet pour un total d'environ 1080 heures * personne. Cette hypothèse est donc utilisée pour déterminer les différentes dates de livraisons d'artefact à l'exception de celles prédéfinies comme jalon dans le projet. Ces dates sont le 12 janvier pour la remise initiale de la description du processus, le 22 janvier pour la remise initiale du document d'exigences, le 29 janvier pour la remise initiale du document actuel (Plan de développement logiciel), le 12 février pour la remise de toute la documentation en plus d'un prototype, le 31 mars pour la remise d'une version bêta du produit, et le 13 avril pour la remise finale de toute la documentation et du produit.

2.3 Biens livrables du projet

Les biens livrables de ce projet sont de deux types: les artéfacts et le logiciel.
Les artéfacts et leurs dates de livraison sont détaillés dans le tableau ci-dessous:

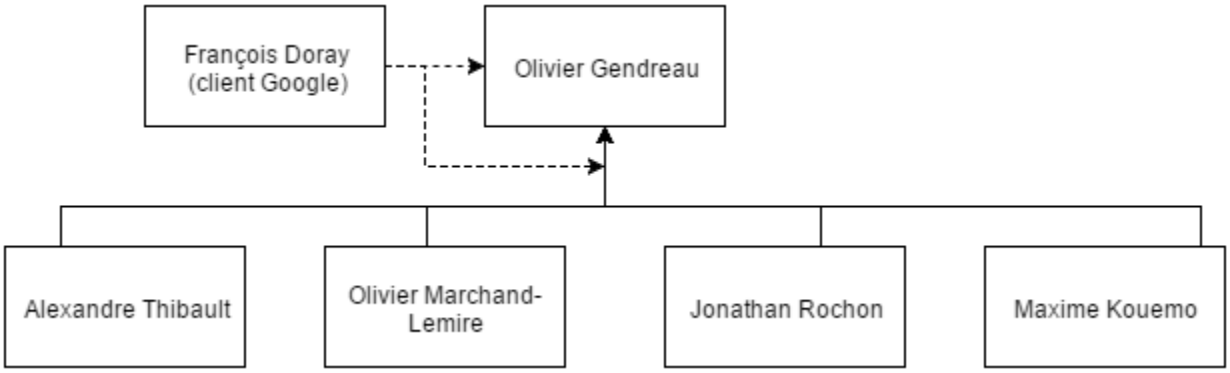
Artéfact	Date prévue de livraison
SRS initial	22 Janvier 2016
Plan de développement logiciel initial	29 Janvier 2016
SRS corrigé	12 Février 2016
Plan de développement logiciel corrigé	12 Février 2016
Plan de tests logiciels	12 Février 2016
Document d'architecture logicielle	12 Février 2016
Processus	12 Février 2016
Prototype	12 Février 2016

Logiciel (version bêta)	31 Mars 2016
Résultats des tests	13 Avril 2016
Produit final (logiciel plus documentation)	13 Avril 2016

3. Organisation du projet

3.1 Structure d'organisation

La structure de l'organisation est composée des quatre membres de l'équipe, sous la direction d'Olivier Gendreau. Le client, François Doray, interagit avec M. Gendreau et avec les membres de l'équipe de développement.



3.2 Interfaces externes

Nous avons deux contacts externes, Polytechnique Montréal et Google. Pour Polytechnique Montréal, les interactions consistent surtout en l'évaluation scolaire du projet et l'organisation haut niveau du projet avec quelques livrables comportant des échéanciers spécifiques. Polytechnique Montréal nous offre aussi un encadrement au cours du projet afin de s'assurer que celui-ci se déroule aussi agréablement que possible. Comme il s'agit du projet de 4e année de génie logiciel, notre contact interne est Olivier Gendreau. Ensuite, le second groupe avec lequel nous sommes en contact est notre client, Google, qui nous a offert le projet auquel nous participons actuellement. Pour ce groupe notre contact externe est François Doray, ingénieur chez Google, qui communique principalement avec Jonathan Rochon notre contact interne.

3.3 Responsabilités

Les responsabilités des membres de l'équipe sont réparties comme suit: Alexandre Thibault est responsable du processus de développement; Jonathan Rochon est responsable de l'assurance qualité et de la conformité entre le produit final et les attentes du client; Maxime Kouemo est responsable technique; Olivier Marchand Lemire est responsable de la configuration et de la planification. Quelle que soit la responsabilité, certaines décisions (lorsqu'elles sont majeures) sont prises à l'issue d'une réunion.

4. Processus de gestion

4.1 Plan de projet

4.1.1 Planification des phases

Ce projet est divisé en quatre phases: les phases d'initialisation, d'élaboration, de construction et de transition. Les requis initiaux du projet seront définis lors de la phase d'initialisation et seront entrés dans le SRS. Il sera possible de modifier celui-ci dans les phases subséquentes sur autorisation du client. Le plan de développement logiciel sera également produit au cours de cette phase. Cette phase sera complétée durant la première itération. La phase d'élaboration est la principale phase de planification initiale de l'architecture du projet. La problématique sera alors étudiée pour produire un document d'architecture, les exigences seront précisées si nécessaire, les plans de tests seront produits en fonction de l'architecture prévue. Un prototype sera également fait pour présenter au client. Cette phase sera également complétée lors de la première itération. La phase de construction est la principale phase d'implémentation du projet. Au cours de celle-ci, les fonctionnalités prévues seront ajoutées au système. Cette phase se déroulera durant les itérations 2 à 5. Finalement, la phase de transition se déroulera après la remise du bêta et elle consistera à améliorer le produit pour s'assurer qu'il réponde aux exigences de qualité du client.

4.1.2 Objectifs d'itération

Le projet sera subdivisé en «sprint» de deux semaines, donc en six itérations. Chacune de ces itérations inclut une période de tests.

Du 25 janvier au 8 février:

La première itération consistera à débuter la production de la documentation, à établir l'architecture du système et à produire l'outil de conversion de base des traces «ETW» en format «JSON». La conversion de base consiste à seulement effectuer la conversion des paramètres lisibles actuellement par l'outil de traçage de chrome. Aucune nouvelle fonctionnalité n'a besoin d'être ajoutée. Le temps et l'effort pour effectuer pleinement cette itération semblent raisonnables.

Du 9 février au 22 février:

Lors de cette seconde itération, nous commencerons par terminer toutes documentations restantes. Ensuite, nous ajouterons les fonctionnalités liées au «FlameGraph» à l'outil de conversion. De plus, nous commencerons la modification de l'interface graphique de l'outil de traçage de Chrome pour y ajouter les paramètres liés au «FlameGraph». Ces deux activités seront séparées en binôme pour cette période de temps. Le temps et l'effort pour effectuer pleinement cette itération semblent au8ssi raisonnables.

Du 23 février au 7 mars:

Lors de cette période, un binôme travaillera à l'amélioration de l'outil de conversion en lui ajoutant les fonctionnalités en lien avec la lecture de disque et de fichiers. Le second binôme ajoutera les fonctionnalités liées au «CPU» à l'outil de conversion. De plus, nous continuerons la modification de l'interface graphique de l'outil de traçage de Chrome pour y ajouter les paramètres liés au «CPU». Le temps et l'effort pour effectuer pleinement cette itération semblent aussi raisonnables.

Du 8 mars au 21 mars:

Lors de ces deux semaines, un binôme commencera le travail sur l'exportation des indicateurs de flux à l'intérieur de Chrome vers «ETW». Pendant ce temps, le second binôme travaillera pendant ce temps à l'écriture du script de génération de trace.

Du 22 mars au 31 mars:

Cette période sera allouée à la recherche de traces réduites de «ETW» à l'aide de «PerfIsight» filtrer ETW/perf et la finition des fonctionnalités qui auraient pu prendre plus de temps que prévues. De plus, nous travaillerons sur la précision des fonctions des traces. Nous n'avons aucune idée de l'ampleur du temps nécessaire à la complétion de ces travaux puisque nous ne comprenons pas, à ce jour, complètement leur fonctionnement. Il est donc probable que ceux-ci s'étirent sur une période dépassant le 1 avril.

Du 1 avril au 13 avril (post beta):

Cette période sera utilisée pour améliorer le produit en fonction des commentaires du client précédemment rencontré. De plus, cette période sera encore utilisée à des fins de finitions de fonctionnalités non peaufinées.

4.2 Suivi de projet et contrôle

4.2.1 Gestion des exigences

La gestion des exigences se fera principalement en conservant un journal des changements apportés au cours de nos échanges par courriel avec le client ainsi que les comptes-rendus des rencontres avec celui-ci. De plus, afin de pouvoir faire des comparaisons nous garderons les différentes versions du SRS.

4.2.2 Contrôle de la qualité

Durant le projet, les modules seront testés de façon unitaire, au fur et à mesure qu'ils seront développés. De plus, les tests fonctionnels du produit seront effectués à chaque fois es modules et/ou fonctionnalités seront fusionnés au produit principal.

Chaque membre de l'équipe de développement va s'imprégner du SRS, et toute question d'approfondissement sera soumise au client.

Lorsqu'un bogue sera découvert, nous évaluerons son importance. Suivant celle-ci nous déciderons des mesures correctives à prendre. Pour un bogue mineur, le membre de l'équipe qui a travaillé sur la fonctionnalité pourra le réparer immédiatement. Pour un bogue majeur, l'équipe entière pourra s'y mettre afin de le régler au plus vite. Tout dépendra donc de l'importance du bogue, laissée à la discrétion de l'équipe de développement.

L'établissement d'un calendrier interne, avec des livrables (artéfacts et prototypes), que le client pourra tester, garantira que le produit final sera conforme aux besoins du client.

4.2.3 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 : ne 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.

Ampleur	Description	Impact	Facteurs	Stratégie de gestion
8	Exigences incorrectes et/ou incomplètes	C	Justesse	Mitiger le risque en validant les exigences avec le client et en les reconfirmant à chaque étape du projet.
6	Difficulté de compréhension du code existant.	E	Complétude, Maintenabilité	Mitiger le risque en communiquant avec le client lorsqu'un problème de compréhension est rencontré.
5	Difficulté à faire approuver le code par le client (Google)	M	Complétude	Transférer le risque. Le représentant du client peut terminer le processus d'approbation une fois le projet terminé.

4.2.4 Gestion de configuration

La gestion de fichiers sera effectuée sur GitHub sur trois niveaux différents. Le premier niveau est représenté par un dépôt dans lequel les quatre membres de l'équipe travailleront. Chacun travaillera sur ses propres branches respectives qui seront fusionnées dans une branche de développement lorsqu'une fonctionnalité est complétée. Lorsque le contenu de la branche de développement satisfera un des requis, celui-ci sera copié sur la branche maître du même dépôt.

Dès qu'un requis terminé est déposé sur la branche maître, un «pull-request» sera effectué sur un second dépôt représentant notre organisation. Celui-ci sera utilisé pour faire le lien entre l'équipe et le client. Ce dernier fera une vérification du code avant d'accepter le «pull-request».

Lorsqu'un requis sera terminé et accepté par le client, un second «pull-request» sera effectué pour ajouter nos ajouts et modifications au dépôt officiel du client.