

GitHub : <https://github.com/insa-4ir-chatsystem/chatsystem-bonnet-cazeneuve>

Jira : <https://pinedepin.atlassian.net/jira/software/projects/CHAT/boards/3>

Bonnet Ronan
Cazeneuve Anna
4 IR B2
2023/2024

Rapport de Management et d'Automatisation du Projet Chatsystem

1. Introduction

Ce document a pour but de retracer l'organisation et l'utilisation des outils d'automatisation dans le travail effectué pour le projet chatsystem. L'objectif de ce projet était de développer une première version d' une application de clavardage dans un réseau d'entreprise. Il s'agit de permettre à des personnes d'un même réseau local de communiquer un à un par le biais de messages textuels. Pour avancer dans ce projet, nous avons utilisé la méthode Agile ainsi que des outils d'automatisation.

Nous tâcherons de présenter dans ce dossier ces deux différentes parties. Premièrement, nous montrerons la mise en place de la méthode Agile dans l'avancement de notre projet, grâce au logiciel de suivi de projet Jira. Enfin, nous traiterons l'automatisation par Intégration Continue dans notre travail sur Github.

Les annexes en fin de dossier viendront compléter et illustrer chaque partie.

2. Application de la méthode Agile

Dans cette partie nous traiterons les étapes de la méthode Agile que nous avons appliquées pour mener à bien notre projet.

Tout d'abord, nous avons cherché à extraire du sujet proposé (notre cahier des charges du **client**) les fonctionnalités requises. Dans cette étape, nous avons déterminé les **user stories** associées à l'application à développer. Nos user stories sont écrites du point de vue des actions effectuées par un client, et donc un utilisateur de l'application. Voici un exemple de user story que nous avons écrite :

«As a User, I can see the history of my messages with another user, so I can read the previous conversation.»

À chaque user story est associé un ticket sur **Jira**, dans lequel on vient rajouter en description des **critères de validation** permettant aux membres de l'équipe d'évaluer rapidement la bonne implémentation de la fonctionnalité décrite par le ticket. Certaines user story sont décomposées en **sous-tâches** servant à mieux répartir le travail et séparer les différentes fonctionnalités logicielles.

Chaque tâche ou sous-tâche est associée ensuite à une **branche** sur **Github**, ce qui nous permet de lier Github et Jira: lorsqu'une tâche est finie, une **pull request** est envoyée pour être passée en revue par l'autre personne du binôme. Une fois la **review** finie, les **commentaires** envoyés et la pull request acceptée, la branche peut être **merge** dans la branche principale (à condition qu'il n'y ait pas de conflits et que les tests passent!). Nos différentes branches sont présentées en annexe (1,2).

La production de ces user stories a constitué un **Backlog** que nous avons ensuite trié par **priorité** des user stories, en prenant le rôle de **product owner**, dans le cadre du **project kick-off**. Ce tri nous permet ensuite d'évaluer quelles fonctionnalités sont les plus urgentes à implémenter ; nous avons ainsi préparé un premier **Sprint**.

A partir de la fin de la première phase du projet, nous avons lancé un nouveau sprint toutes les deux semaines, puis en fin de projet un par semaine. Le sprint planning a été fait à chaque fin de sprint, après le **sprint review** et un **retrospective meeting**, en se concertant sur ce qui avait été fait et ce qui restait à réaliser. Parfois, des tickets étaient en cours sur plusieurs Sprints car nos sprints étaient trop courts par rapport au temps que nous pouvions accorder au projet ou nous avons mal évalué le temps alloué à chaque ticket.

En début de chaque séance de travail (nous avons pour la majorité organisé des séances de travail ensemble), nous avons pris le temps de retourner sur le Jira, pour simuler un **scrum daily meeting** : chaque membre du binôme présentait l'état d'avancement dans ses missions prévues dans le sprint ouvert. Des problèmes ou questions émergeaient qui étaient résolues avant de continuer chacun l'avancement dans nos missions précises.

Les résultats que nous avons eu sur Jira sont présentés en annexe (3, 4).

3. Intégration continue

Nous avons appris également par ce projet à utiliser l'**intégration continue automatisée** grâce à divers outils.

La compilation de notre projet, ainsi que la production d'artéfact .jar sont automatisées à l'aide de **Maven**. Les tests réalisés sont des tests unitaires basés sur le framework **JUnit**. Etant donné que le projet est de réaliser une première version "à étendre" de l'application, nous avons essayé de couvrir le maximum du code avec des tests JUnit, ce qui nous donne un **coverage** global moyen de ~70% du code.

Nous avons utilisé les **GitHub Actions** sur 2 workflows.

Le premier teste chacune de nos pull request sur Github : il impose le set up de la JDK 17, puis il compile et teste (tests unitaires JUnit) avec Maven. La pull request peut être acceptée seulement si Maven compile et passe tous les tests, vérifications dont s'occupent les GitHub Actions.

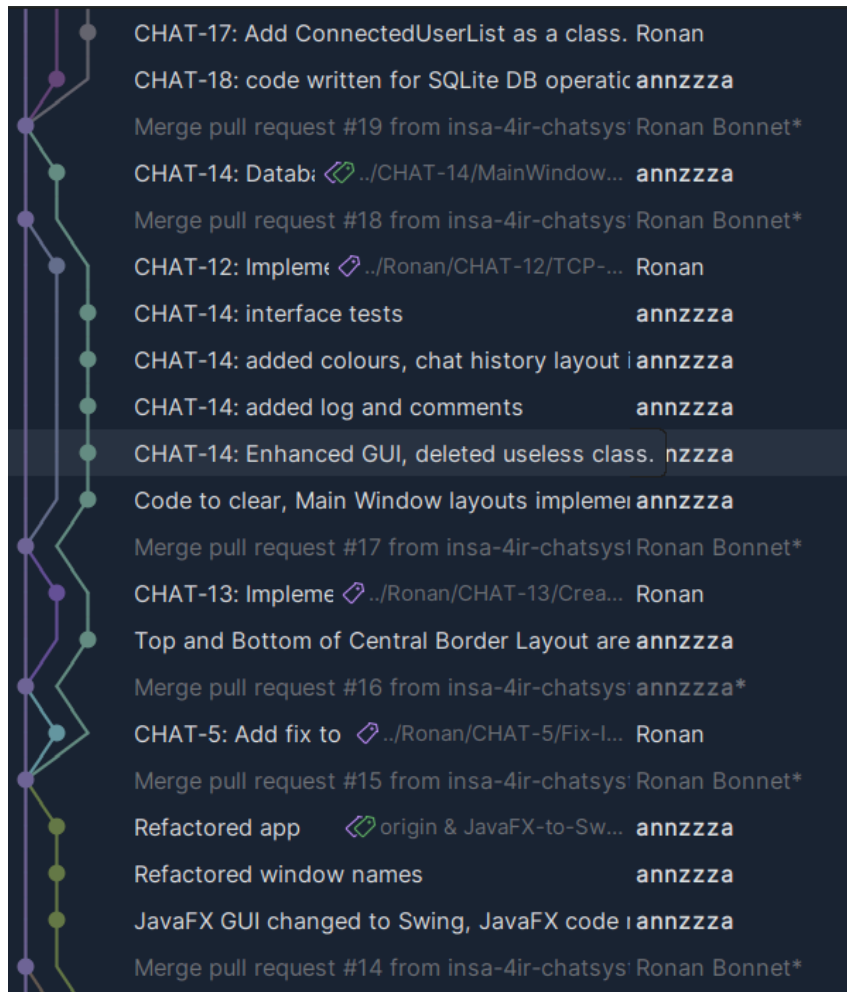
Le deuxième pour déployer le .jar directement sur GitHub. Il s'effectue lorsque nous mettons le tag "[release]" dans le message d'une pull request qui va sur la branche main. Il récupère la version du projet directement dans la configuration Maven. Ainsi un utilisateur lambda peut directement télécharger le logiciel et le lancer sans avoir à se préoccuper de la compilation.

Annexes

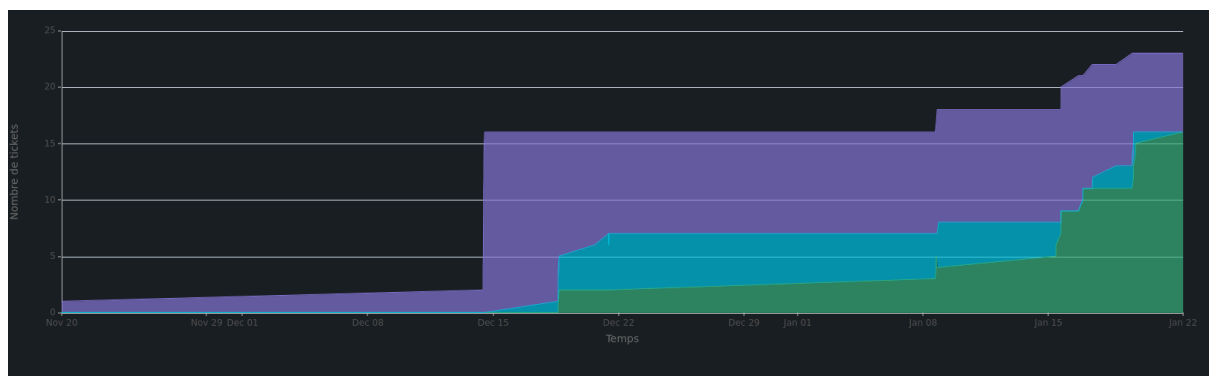
1. Branches Github

Branch	Updated	Check status	Behind	Ahead	Pull request
main	1 hour ago	1 / 2		Default	#1
CHAT-22/Exception-handling	1 hour ago	1 / 2	10	0	#26
Ronan/simplify-main	1 hour ago	1 / 2	6	0	#25
Remove-LocalDatabase	3 days ago	1 / 1	16	0	#24
CHAT-19/link-database	3 days ago	1 / 1	25	0	#23
Ronan/CHAT-21/Fix-discovery	last week	1 / 1	31	0	#22
Ronan/CHAT-17/Message-exchange	last week	1 / 1	33	0	#21
CHAT-18/DAOImplementation	last week	1 / 1	42	0	#20
Anna/CHAT-14/MainWindowGUI	2 weeks ago	1 / 1	59	0	#19
Ronan/CHAT-12/TCP-Client	2 weeks ago	1 / 1	61	0	#18
Ronan/CHAT-13/Create-TCP-Server	last month	1 / 1	63	0	#17
Ronan/CHAT-5/Fix-IP-problems-w...	last month	1 / 1	65	0	#16
JavaFX-to-Swing	last month	1 / 1	69	0	#15
review	last month	1 / 1	74	0	#14
cleaning&javadoc	last month		72	0	
UDPCClient-update	2 months ago	1 / 1	106	0	
feedback	3 months ago		151	0	

2. Log graph (partiel)



3. Jira Cumulative Flow Diagram



4. Jira Rapport de Burnup -Sprint 3

