

---

## Logiciel « Chat »

---

### Plan de test

---

Auteur	Contact
Bastien Potiron	bastien.potiron@etudiant.mines-nantes.fr

Version	Date	Commentaire
0.1	15/01/2017	Création du corps du plan de test ;
0.2	27/01/2017	Ajout de nouveaux éléments et commentaires sur le fichier ;
0.3	01/02/2017	Complétion des listes de test effectués ;
0.4	02/02/2017	Ajout de la partie 4 du document ;

## Table des matières

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Objectif des spécifications . . . . .	3
1.2	Portée du document . . . . .	3
1.3	Modification du plan de test . . . . .	3
1.4	Termes et abréviations . . . . .	3
<b>2</b>	<b>Références</b>	<b>4</b>
2.1	Normes et standards . . . . .	4
2.2	Documents internes . . . . .	5
<b>3</b>	<b>Périmètre de test</b>	<b>5</b>
3.1	Composants . . . . .	5
3.1.1	Composants concernés par les tests . . . . .	5
3.1.2	Composants non concernés par les tests . . . . .	5
3.2	Fonctionnalités . . . . .	5
3.2.1	Fonctionnalités testées . . . . .	5
3.2.2	Fonctionnalités non testées . . . . .	6
3.3	Critères d'acceptation des tests . . . . .	7
<b>4</b>	<b>Processus et stratégie de test</b>	<b>7</b>
4.1	Activités . . . . .	7
4.2	Techniques de test . . . . .	7
4.3	Outils de test . . . . .	7
4.4	Procédures de test et gestion des anomalies . . . . .	8
<b>5</b>	<b>Infrastructure de test</b>	<b>10</b>
<b>6</b>	<b>Documents de test et livrables</b>	<b>10</b>
<b>7</b>	<b>Responsabilités</b>	<b>10</b>
<b>8</b>	<b>Équipe de test</b>	<b>10</b>
<b>9</b>	<b>Planning effectif</b>	<b>10</b>

## 1 Introduction

### 1.1 Objectif des spécifications

Ce document décrit l'activité de test qui sera menée par Bastien POTIRON durant le projet Test Logiciel dans le but de valider les produits Felix & Camix logiciel de chat. Il est rédigé sous la responsabilité du responsable de test.

### 1.2 Portée du document

Sont concernés par ce document :

- les testeurs : afin que ceux-ci sachent ce qu'ils vont tester, comment ils le testent et comment ils rendent compte des résultats de ces tests ;
- les développeurs : à titre informatif, afin que ceux-ci sachent comment va être validée leur production ; à titre indicatif afin qu'ils sachent, par la description de la gestion des anomalies, comment ils s'interfaceront avec l'équipe de test ;
- le client : ce plan de test fait l'objet d'une contractualisation avec le client pour déterminer le périmètre des tests menés pour valider le produit livré et les niveaux d'acceptation de cette validation ;
- les auditeurs : ce plan de test, ainsi que son implication, feront l'objet d'audits par Matthias Brun et Camille Constant ;

### 1.3 Modification du plan de test

Le plan de test peut être soumis à des modifications. Toute nouvelle version sera signalée par mail à l'ensemble des intervenants du projet.

### 1.4 Termes et abréviations

Les termes et abréviations donnés ci-dessous seront utilisés dans tout le document. En cas d'interrogation vous pourrez donc vous reporter à cette section.

Les définitions données sont principalement issues du glossaire des tests de logiciel du CFTL (Comité Français de Test Logiciel).

Abr.	Signification	Définition
CdP	Bouchon	une implémentation spéciale ou squelettique d'un composant logiciel, utilisé pour développer ou tester un composant qui l'appelle ou en est dépendant. Cela remplace un composant appelé. [d'après IEEE 610] ;
	Cahier de Test	liste exhaustive de tous les tests pratiqués par le testeur avant la livraison du produit ;
	Chef de Projet	personne chargée de s'assurer du bon déroulement du projet de test ;
	Plan de Test	Document décrivant l'étendue, l'approche, les ressources et le planning des activités de test prévues. Il identifie entre autres les éléments et caractéristiques à tester, l'affectation des tâches, le degré d'indépendance des testeurs, l'environnement de test, les techniques de conception des tests et les techniques de mesure des tests à utiliser ainsi que tout risque nécessitant la planification de contingence. Il constitue la documentation du processus de planification de test. [d'après IEEE 829]
RT	Responsable Test	Personne responsable du test et de l'évaluation d'un objet de test. Individu qui dirige, contrôle, gère l'organisation et règle l'évaluation d'un objet de test ;
	Spécification de Test	Document qui consiste en une spécification de conception du test, des spécifications de cas de test et/ou des spécifications de procédures de test ;
TI	Test d'Intégration	tests effectués pour montrer des défauts dans les interfaces et interactions de composants ou systèmes intégrés ;
TU	Test Unitaire	le test de composants logiciels individuels [d'après IEEE 610] ;
TV	Test de Validation	assurent que les exigences client sont respectées d'un point de vue cas d'utilisation ;
	Validation	Confirmation par l'examen et la fourniture de preuves objectives que les exigences, pour un usage ou une application voulue, ont été satisfaites. [ISO 9000] ;

## 2 Références

### 2.1 Normes et standards

Nom	Version	Lien avec le PT	Source
Norme 829	01/2016	Norme IEEE dont le document s'inspire pour la rédaction ;	lc.cx/JNe2
Glossaire des tests logiciel	03/2015	utilisé pour obtenir les définitions normées par la CFTL des éléments utilisés dans le glossaire ;	lc.cx/JNe6

## 2.2 Documents internes

Nom	Version	Lien avec le PT	Source
Spécifications	0.5	Description des CU utilisés dans le plan de test ;	Archive finale du projet
Cahier de test	1.0	Archive tous les résultats obtenus sur les tests effectués par l'équipe ;	Archive finale du projet

## 3 Périmètre de test

Cette section s'appuie sur les spécifications du projet.

### 3.1 Composants

Cette section s'appuie sur la section déploiement du document de spécifications.

#### 3.1.1 Composants concernés par les tests

Seront concernés par l'activité de test les composants logiciels développés durant le projet :

- Felix : composant utilisé pour se connecter au serveur de chat et interagir dans un salon de discussion ;
- Camix : serveur de chat permettant l'hébergement de salon de discussion.

La couche applicative de communication par protocole TCP/IP (modèle OSI) permettant la communication entre les composants du projet sera également concernée par l'activité de test.

#### 3.1.2 Composants non concernés par les tests

Ne seront pas concernés par les tests :

- les supports d'exécution logiciels (systèmes d'exploitations, environnements Java (JVM)) ;
- les supports matériels (machines lançant les composants) ;
- les supports de communication matériels (routeur, borne wifi, câble RJ45) ;
- les supports de communication logiciels (pile TCP/IP) ;
- les logiciels annexes au projet ne seront pas non plus concernés par l'activité de test (logiciel de base de données) ;

### 3.2 Fonctionnalités

Cette section s'appuie sur la section CU du document de spécifications.

#### 3.2.1 Fonctionnalités testées

Pour le tableau suivant, on distingue trois niveau de validation :

- A : correspond à une fonctionnalité entièrement testée et considérée comme fonctionnelle ;
- B : correspond à une fonctionnalité considéré comme fonctionnelle mais non entièrement testée ;
- C : correspond à une fonctionnalité dont le test n'est pas concluant ;

Fonctionnalité	CU Concerné	Validation
Connexion au serveur de chat	Connexion au chat	A
Gestion de l'erreur de connexion au chat	Connexion au chat	A
Ouverture de la fenêtre de chat	Connexion au chat	A
Fermeture du programme Felix	Sortir du chat	C
Sortie du chat pour retour sur le fenêtre de connexion ('/q')	Sortir du chat	C
Changer de canal de discussion	Gérer des canaux de discussion	A
Créer un canal de discussion (+ alternatives)	Gérer des canaux de discussion	A
Supprimer un canal de discussion (+ alternatives)	Gérer des canaux de discussion	A
Ajout d'un client dans un canal	Entrer dans le chat	A
Informé du départ d'un client	Sortir du chat	A
Camix transmet le message d'accueil au nouveau client	Entrer dans le chat	A
Camix change le surnom d'un client	Changer de surnom	A
Le composant Felix du client affiche un message d'accueil dans le chat	Entrer dans le chat	A

### 3.2.2 Fonctionnalités non testées

Le tableau ci-dessous contient toutes les fonctionnalités non testées ainsi que les CU correspondant (se reporter aux spécifications pour plus de détail) :

Fonctionnalité	CU Concerné
Camix inscrit le client dans le canal par défaut	Entrer dans le chat
Camix informe les utilisateurs de l'arrivée d'un client	Entrer dans le chat
Chaque composant Felix concerné affiche un message dans le chat	Entrer dans le chat
Felix transmet une commande à Camix	Communiquer avec d'autres clients
Camix transmet à Felix les informations sur les commandes disponibles (aide)	Obtenir de l'aide
Felix affiche un message d'aide sur les commandes disponibles	Obtenir de l'aide
Camix transmet un message aux composants Felix des utilisateurs inscrits dans le canal du client	Echanger des messages
Chaque composant concerné affiche le message	Echanger des messages
Camix informe le client que la commande n'est pas valide (cas alternatif)	Changer de surnom
Afficher ses informations personnelles	Afficher des informations
Afficher des informations d'aide sur les canaux	Afficher des informations

### 3.3 Critères d'acceptation des tests

Le niveau de validation attendu pour les tests implémentés est de 100% pour les tests de validation, 90% pour les test unitaires et 80% pour les tests d'intégration.

En pratique le niveau de validation obtenu est de 0% pour le test de validation, 90% pour les test unitaires et 100% pour les test d'intégration.

## 4 Processus et stratégie de test

### 4.1 Activités

- TV comportement nominal de Camix et de Felix ;
- TI du composant Camix (serveur) ;
- TU nominaux sur certains modules du composant serveur (Camix) et sur certaines classes du composant client (Felix) ;

### 4.2 Techniques de test

- analyse partitionnelle ;
- analyse aux limites ;

Les résultats des analyses sont indiqués dans le cahier de test joint à ce document.

### 4.3 Outils de test

Pour effectuer les différentes activités de test, les outils suivant ont été utilisés :

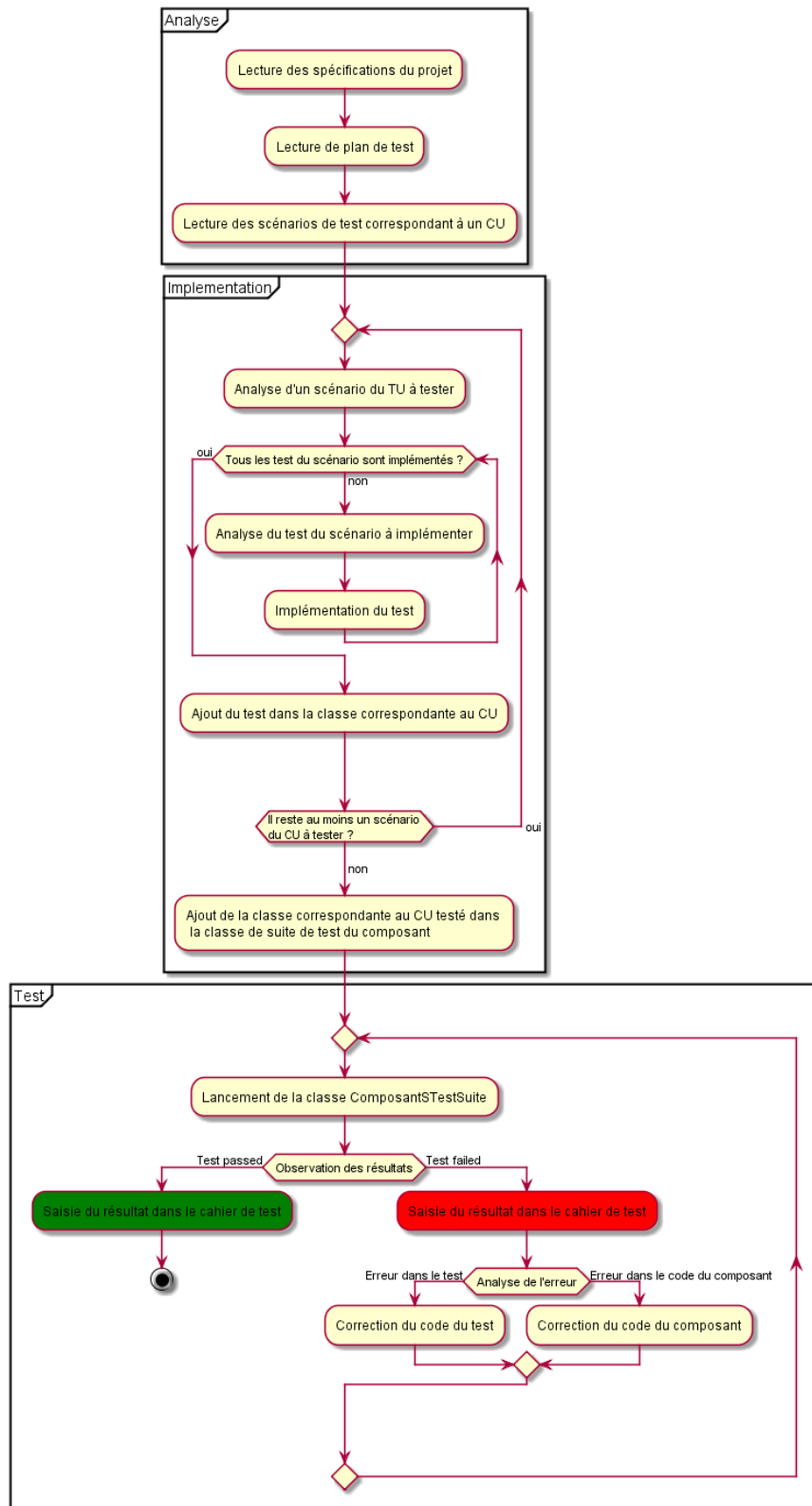
- Tests unitaires :
  - JUnit v4

- Bouchonnage avec EasyMock v3.4
- Jemmy v2.2.7.5
- Tests d'intégration :
  - JMeter v3.1
- Tests de validation :
  - JUnit v4
  - Jemmy v2.2.7.5
- Rédaction du cahier de test :
  - Microsoft Excel 2016
- Rédaction des documents de spécification et plan de test en Latex, via l'outil de collaboration 'Overleaf' ;

#### 4.4 Procédures de test et gestion des anomalies

Le diagramme d'activité ci-dessous décrit les procédures suivies lors de l'implémentation et le lancement des tests. Si le diagramme n'est pas lisible, notez que vous trouverez les sources dans le rendu final du projet, sous le nom 'diagrammeActiviteTU.tex' ou en version PNG sous le nom 'diagrammeActiviteTU.png'.





## 5 Infrastructure de test

Tous les tests liés au projet ont été lancés sur une machine Asus K501UW sous le système d'exploitation Windows 10. La machine était connectée via câble RJ45 à un routeur JCG U700 lui-même connecté à internet via Wifi lors de l'exécution des tests. Les tests ont tous été effectués sur un serveur local (localhost).

## 6 Documents de test et livrables

Nom	Objet	Format	Livable
Plan de test	Recenser l'ensemble de l'activité de test effectué	pdf	oui
Cahier de test	Recenser le résultat des tests effectués	xls	oui
Scénario des tests de validation	Expliquer le scénario de chaque TV mené	pdf	oui
Codes de test	Code informatique des tests effectués	java / xml	oui
Spécifications techniques du projet	Permet d'étudier la conception des programmes	pdf	oui
Code du projet	Permet de comprendre le fonctionnement interne des programmes	Java	oui

## 7 Responsabilités

- Le CdP : responsable des moyens mis à disposition pour mener à bien l'activité de test.
- Le RT : responsable de l'organisation et du déroulement de l'activité de test.
- Les testeurs : responsables des résultats de test reportés dans le cahier de test.

Il est important de spécifier que l'équipe de test ne peut être tenue responsable des répercussions d'une défaillance d'une fonctionnalité non validée.

## 8 Équipe de test

Chef de projet : Bastien Potiron.  
Responsable test : Bastien Potiron.  
Testeur : Bastien Potiron.

## 9 Planning effectif

Il s'agit ici d'un planning rédigé grossièrement permettant d'évaluer le temps passé sur chaque tâche dans le cadre du projet. Si parfois le temps passé sur une tâche peut sembler

long, c'est que celui-ci inclus le temps de montée en compétence à la fois sur l'architecture du projet mais également sur le concept de test derrière la tâche effectuée. Enfin une période de montée en compétence globale sur les différents éléments du projet, et leur liens a également été nécessaire. Ne pouvant être chiffrée, ce temps a été également réparti sur l'ensemble des tâches du projet.

<b>Tâche effectuée</b>	<b>Temps passé</b>
Rédaction du plan de test	7h
Mise à jour des spécifications des tests unitaires	4h
Développement des tests EasyMock pour Camix (ServiceChat)	4h
Développement des TI avec JMeter pour le plan de test	1,5h
Développement des test EasyMock pour Camix (ClientChat)	2,5h
Rédaction des scénarios de test	3h
Rédaction du cahier de test	2h
Développement des TU pour Felix (FenetreConnexion)	5h
Développement des TV pour Felix (Felix)	3h

On notera enfin que dans le cadre du projet, certaines négligences ont été commises, souvent par manque de connaissance dans le domaine et par soucis de rapidité. Exemple :

- Changement d'une classe de Camix : passer de 'public final class' à 'public class' pour faciliter l'utilisation du mock.
- Modification du constructeur de ClientChat (Camix) pour accepter les connexion avec un socket null.
- Lorsque certains points n'étaient pas clairs concernant les documents de conceptions attendus, des décisions arbitraires ont été prises. Celle-ci sont alors explicitées et justifiées.
- Rendu compte trop tard qu'il fallait insérer dans le cahier de test les traces d'exécution lorsqu'un erreur se produit. Ne pouvant prendre le temps de re-générer des erreurs identiques, le choix a été fait de ne pas noter les trace d'exécution si le test fail.