

Plan de projet

Application de téléphonie IP

1.1

Falimanana Razafindrabe

Johan Yémanlin Sintondji

Joël Villeneuve

29 octobre 2019

# Historique des révisions

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Auteur** |
| 2019-10-29 | 1.3 | Première ébauche | SoraTech |

# Définitions

|  |  |
| --- | --- |
| Terme | Définition |
|  |  |

# Abréviations/acronymes

|  |  |
| --- | --- |
| Abré./Acro. | Définition |
|  |  |

# Table des matières

[Historique des révisions 2](#_Toc525472854)

[Définitions 2](#_Toc525472855)

[Abréviations/acronymes 2](#_Toc525472856)

[Table des matières 3](#_Toc525472857)

[1. Introduction 4](#_Toc525472858)

[1.1. Vue d’ensemble du projet 4](#_Toc525472859)

[1.2. Références 4](#_Toc525472860)

[1.3. Portée 4](#_Toc525472861)

[1.4. Livrables 4](#_Toc525472862)

[2. Organisation du projet 5](#_Toc525472863)

[2.1. Composition de l’équipe du projet 5](#_Toc525472864)

[2.2. Stratégie de contrôle des versions 5](#_Toc525472865)

[2.3. Directives de livraison 5](#_Toc525472866)

[3. Ressources matérielles, calendrier et budget 6](#_Toc525472867)

[3.1. Ressources matérielles 6](#_Toc525472868)

[3.2. Calendrier des taches du projet 6](#_Toc525472869)

[3.3. Budget 6](#_Toc525472870)

[4. Gestion des risques 7](#_Toc525472871)

[Annexe A – Directives de livraison 8](#_Toc525472872)

[Tableau 1- Livrables sprint1 5](#_Toc23256128)

[Figure 1 - Vue d'ensemble du projet 4](#_Toc23256159)

[Figure 2-Diagramme de Gantt 8](#_Toc23256160)

# 

# Introduction

## Vue d’ensemble du projet

Le projet consiste au développement d’un système de téléphonie IP. Ainsi, l’équipe SoraTech poursuit son engagement avec la réalisation d’un programme avec une nouvelle interface graphique représentant un softphone(client) propre à l’entreprise et de l’automatisation de la mise en place d’un serveur (Freeswitch). Dans la première étape, il sera question de développer un softphone avec le langage python en utilisant une librairie développée en C/C++ pour la communication avec le serveur. La librairie utilisée devrait permettre le transport de flux de données (voix, données, images ou vidéo) sur tout réseau et sur la même infrastructure, numérique ou analogique, acceptant le protocole TCP/IP (Ethernet, RNIS, PPP, etc.). De plus, elle aura les fonctionnalités du PABX traditionnel comme les renvois d’appel, mise en attente, conférence etc. Pour se faire, on a opté pour FreeSwitch comme serveur de communications (autocommutateur téléphonique privé, PABX-IP) en raison de sa facilité d’installation sur un système d’exploitation*Windows.* La version utilisée sera Freeswitch 1.10.1 qui permet entre autres la messagerie vocale, la prise en charge des files d’attente, les vidéoconférences, la télécopie etc.

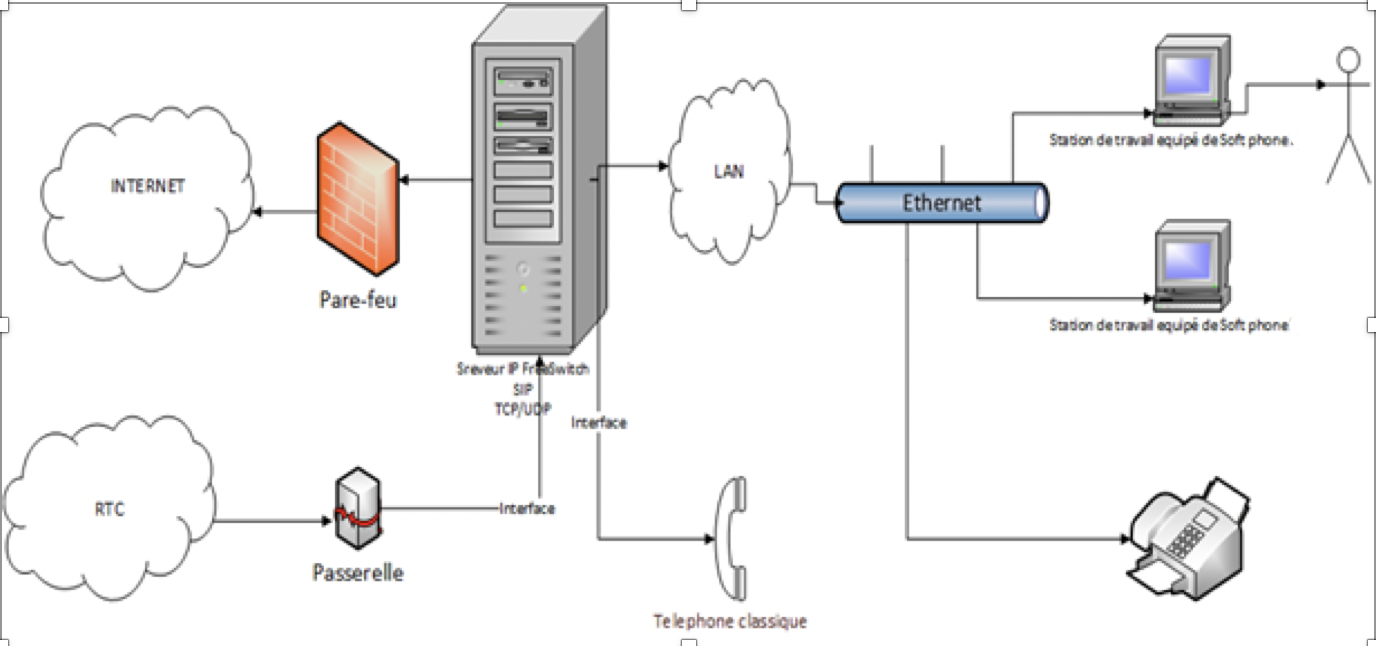


Figure 1 - Vue d'ensemble du projet

## Références

## Portée

L’équipe de SoraTech procédera à la mise en place de sa propre application de téléphonie sur ordinateur. Elle s’engage donc à:

Créer une interface pour l’application avec tkinter

Établir la communication entre l’application et le serveur FreeSwitch

Implémenter la solution complète pour la communication par VoIP

Migration complète des 25 postes de travail et les téléphones cellulaires en communications par IP.

Par ailleurs, ce projet ne tient pas compte de la mise en place de l’installation terminaux-IP que l’on nomme IP- Phones. Le présent plan de projet assume aussi que l’entreprise ACME Inc est seule responsable de l’accès et les problèmes liés à la connexion sur Internet.

## Livrables

|  |  |
| --- | --- |
| **Livrables** | **Date de livraison** |
| Diagramme des taches | 29/10/2019 |
| Fichier de configuration | 29/10/2019 |
| Plan de projet\* | 04/12/2019 |
| Spécifications\* | 04/12/2019 |
| Composantes\* | 04/12/2019 |
| Architecture\* | 04/12/2019 |
| Application de téléphonie | 04/12/2019 |

Tableau 1- Livrables sprint1

\*Les 4 documents seront remis au client à chaque semaine afin de montrer l’avancement.

# 

# Organisation du projet

## 2.1. Composition de l’équipe du projet

Falimanana Razafindrabe est responsable du côté technique du projet, il s’assure que l'application fonctionne correctement avant de l’installer chez le client. Son rôle est de corriger toute imperfection au niveau du programme lui-même, de l’améliorer et d’être en mesure de l’installer chez le client.

Johan Sintondji est le responsable du côté client. Son rôle est de rejoindre le client afin de s’assurer que nous répondons à ses besoins. Il doit aussi s'assurer des dates de livraison, que le client a les ressources nécessaires à l’implantation du serveur ou s’il est prêt à faire toute modification nécessaire à l’installation de celui-ci.

Joël Villeneuve est le responsable du côté matériel du projet. Il s’assure d’avoir toutes les ressources en main afin de que le projet soit une réussite. Si celle-ci son manquante, son rôle est de voir à ce qu’on les obtienne à temps pour les livraisons. De plus, il est le scrum master de cette semaine. Il veillera donc à ce que les tâches soient bien effectués et dans la plus grande efficacité.

## 2.2. Stratégie de contrôle des versions

Notre stratégie est simple. Il s’agit de numéroter les versions selon le format SEMver. Celui-ci définit un système de numérotation sur 3 aspects (x.y.z, e.g.: 1.1.0). Lorsqu’il y a changement de x, on dit qu’il y a eu un changement majeur au niveau de l’interface de programmation. Lors d’une augmentation au niveau y, on parle d’un changement mineur comme un ajout de fonctionnalité et au niveau z on parle d’une légère mise à niveau comme la résolution d’un bug. On débutera avec la version 0.1.0 et ne changeront à la version 1.1.0 qu’après installation du programme. Afin de contrôler toutes ces versions, nous ferons une documentation numérotée de la même façon qui commentera chacun des changements effectués entre les versions.

## 

## 2.3. Directives de livraison

La livraison sera procédée comme suit:

* Démonstration de la connexion entre le serveur et l’application
* Démonstration d’un appel entre deux clients utilisant notre application
* Démonstration du transfert d’appel, de la boîte vocale et de l’appel en attente à l’aide de l’application

## 2.4. Outils d’organisation

Les outils utilisés pour l’organisation de ce projet sont:

* GIT et OneDrive pour le partage des fichiers
* Facebook pour la communication
* MS Project pour la planification

# Ressources matérielles, calendrier et budget

## 3.1. Ressources matérielles

Les différents matérielles nécessaires pour la réalisation du projet sont comme suit :

* Local pour travailler équipé de serveur sous Windows 10 et de deux postes de travail.
* Deux casques d’écoutes avec microphones
* Programmes d’installation du PABX-IP(FS)
* Programme d’installation du nouveau softphone

Les casques d’écoutes seront commandés par notre responsable en matérielles et le programme sont fourni par nos spécialistes. 

## 3.2. Calendrier des taches du projet

Tableau des tâches du sprint 2

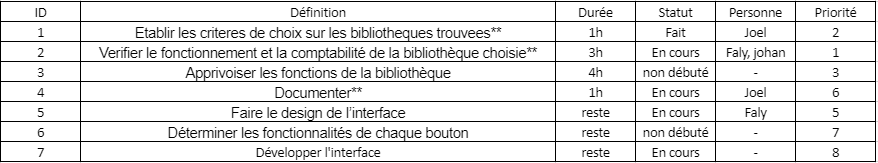


Tableau des tâches du sprint 3

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Tâche** | **Découpage** | **Estimé** | **Attribution** | **Statut** | **Réel** | **Explication** |
| **Continuer les 4 documents** | Plan de projet | 30 mins | Joël | Fait | 10min | Nous avons compris le fonctionnement du PJSUA et il fut donc plus facile de le documenter |
| 2 heures | Spécifications | 30 mins | Faly | Fait | 10min |
|  | Composantes | 30 mins | Joël | Fait | 10min |
|  | Architecture et conception | 30 mins | Joël | Fait | 10min |
| **Interface GUI** | Clavier numérique | 30 mins | Faly | Fait | 1h30 | Il a fallut se documenter sur l'utilisation de tkinter |
| 1 heure | Boutons appel hold … | 30 mins | Faly | En cours | 30min |
| **Module PJSUA en python 3.6** | Installer les composants | 1h | Johan | Fait | 30 mins | La compilation du PJSUA fut plus difficile que prévue, on s'est rendu compte qu'il y avait des bugs au niveau du code en lui-même de PJSUA et sur les compilateurs aussi |
| 3 heures | Génerer l'extension en C | 30mins | Johan | Fait | 1h |
|  | Créer module et l'installer | 30 mins | Johan | Fait | 5h30 |
|  | tester | 1h | Johan | Fait | 1h |
| **Test des fonctionnalité PJSUA** | Test création client | 1h | Johan/Faly | Fait | 20 mins | Lorsque le PJSUA fut finalement installé, on s'est rendu compte qu'il y avait des tests de fonctionnalité déjà présent et ce fut donc plus rapide que prévu |
| 3 heures | Appeler | 1h | Johan/Faly | Fait | 20 mins |
|  | Test en ligne | 1h | Johan/Faly | Fait | 20 mins |

Tableau 2Tableau des taches du sprint4

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Tâche** | **Découpage** | **Estimé** | **Attribution** | **Statut** | **Réel** | **Explication** |
| **Continuer les 4 documents** | Plan de projet | 30 mins | Groupe | fait | 30mins |  |
| 2 heures | Spécifications | 30 mins | Groupe | fait | 30mins |
|  | Composantes | 30 mins | Groupe | fait | 30mins |
|  | Architecture et conception | 30 mins | Groupe | fait | 30mins |
| **Terminer l'interface tkinter** | Faire les derniers boutons | 30 mins | Fali | fait | 1h | Freeswitch aprit plus de temps car on a structuré le programme pour faciliter l'ajout des fonctions du pjsua |
| **Connecter PJSUA au serveur FreeSWITCH** | Se documenter sur les classes et les fonctions du PJSUA | 3h | Johan/Fali | fait | 3h |
| **7.5h** | Refaire l'installation de FreeSWITCH | 30 mins | Johan | fait | 15mn |  |
|  | Créer des clients à partir de PJSUA | 2h | Johan | fait | 3h |
|  | Configurer les protocoles de transport du données avec PJSUA | 2h | Joel | en cours | 2h |

## 3.3. Budget

Les coûts fixes et les coûts récurrents pour réaliser le projet sont décrits dans le tableau ci-dessous :

Tableau 1 - Représentation du budget

|  |  |  |  |
| --- | --- | --- | --- |
| **Coûts fixes** | **Quantité** | **Prix/unité** | **Sous-total** |
| Câblage | 25 | 60 | 1500 |
| Casque avec microphone | 25 | 24 | 600 |
| Total |  |  | 2100 |
| **Coûts récurrents** | **Quantité** | **Prix($/mois)** | **Sous-total** |
| Lignes fixes | 25 | 20 | 500 |
| Cellulaires | 15 | 30 | 450 |
| Total |  |  | 950 |

# Gestion des risques

**Risque 1:** Latence

*Description:*   
La latence est le temps de réaction du système c’est à dire le temps écoulé entre le stimulus et la réponse correspondante. Suite à l’installation des services, le système pourrait être sujet à la présence de latence en communication vocale. Plusieurs facteurs pourraient être en causes comme le débit de transmission sur chaque lien, le nombre d’éléments réseaux traversées, la qualité du matériel (routeur, câbles etc.), le délai de propagation de l’information, le codage des flux analogiques, les codecs utilisés...De plus chaque paquet IP fait son chemin indépendamment de l’ordre initial. Le délai de transmission des paquets est très important en téléphonie IP pour éviter la perception d'écho ou de coupure.

*Méthodes de diminution ou contention :*

Pour garantir un délai minimal d’acheminement afin d’avoir une conversation acceptable, il est nécessaire d’utiliser un système de qualité:

Avoir un mécanisme pour stocker les paquets à l’avance

Avoir une bande passante acceptable

Avoir un protocole de transports simplifiés pour ne pas ralentir l’acheminement des données (UDP)

**Risque 2** : Gigue

*Description:*   
La gigue est la variance statistique du délai de transmission. La gigue est générée par la variation de charge du réseau (variation de l'encombrement des lignes ou des équipements réseau) et donc à la variation de routes dans le réseau. Chaque paquet est en effet susceptible de transiter par des combinaisons différentes de routeurs entre la source et la destination.

*Méthodes de diminution ou contention:*

Pour compenser la gigue, on peut utiliser des buffers (mémoire tampon) côté récepteur, afin de reconstituer un train continu et régulier de paquets voix. Leur taille doit donc Ítre soigneusement définie et, si possible, adaptée de manière dynamique aux conditions du réseau. Toutefois, cette technique a l'inconvénient de rallonger le délai d'acheminement des paquets. Il est donc préférable de disposer d'un réseau à gigue limitée.  Leur taille doit donc Ítre soigneusement définie et, si possible, adaptée de manière dynamique aux conditions du réseau.

**Risque 3** : Perte de paquets

*Description:*

Lorsque les mémoires des différents éléments du réseau IP sont congestionnés, ils libèrent automatiquement de la bande passante en se débarrassant d'une certaine proportion des paquets entrants, en fonction de seuils prédéfinis. Si aucun mécanisme performant de récupération des paquets perdus n'est mis en place, alors la perte de paquets IP se traduit par des ruptures de la conversation et une impression de coupure de la parole.

*Méthodes de diminution ou contention:*

Le moyen le plus efficace de lutter contre la perte d'informations consiste à transmettre des informations redondantes (code correcteur d'erreurs), qui vont permettre de reconstituer l'information perdue. Des codes correcteurs d'erreurs, comme le Reed Solomon, permettent de fonctionner sur des lignes présentant un taux d'erreur de l'ordre de 15 ou 20 %. Une fois de plus, ces codes correcteurs d'erreurs présentent l'inconvénient d'introduire une latence supplémentaire.

**Risque 4**: Sécurité

*Description:*   
Les réseaux VoIP sont exposés aux mêmes virus et attaques de pirates que ceux des données de base.

Les outils classiques sont répandus d’écoute de réseau et peuvent être directement utilisés pour attaquer un réseau VoIP. À ceux-ci, s’ajoutent la faible sécurité des protocoles utilisés par la voix sur IP ainsi que les protocoles qui ont déjà fait ou font l’objet de failles de sécurité. Conséquemment, dès que le pirate informatique est entré sur le réseau de l’entreprise au moyen d’outils d’intrusion et de reniflage, ce dernier aura accès aux communications de l’entreprise et à ses données.

*Méthodes de diminution ou contention:*   
Pour améliorer la sécurité en communication par IP, plusieurs éléments de l’infrastructure doivent être examinés et traités:

L’ensemble du réseau filaire ou sans fil, les applications des systèmes et les terminaux

Les flux de téléphones doivent être isolés des flux de données

Vérifier l'existence de mécanismes de détection des attaques

Chiffrer les données

Utilisations de protocole sécurisé, authentification de l’utilisateur, gestion étroite des droits d’accès

Envisager plus tard de migrer sur des systèmes d’exploitation plus sécuritaires comme Linux.

# Annexe A – Directives de livraison

Projet : Application de téléphonie

Client : ACME

Date : 04 décembre 2019

Identification des livrables

Plan de projet, spécifications, composantes et architecture du logiciel

Exigences de livraison

* Connexion entre le serveur et l’application
* Appel entre deux clients utilisant l’application
* Transfert d’appel
* Appel en attente
* Boîte vocale

Critère d’acceptation Date de complétion

Composant Version livré

Approuvé par : Jean-Luc Cyr

Date d’approbation : 29-10-2019