

Plan de développement 0.1

3 janvier 2015

Auteur(s): Pierre-Luc BLOT, Kheireddine BERKANE
Relecteur(s): Nasser ADJIBI

Version	Date	Changelog
0.1	23/12/2014	Version initiale.

Table des matières

1	Contexte du projet	2
1.1	Origine du projet	2
1.2	Contexte de développement	2
1.3	Les principaux acteurs	3
1.4	Objectifs poursuivis	3
1.5	Documents de référence	3
2	Méthodologie de développement	3
2.1	Méthode de développement	3
2.2	Découpage général en cycles	5
2.3	Réunion de début d'itération	7
2.4	Développement et corrections	7
2.5	Livraison du produit intermédiaire	7
2.6	Corrections	8
2.7	Réunion de fin d'itération	8
2.8	Suivi hebdomadaire	8
2.9	Versionning	8
2.10	Documents	8
3	Organisation et responsabilité	9
3.1	Attribution des rôles au sein de l'équipe de développement	9
3.2	Chef de projet	9
3.3	Chargé client	9
3.4	Testeur	9
3.5	Responsable technique et développement	10
3.6	Développeur, programmeur	10
3.7	Manager	10
3.8	Client	10
3.9	Professeur référent	10
4	Procédés de gestion	11
4.1	Gestion de la documentation	11
4.1.1	Spécification Technique du besoin (STB)	11
4.1.2	Description Architecture Logicielle	11
4.1.3	Cahier de recettes	12
4.1.4	Analyse des Risques (AdR)	12
4.1.5	Plan de Développement (PdD)	13
4.1.6	Manuel d'utilisation	13
4.2	Gestion des configurations	13
5	Revue et points clefs	14

1 Contexte du projet

1.1 Origine du projet

L'objectif à travers le choix du langage **kawa** qui est similaire à **java** (un langage de haut niveau d'abstraction) c'est de permettre une facilité ainsi qu'une rapidité de développement par rapport à d'autres langages de programmation tel que le «C». Le code Java est compilé dans un langage intermédiaire Bytecode et est exécuté dans un environnement d'exécution JVM (Java Virtual Machine), dans le cadre de notre projet nous allons montrer que l'on peut compiler un code similaire à java (kawa) en code natif sans passer par une machine virtuelle (MV).

1.2 Contexte de développement

Ce projet s'inscrit dans le cadre de la formation Master 1 Génie de l'Informatique Logicielle dispensée à l'UFR des Sciences et Techniques de Saint Etienne du Rouvray. Il nous a été proposé par nos professeurs référents, notamment Mr Nicart. Pour cela, nous disposons d'environ 11 semaines pour réaliser tous les documents techniques, et d'un peu plus de 15 semaines pour le développement.

Ce projet nous été proposé le 17/10/2014. Notre équipe de développement est composée de sept membres :

- ADJIBI Nasser
- PETRE Alexandre
- TASSERIE Majid
- BLOT Pierre-Luc
- BERKANE Kheireddine
- IDRISOU Hamzath
- AHOuate Abdellatif

Dates	Sujets
17/10/14	Remise du sujet
23/10/14	Première réunion client
03/11/14	Deuxième réunion client
14/11/14	Troisième réunion client
20/11/14	Quatrième réunion client
26/11/14	Cinquième réunion client
05/12/14	Sixième réunion client
22/12/14	Dernière réunion client
03/01/15	Remise des documents
19/01/15	Revue de lancement du projet

Et les dates clés du projet pendant la phase de développement :

Dates	Sujets
06/03/15	Premier livrable
17/04/15	Deuxième livrable
15/05/15	Troisième livrable

1.3 Les principaux acteurs

Rôles	Acteurs
Client	Mr Nicart
Professeur référent	Mr Nicart
Equipe de développement	ADJIBI Nasser ; PETRE Alexandre ; TASSERIE Majid ; BLOT Pierre-Luc ; BERKANE Kheireddine ; IDRISOU SOULER Hamzath ; AHOATE Abdellatif

1.4 Objectifs poursuivis

L'objectif du projet kawa est de livrer un compilateur fonctionnant avec la technologie LLVM au client au mois de Mai 2015. Ce projet va permettre à l'ensemble de notre groupe de travailler ensemble durant ces semaines et de développer ainsi nos compétences dans les technologies C++ et LLVM ainsi que ELF pour UNIX.

1.5 Documents de référence

Nom du document	Description
STB.pdf	Ce document regroupe les attentes du client et établit une référence.
ADR.pdf	Ce document rescence les difficultés qui pourraient ralentir la bonne avancée du projet.
DAL.pdf	Ce document traduit les solutoins techniques conçue pour répondre aux exigences définies dans le document STB.pdf.
CDR.pdf	Ce document définit les moyens et précédés mis en oeuvre pour assurer la recette du logiciel développé.

2 Méthodologie de développement

2.1 Méthode de développement

Le projet va être développé en s'appuyant sur les méthodes agiles. Nous avons décidé de nous appuyer notamment sur deux de ces méthodes : la méthode DSDM (Dynamic Systems Development Method) ainsi que la méthode XP (eXtrem Programming). En effet, nous allons produire différentes versions du logiciel. Lors de chaque version, de nouvelles fonctionnalités seront ajouter afin d'arriver à la version finale comprenant toutes les fonctionnalités demandées par le client. Voici les points sur lesquels vont reposer notre projet :

- La communication : Ce projet est un travail d'équipe.
- La simplicité : Une solution doit uniquement répondre au problème posé.
- Le feedback : L'avancement sera évalué sur des éléments concrets régulièrement. Des versions préliminaires seront livrées au client pour obtenir leurs retours (l'avis du client est toujours pertinent). De plus, le client fait partie intégrante de l'équipe, son retour sera bénéfique et essentiel. Il est impliqué tout au long du cycle de développement. Tout ceci permet une validation par étape. Ceci permet enfin une visibilité du résultat.
- Le développement en binôme : Ce développement en binôme permet tout d'abord une meilleure répartition des tâches. De plus, ceci permet également une relecture du code au fur et à mesure de l'avancement. Ainsi, la cohésion de l'équipe est améliorée, et la connaissance de l'application est mieux répartie.

- L'intégration continue.
- Tests continus : Ils permettent de garantir le bon fonctionnement de l'application à chaque étape de développement.

2.2 Découpage général en cycles

N°	Nom_tâche	Durée	Date_début	date_fin	Prédécesseurs	Nb ressources
1	Compilation kawa	89 jours ?	20/01/2015 09 :00	22/05/2015 18 :00		
2	Application mode monolithique	39 jours	20/01/2015 09 :00	13/03/2015 18 :00		
3	Recherche sur LLVM	9 jours	20/01/2015 09 :00	30/01/2015 18 :00		7 personnes
4	Analyse lexicale (Lexer)	5 jours	02/02/2015 09 :00	06/02/2015 18 :00	3	1 personne
5	Analyse syntaxique (Parser)	15 jours	02/02/2015 09 :00	20/02/2015 18 :00	3	3 personnes
6	Analyse sémantique 1	5 jours	02/02/2015 09 :00	06/02/2015 18 :00	3	3 personnes
7	Analyse sémantique 2	10 jours	09/02/2015 09 :00	20/02/2015 18 :00	6	4 personnes
8	Tests	5 jours	23/02/2015 09 :00	27/02/2015 18 :00	7	2 personnes
9	Génération de IR	5 jours	23/02/2015 09 :00	27/02/2015 18 :00	7	5 personnes
10	Back-end	5 jours	02/03/2015 09 :00	06/03/2015 18 :00	9	7 personnes
11	Retour client et correction	5 jours	09/03/2015 09 :00	13/03/2015 18 :00	10	7 personnes
12	Application mode partagé	50 jours	16/03/2015 09 :00	22/05/2015 18 :00		
13	Analyse syntaxique (Parser)	10 jours	16/03/2015 09 :00	27/03/2015 18 :00	11	3 personnes
14	Analyse sémantique	15 jours	16/03/2015 09 :00	03/04/2015 18 :00	11	4 personnes
15	Génération de IR	5 jours	06/04/2015 09 :00	10/04/2015 18 :00	14	5 personnes
16	Tests	5 jours	06/04/2015 09 :00	10/04/2015 18 :00	14	2 personnes
17	Back-end	5 jours	13/04/2015 09 :00	17/04/2015 18 :00	15	7 personnes
18	Retour client et correction	5 jours	20/04/2015 09 :00	24/04/2015 18 :00	17	7 personnes
19	Bibliothèque partagée	20 jours	27/04/2015 09 :00	22/05/2015 18 :00		
20	Sémantique de bibliothèque partagée	5 jours	27/04/2015 09 :00	01/05/2015 18 :00	18	7 personnes
21	Génération de IR	5 jours	04/05/2015 09 :00	08/05/2015 18 :00	20	5 personnes
22	Tests	5 jours	04/05/2015 09 :00	08/05/2015 18 :00	20	2 personnes
23	Back-end	5 jours	11/05/2015 09 :00	15/05/2015 18 :00	21	7 personnes
24	Retour client et correction	5 jours	18/05/2015 09 :00	22/05/2015 18 :00	23	7 personnes
26	Livrables	51 jours ?	06/03/2015 09 :00	17/04/2015 18 :00		
27	Livrable 1	1 jour ?	06/03/2015 09 :00	06/03/2015 18 :00		
28	Livrable 2	1 jour ?	17/04/2015 09 :00	17/04/2015 18 :00		
29	Livrable 3		15/05/2015 09 :00	15/05/2015 18 :00		

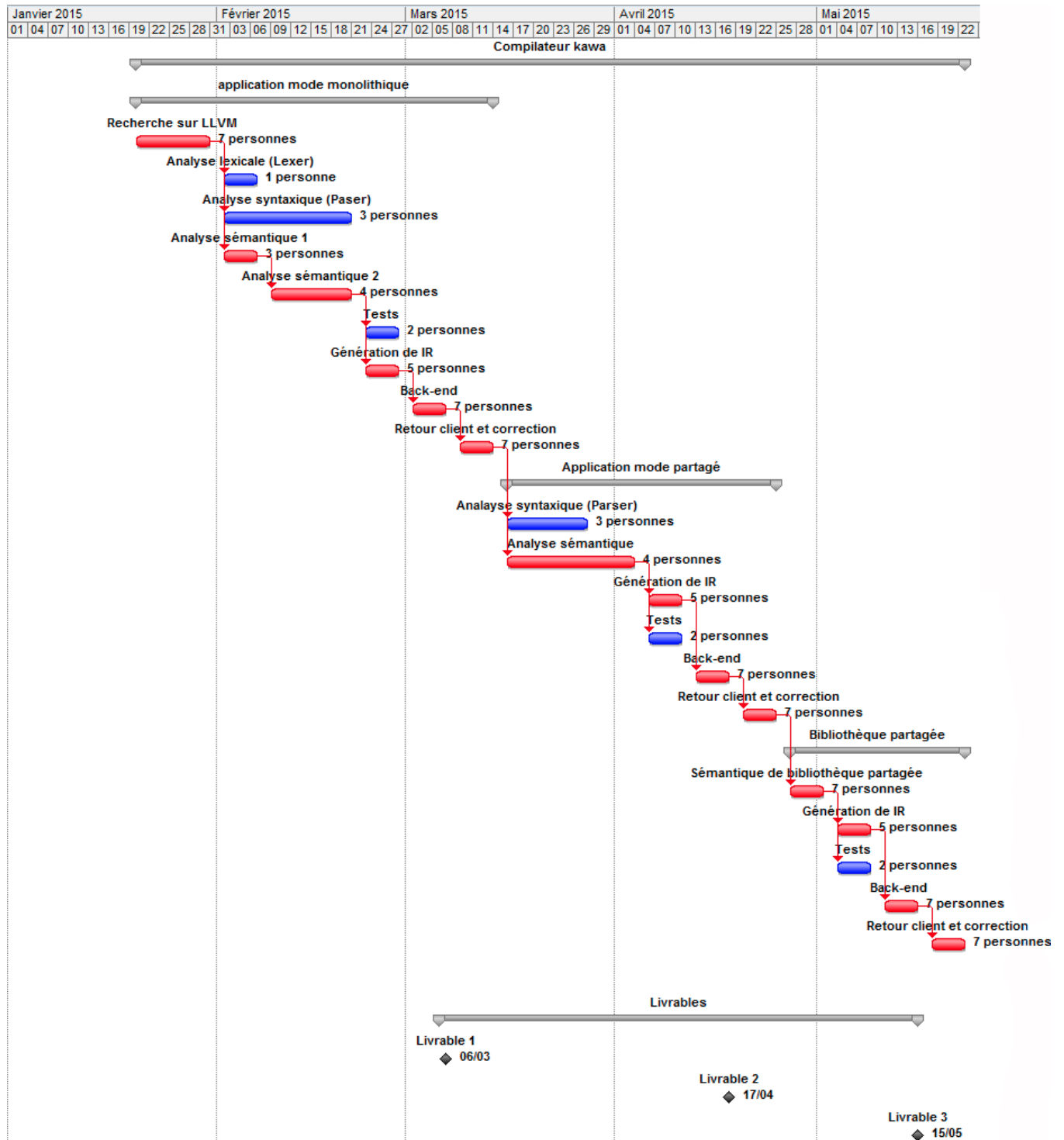


FIGURE 1 – Decoupage général en cycles.

Notre projet sera décomposé en trois cycles, eux-mêmes découpés de la façon suivante :

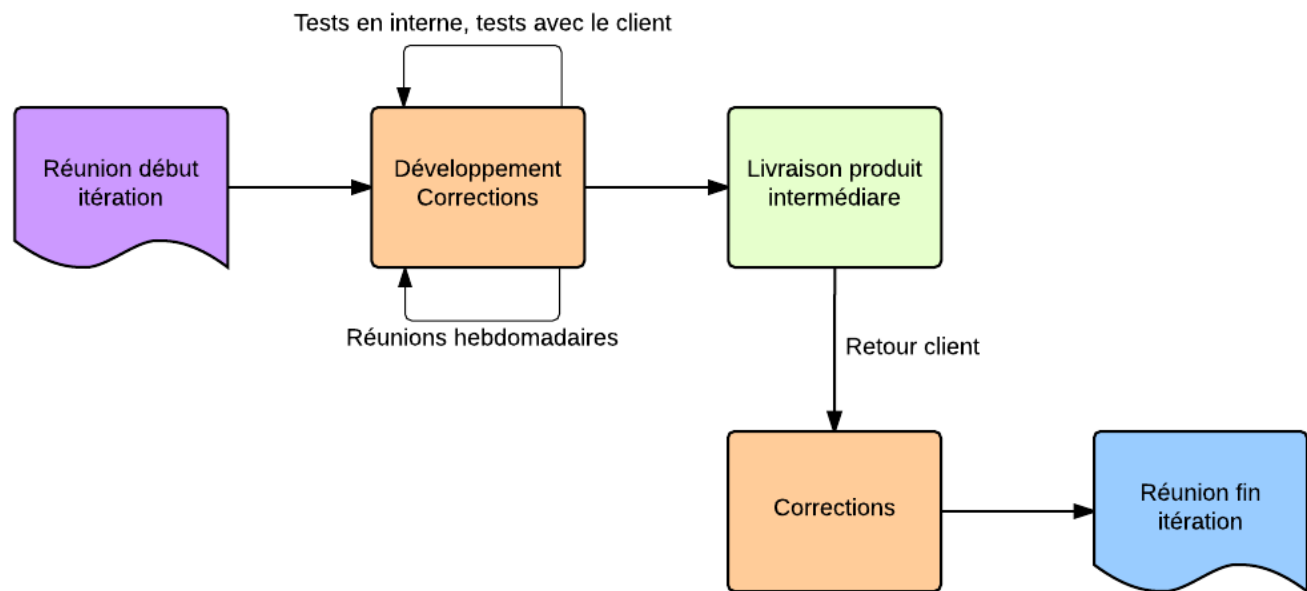


FIGURE 2 – Décomposition d'un cycle de développement

2.3 Réunion de début d'itération

Cette réunion se fera systématiquement le premier jour du cycle. Elle permet au groupe de se réunir, de fixer les objectifs du cycle ainsi que rappeler les tâches à effectuer par binôme et les dates limites. Elle permet de répondre aux éventuelles interrogations des membres de l'équipe.

2.4 Développement et corrections

Cette phase comporte plusieurs étapes. Tout d'abord il y a une première phase de développement. Pendant cette phase de développement, des réunions ou contacts réguliers avec le client sont organisés. Il est régulièrement tenu au courant (bilan hebdomadaire) de l'avancement, que ce soit par mail (avec des imprimés écrans montrant les fonctionnalités, ou un site hébergé de manière temporaire et privé), ou par réunion. Le client peut alors tester les fonctionnalités et nous faire parvenir ses retours et impressions. En parallèle, les tests sont effectués à chaque fois qu'une fonctionnalité est développée. Lors d'un retour, (qu'il parvienne du client ou du binôme ayant effectué les tests), les anomalies sont corrigées.

2.5 Livraison du produit intermédiaire

Cette phase a lieu quelques jours avant la fin de l'itération. C'est lors de cette réunion avec le client que les différents livrables (application et documents) lui sont présentés. Il peut alors nous faire parvenir ses réactions à chaud, ainsi que par mail quelques jours ensuite. De plus, le responsable client transmettra régulièrement par mail l'état de l'avancement du projet aux clients. Enfin, les clients pourront évaluer en temps réel l'avancement de l'application grâce à l'hébergement provisoire et privé mis en place. Ils pourront alors nous faire part de leurs impressions, et remplir le journal de tests s'ils rencontrent des problèmes. Un point client sera alors mis en place régulièrement.

2.6 Corrections

Après les retours du client, cette phase consiste à apporter les modifications demandées, ainsi que corriger les éventuels bugs détectés.

2.7 Réunion de fin d'itération

A la fin de chaque cycle aura lieu une réunion de fin d'itération. Elle aura pour but de réaliser un bilan du cycle passé et de déterminer les évolutions à adopter pour les prochains cycles si nécessaires.

2.8 Suivi hebdomadaire

Durant chaque cycle, une réunion de groupe hebdomadaire aura systématiquement lieu. Durant ces réunions, nous étudierons l'avancement des différentes tâches, nous évoquerons les éventuelles difficultés rencontrées et tenterons d'y trouver une solution. Des réunions supplémentaires pourront être fixées si nécessaire.

2.9 Versionning

À la fin de chacun de ces cycles sera livrée une version intermédiaire du logiciel. Nous livrerons donc trois versions du logiciel au client.

- 06/03/15 : Version 0.1
- 17/04/15 : Version 0.2
- 15/05/15 : Version 1.0

2.10 Documents

Les différents documents doivent être tenus à jour tout au long du projet. Pour chaque action effectuée, chaque membre du projet est tenu de compléter les documents correspondants.

3 Organisation et responsabilité

Notre équipe de projet est composée de sept personnes suivant l'organigramme suivant :

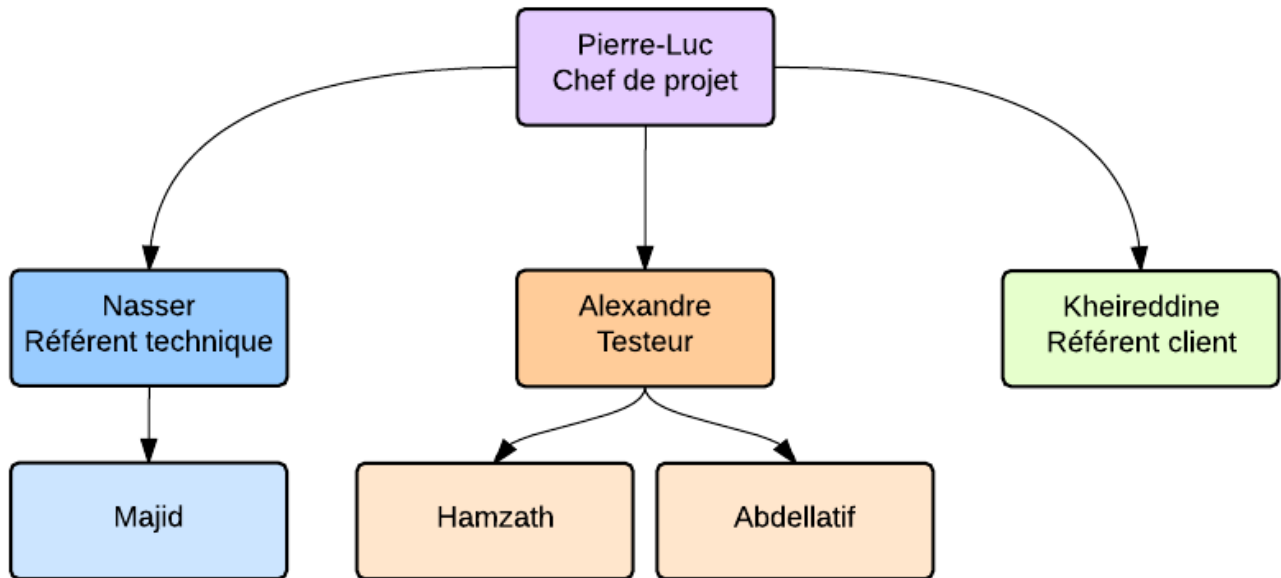


FIGURE 3 – Organigramme

3.1 Attribution des rôles au sein de l'équipe de développement

Nous avons décidé de répartir les rôles de manière équitable. En effet, chaque membre du groupe est un développeur mais possède également un rôle plus précis qui implique des responsabilités.

3.2 Chef de projet

Le chef de projet est le support de l'équipe. En effet il est le coordinateur de l'équipe car il doit mener le projet et gérer son bon déroulement. Pour cela, il doit estimer et planifier les phases de développement ainsi qu'assurer la qualité du logiciel. Le chef de projet est amené à suivre l'avancement du projet. De plus, il anime l'équipe et s'occupe de la communication entre les membres de l'équipe et doit gérer les éventuels problèmes. Pierre-Luc a été désignée chef de projet, pour ses capacités d'encadrement, son organisation et sa rigueur.

3.3 Chargé client

Le chargé client représente l'équipe auprès du client. Il a une vision à la fois technique et générale du projet. Il est amené à communiquer avec le client pour lister ses besoins et définir les ordres du jour lors des réunions. Il doit informer le client sur l'avancement du projet et maintenir le contact avec lui tout au long du projet. Kheireddine a été désigné chargé client car il aime le contact et communique aisément. Pierre-Luc a été désignée chargée client suppléante en cas d'absence de Kheireddine.

3.4 Testeur

Le testeur s'occupe de la planification, la création et la réalisation des tests. Il est chargé de produire des rapports réguliers décrivant les erreurs trouvées dans le produit. Dans ce rapport doivent figurer les

étapes nécessaires pour reproduire le bug. De ce fait, il doit faire remonter tous les problèmes rencontrés lors des tests le plus rapidement possible au chef de projet. Son principal rôle est donc de vérifier le bon fonctionnement du logiciel. Alexandre a été désignée testeur pour ses connaissances en POO. Pour les même raisons, Hamzath et Abdallatif ont également été désignés.

3.5 Responsable technique et développement

Le responsable technique et développement est l'expert technique du projet. Il est chargé de modéliser les différents composants et créer le modèle d'architecture logicielle. Il doit pour cela comprendre le fonctionnement interne de l'application. C'est lui qui est également chargé de définir les éventuels API qui vont être utilisés ainsi que les différents langages de programmation. Il doit être une référence pour les autres développeurs dans les langages et API qu'il a choisi. Nasser a été désigné responsable technique et développement car il a de bonnes connaissances LLVM et il les maintient à jour régulièrement. De plus, il a un esprit synthétique permettant de définir clairement les différents composants. Pour les mêmes raisons, Majid a également été désignée sur ce rôle.

3.6 Développeur, programmeur

Le développeur, programmeur doit suivre les différentes directives du responsable technique et du chef de projet. Il est amené à proposer des idées d'amélioration. Il est chargé de développer l'application en respectant les différents documents livrables. Il doit corriger les erreurs remontées lors des différents tests. Il doit avoir des connaissances dans les langages et API utilisés, ou se former si nécessaire. Nous sommes tous développeur, programmeur en plus de nos précédents rôles.

3.7 Manager

Le manager joue un rôle de conseiller extérieur. Il fournit un ensemble de méthodes et nous guide tout au long du projet. Ceci dans le but d'aider l'équipe à acquérir les points clés nécessaires au bon déroulement du projet. Il est également présent pour mesurer l'avancement du projet et nous signaler d'éventuels écarts ou manquements à nos devoirs afin de nous aiguiller. Le manager possède une aisance pour la communication et pour l'enseignement. Il dispose de connaissances à la fois générales et techniques. Notre manager est Karim Abdellah Godard.

3.8 Client

Le client fait appel à nos services pour nous exposer son besoin. Il nous accompagne au maximum tout au long du projet pour nous faire part de ses remarques et valider notre travail. Il est également présent pour tester les fonctionnalités et remonter d'éventuels problèmes à la fin de chaque cycle de développement. Son étroite collaboration tout au long du projet doit permettre aux développeurs d'aller dans son sens dans le but de produire un travail qui réponde le plus possible au besoin initial. Notre client est Mr Nicart.

3.9 Professeur référent

Le professeur référent agit comme une interface entre l'équipe de développement et le client. Il nous aide à comprendre son besoin, à le contacter et à établir des moments de rencontre. Il suit l'avancement de notre travail et le valide. Notre professeur référent est Mr Nicart.

4 Procédés de gestion

4.1 Gestion de la documentation

Différent documents sont à produire pendant le projet, en voici la liste et le détail des responsabilités.

4.1.1 Spécification Technique du besoin (STB)

Nom	Spécification Technique du Besoin (STB)
But du document	<ul style="list-style-type: none"> — Traduire les besoins des utilisateurs et établir une référence pour sa validation — Recenser les exigences que l'équipe de développement s'engage à satisfaire.
Responsabilités de rédaction	<ul style="list-style-type: none"> — ADJIBI Nasser — BERKANE Kheireddine
Responsabilité de relecture	<ul style="list-style-type: none"> — BLOT Pierre-Luc — PETRE Alexandre
Appobation requise	Mr Nicart

4.1.2 Description Architecture Logicielle

Nom	Description Architecture Logicielle
But du document	<ul style="list-style-type: none"> — Décrire les solutions techniques pour répondre aux exigences — Description et identification des différents modules ou constituants logiciel ainsi que leurs interfaces.
Responsabilités de rédaction	<ul style="list-style-type: none"> — ADJIBI Nasser — TASSERIE Majid
Responsabilité de relecture	<ul style="list-style-type: none"> — BLOT Pierre-Luc — BERKANE Kheireddine
Appobation requise	Mr Nicart

4.1.3 Cahier de recettes

Nom	Cahier de Recettes (CdR)
But du document	<ul style="list-style-type: none"> — Définition des moyens et procédés mis en oeuvre pour assurer la recette du logiciel développé. — Vérifier que le logiciel est conforme à la spécification technique du besoin. — Recenser les objectifs de tests de validation et les moyens nécessaires pour les atteindre.
Responsabilités de rédaction	<ul style="list-style-type: none"> — PETRE Alexandre — IDRISSOU Hamzath
Responsabilité de relecture	<ul style="list-style-type: none"> — BLOT Pierre-Luc
Appobation requise	Mr Nicart

4.1.4 Analyse des Risques (AdR)

Nom	Analyse des Risques (AdR)
But du document	<ul style="list-style-type: none"> — Recenser les risques ainsi que leurs probabilités. — Les actions à effectuer en cas de réalisation de ces risques.
Responsabilités de rédaction	<ul style="list-style-type: none"> — BLOT Pierre-Luc
Responsabilité de relecture	<ul style="list-style-type: none"> — BERKANE Kheireddine
Appobation requise	Mr Nicart

4.1.5 Plan de Développement (PdD)

Nom	Plan de Développement (PdD)
But du document	— Décrire l'organisation du projet et les règles de gestion.
Responsabilités de rédaction	— BLOT Pierre-Luc — BERKANE Kheireddine
Responsabilité de relecture	— ADJIBI Nasser
Appobation requise	— Mr Nicart — Mr Abdellah

4.1.6 Manuel d'utilisation

Nom	Manuel d'utilisation
But du document	— Expliquer l'utilisation du compilateur.
Responsabilités de rédaction	— ADJIBI Nasser — PETRE Alexandre — TASSERIE Majid — BLOT Pierre-Luc — BERKANE Kheireddine — IDRISSOU Hamzath — AHOUATE Abdellatif
Responsabilité de relecture	— ADJIBI Nasser — PETRE Alexandre — TASSERIE Majid — BLOT Pierre-Luc — BERKANE Kheireddine — IDRISSOU Hamzath — AHOUATE Abdellatif
Appobation requise	Mr Nicart

4.2 Gestion des configurations

Les documents sont stockés dans un dépôt git privé sur gitlab. Tous les membres de l'équipe y ont accès. Cela nous permet d'obtenir les documents rapidement et dans leurs versions les plus récentes. Chaque personne doit rédiger au moins un document et un document ne doit pas être rédigé par plus de deux personnes. Les versions des documents doivent être relues par l'ensemble de l'équipe (et donc corrigé si nécessaire) avant d'être soumises au client et aux professeurs concernés.

Les sources sont hébergées également sur gitlab. Cela nous permettra d'avoir une version centralisée

dans le dépôt git à laquelle tous les membres du groupe pourront accéder. De plus, nous aurons nos propres versions sur nos ordinateurs sur des copies du dépôt git. Cela limite très fortement les risques de perte de données. De plus git permet de gérer les versions par commit ou par tag.

Les membres de l'équipe seront chargés de mettre en ligne leurs fichiers développés en ayant vérifié sur leur version de l'application que l'intégration ne posait pas de problème. Le tout doit se faire sur la branche global du dépôt git.

Cette branche global contient l'avancement du projet, elle peut contenir des bugs. Dès qu'une version sans bug est implémentée, alors on fusionne la branche global sur la branche master. Ainsi, la branche master contient seulement les versions du projet qui sont fonctionnelles et correctes.

L'intégration se fera automatiquement puisque grâce à git nous pouvons directement travailler par itération sur plusieurs branches parallèles pour enfin toutes les fusionner au moment voulu. Ces fusions seront coordonnées par le responsable technique.

De plus, l'équipe travail slack. Slack est une plateforme sur laquelle tous les membres de l'équipe sont inscrit. Slack permet aux membres de l'équipe de communiquer via une messagerie instantanée organisée en chaîne de discussion. Slack permet de conserver les historiques de conversation, d'échanger des fichiers, de connecter le compte gitlab sur une chaîne. Ainsi dès que des modifications sont apportées sur le dépôt git en ligne, une notification apparaît avec les trois derniers commit effectués.

Slack peut être installé sur smartphone, et cela offre une grande flexibilité et offre une meilleure communication.

Le client peut accéder au dépôt privé sur gitlab et reçoit par mail une notification dès qu'une version du projet est pushée sur le dépôt en ligne.

5 Revues et points clefs

Voici un récapitulatif rapide de l'ensemble des dates clés du projet kawa. Cette liste n'est pas exhaustive et les dates peuvent être amenées à changer en cours de projet.

Dates	Evénements
17/10/14	Remise du sujet
06/01/15	Remise des documents v1.0 : AdR, CdR, DAL, PdD et STB
19/01/15	Revue de lancement du projet
20/01/15	Début de la phase de développement
06/03/15	Premier livrable
17/04/15	Deuxième livrable
15/05/15	Troisième livrable