

Présentation détaillé du projet

Application de collecte d'informations sur les machines d'infrastructure

by

Théo Da Conceicao

Thomas Chevalier

Alban Burlot

Lucas Bunel

Ulysse Parmentier

Sommaire

I - Introduction.....	3
II - Méthode Agile - Scrum.....	4
II - I - Présentation du Product Backlog.....	4
II - II - Création des Sprint.....	5
III - Outils de gestion de projet.....	7
III - I - GIT.....	7
III - II - Gestion des tâches.....	9
IV - Présentation de l'application.....	10
ANNEXE.....	11

I - Introduction

Le présent cahier des charges expose en détail les spécifications du projet, dédié au développement d'une application Python destinée à la collecte d'informations sur les composants d'une infrastructure, englobant les réseaux, serveurs, machines virtuelles, et autres éléments. Ce projet a été entrepris par notre équipe de cinq personnes, avec une allocation totale de 16 heures de travail. Le projet a débuté le 12 décembre et s'est terminé le 2 février.

Dans le cadre de ce projet, l'approche adoptée repose sur les méthodologies agiles, favorisant un processus de développement itératif et collaboratif. L'équipe s'engage à suivre un cycle de travail dynamique, permettant des ajustements continus en fonction des retours et des évolutions requises tout au long du processus de développement.

L'application Python envisagée sera conçue de manière à garantir une collecte d'informations efficace et précise sur les divers composants de l'infrastructure ciblée. Le recueil de données inclura des détails spécifiques sur les machines, les configurations réseau, les serveurs et les machines virtuelles, afin de fournir une vue complète de l'environnement informatique considéré.

En déployant une approche Agile - scrum, nous pouvons diviser notre travail en 3 sprint pour répartir équitablement les temps de travail sur les périodes allouées au projet. Il est important de noter qu'avec le volume horaire nous n'aurons pas le temps de mener à terme les sprint 2 & 3.

II - Méthode Agile - Scrum

La méthode Scrum, axée sur l'agilité, repose sur deux concepts fondamentaux : le Product Backlog et les Sprints. Le Product Backlog est une liste dynamique des fonctionnalités à développer, priorisées en fonction de la valeur ajoutée pour le produit final. Les Sprints, sont des cycles de développement courts (habituellement de deux à quatre semaines), permettent de se concentrer sur des objectifs spécifiques du Product Backlog. Cette approche itérative offre une gestion de projet flexible, favorisant la livraison continue de fonctionnalités tout en permettant des ajustements rapides en fonction des besoins évolutifs.

II - I - Présentation du Product Backlog

Initialement, nous amorçons notre projet par la rédaction du Product Backlog et des user story en alignement avec les exigences du cahier des charges. Cette démarche englobe divers éléments, parmi les suivants :

- Les besoins exprimés par le client, mettant en avant les fonctionnalités essentielles et les attentes spécifiques.
- Les technologies employées, en considérant les contraintes et les possibilités offertes par ces outils dans le développement de chaque fonctionnalité.
- Les options que le client souhaite intégrer dans l'application, permettant de personnaliser et d'adapter le produit final à ses préférences.
- L'établissement de priorités pour définir l'ordre d'implémentation des différentes user stories, assurant ainsi la satisfaction des besoins cruciaux en premier.
- La détermination claire des critères d'acceptation, définissant les conditions spécifiques qui indiqueront la complétion satisfaisante de chaque user story.

En harmonisant ces éléments, nous nous assurons d'articuler des user stories qui capturent de manière globale les attentes du client, tout en intégrant les considérations technologiques, les options souhaitées, les priorités, et les critères d'acceptation indispensables à la réussite du projet.

Voici une partie de notre Product Backlog content : les EPICS, les user story, les difficultés ainsi que les tâches et le temps estimé.

Epics	Story	CLXC	Taches	Temps
Interface utilisateur	L'utilisateur veut un tableau de bord intuitif pour la visualisation des données.	5	Design du tableau de bord	1.5
			Choix des éléments de visualisation	0.5
			Tests d'ergonomie	0.5
	L'utilisateur visualise des données mises à jour en temps réel sur le tableau de bord.	8	Intégration de flux de données en temps réel	1.5
			Mise en place de la mise à jour automatique	0.5
			Tests de performance	0.5
	L'utilisateur se connecte de manière sécurisée à l'interface utilisateur.	2	Mise en place d'un système d'authentification	0.5
			Tests de sécurité	0.5
			Configuration SSL/TLS	0.5
Collecte des informations	L'utilisateur doit pouvoir accéder à des informations collectées de manière fiable.	5	Développement du module de collecte d'informations	1.5
			Tests de fiabilité	0.5
			Vérification des sources de données	0.5
	Les informations collectées doivent être enregistrer en base de donnée. (Informations utilisateurs pour la connection comprise)	6	Conception de la base de données	2
			Intégration avec le système de collecte	1
			Tests de sauvegarde et récupération	0.5
	L'application récupère des informations sur divers systèmes d'exploitation UNIX.	13	Adaptation du module pour différents OS UNIX	2.5
			Tests multi-plateformes	1
			Documentation spécifique OS	1

Figure 1 : Product backlog

Vous pourrez retrouver en annexe le Product Backlog complet cf. ANNEXE 1

II - II - Création des Sprint

Dans notre cas, nous prévoyons de diviser le Product Backlog en trois sprints distincts, chacun représentant une période de développement spécifique. Cette approche permettra de planifier et d'organiser nos efforts de manière à atteindre des objectifs concrets à chaque étape. En envisageant cette division, nous avons pris en compte notre vélocité, c'est-à-dire la capacité de l'équipe à livrer un certain volume de travail pendant un sprint donné.

Il est important de préciser que nous avons intégré les rédactions des livrables dans notre sprint 1.

Rappel : contenue de notre vélocité et du temps alloué au projet nous n'effectuons pas les 3 sprint dans leurs intégralité.

Sprint 1 :

Epics	Story	CLXC	Taches	Temps
Interface utilisateur	L'utilisateur veut un tableau de bord intuitif pour la visualisation des données.	5	Design du tableau de bord	1,5
			Choix des éléments de visualisation	0,5
			Tests d'ergonomie	0,5
	L'utilisateur visualise des données mises à jour en temps réel sur le tableau de bord.	8	Intégration de flux de données en temps réel	1,5
			Mise en place de la mise à jour automatique	0,5
			Tests de performance	0,5
	L'utilisateur se connecte de manière sécurisée à l'interface utilisateur.	2	Mise en place d'un système d'authentification	0,5
			Tests de sécurité	0,5
			Configuration SSL/TLS	0,5
Collecte des informations	L'utilisateur doit pouvoir accéder à des informations collectées de manière fiable.	5	Développement du module de collecte d'informations	1,5
			Tests de fiabilité	0,5
			Vérification des sources de données	0,5
	Les informations collectées doivent être enregistrer en base de donnée. (Informations utilisateurs pour la connexion comprise)	8	Conception de la base de données	2
			Intégration avec le système de collecte	1
			Tests de sauvegarde et récupération	0,5
	L'application récupère des informations sur divers systèmes d'exploitation UNIX.	13	Adaptation du module pour différents OS UNIX	2,5
			Tests multi-plateformes	1
			Documentation spécifique OS	1
Rédaction des livrables	Le Product Owner souhaite des livrables	3	Documentation du code source	0,5
			Manuel d'utilisation	0,5
			Rapport de test	0,5
			Présentation détailler du projet	0,5
Vélocité de 40		39		16,5

Figure 2 : Sprint 1

Vous pourrez retrouver en annexe le Product Backlog complet cf.ANNEXE 2 & 3

Un autre élément clé des sprints réside dans notre pratique consistant à reporter toutes les tâches inachevées vers le sprint suivant. Cette approche nous assure de couvrir l'intégralité du projet de manière progressive, sans négliger aucune fonctionnalité. En transférant les travaux non achevés d'un sprint à l'autre, nous nous engageons à maximiser notre efficacité et à garantir que chaque aspect du projet soit pris en compte dans le cadre de nos itérations successives. C'est en autre pourquoi nous avons décidé de varier les complexité d'un sprint à l'autre.

III - Outils de gestion de projet

Étant donné les contraintes de temps auxquelles nous avons été confrontés lors de la réalisation de ce projet, il était impératif d'adopter une gestion de projet efficace dès le départ. Malgré des délais serrés, il était crucial de mettre en place une planification minutieuse, une allocation judicieuse des ressources et une coordination étroite entre les membres de l'équipe pour maximiser notre productivité et atteindre nos objectifs dans les délais impartis. Une gestion de projet efficiente nous a permis de prioriser les tâches, de répartir efficacement les responsabilités et de maintenir un suivi rigoureux de l'avancement du projet, garantissant ainsi une utilisation optimale du temps disponible.

Pour profiter au maximum de ce temps nous avons utilisé certains outils

III - I - GIT

GitHub, intégré à la méthode Scrum, simplifie la collaboration en permettant à plusieurs membres de travailler simultanément sur le code. Les fonctionnalités telles que les "pull requests" facilitent les revues de code, garantissant une meilleure qualité avant l'intégration. La visibilité sur l'historique des modifications, la gestion intégrée des problèmes, et la possibilité d'automatiser les tests contribuent à une gestion efficace du projet. GitHub joue un rôle clé dans la mise en œuvre réussie de la méthode Scrum en favorisant la collaboration, la transparence et la livraison continue.

Voici le contenu de notre GIT, nous pouvons retrouver le cahier des charges qui nous à été transmise par le Product Owner ainsi que des fichiers permettant de mettre en place notre environnement de développement.

Il est important de comprendre qu'un repository GIT se présente comme l'arborescence d'un ordinateur classique ce qui facilite la gestion des fichiers.

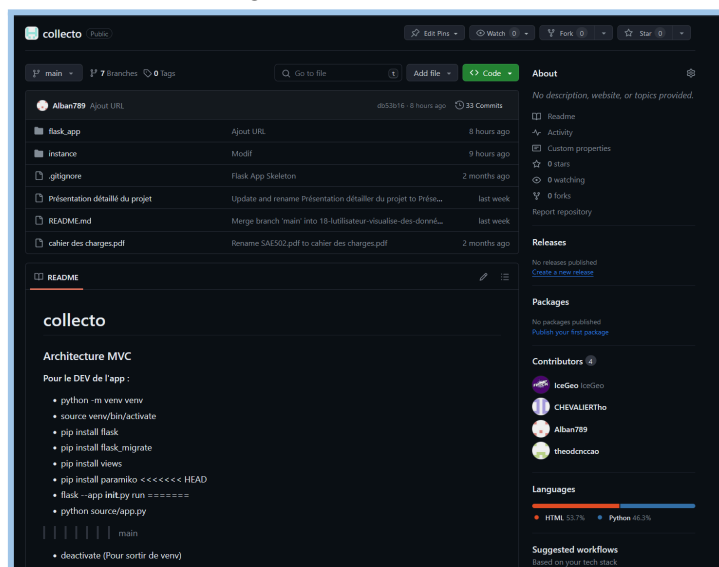


Figure 3 : Présentation du GIT

Dans notre contexte, nous avons généré plusieurs "issues" pour représenter les différentes tâches à accomplir. Chaque membre de l'équipe a la possibilité de choisir une tâche spécifique et de créer une branche dédiée, permettant ainsi le développement indépendant de son code sans perturber le travail déjà réalisé par les autres membres de l'équipe.

Nos issues représente les différentes storys de nos sprints :

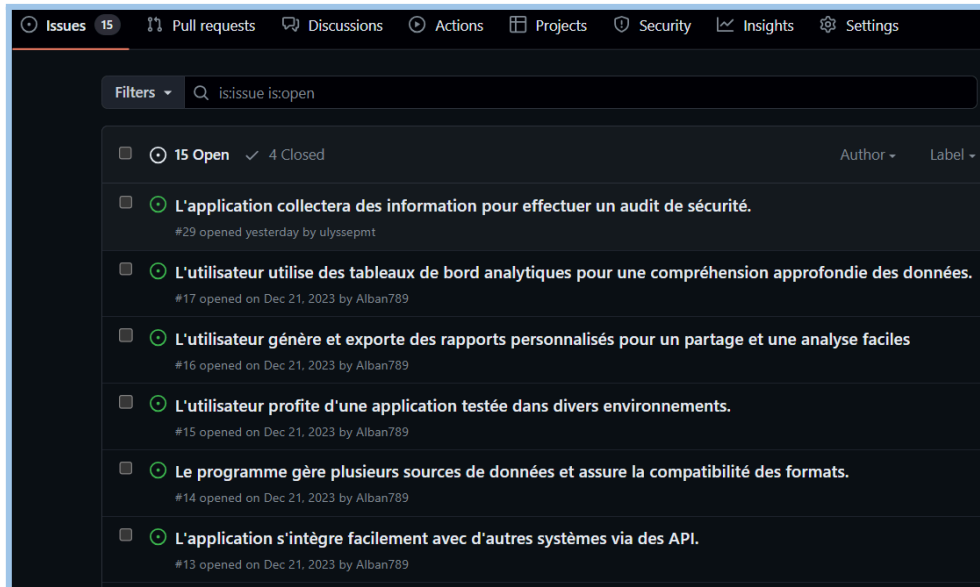


Figure 4 : Présentation des issues

L'un des bénéfices clés des issues de Git réside dans leur capacité à offrir un suivi transparent de l'avancement. En illustrant ce point, nous avons une issue récapitulant tous les "commits" effectués, accompagnés de leurs descriptions et des noms des membres de l'équipe ayant contribué à ces modifications. Cette fonctionnalité permet une gestion simplifiée de l'évolution du projet en fournissant une traçabilité claire des contributions individuelles. Il est également possible de définir les statuts d'avancement de la tâche.

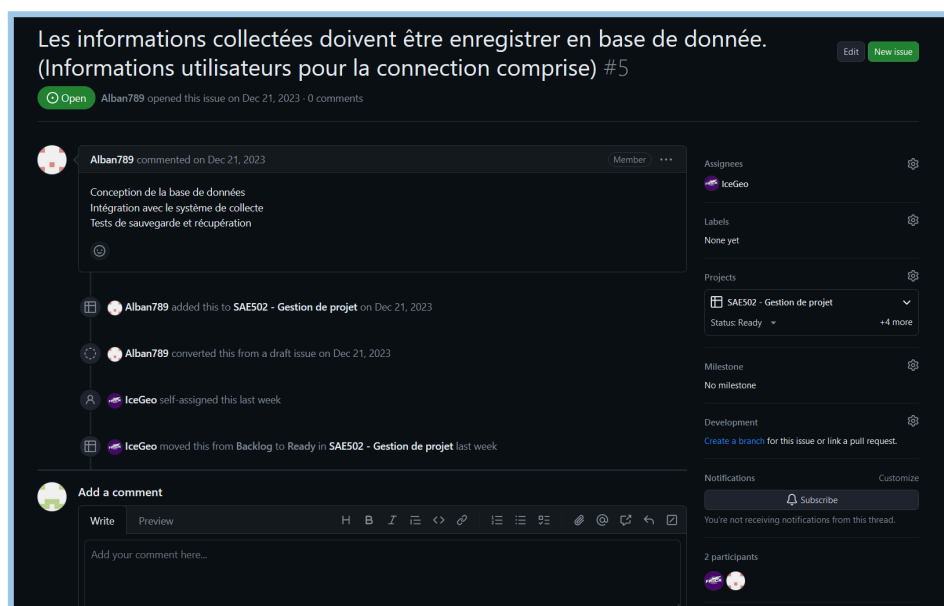


Figure 5 : Présentation d'une issue

III - II - Gestion des tâches

Pour la gestion des tâches, Git propose un outil similaire à Trello, offrant des fonctionnalités équivalentes telles que l'assignation des tâches, la définition des échéances, la possibilité de commenter et de suivre l'avancement de chaque tâche. L'interface présente une ressemblance marquée avec celle de Trello.

Nous avons donc toutes nos tâches répertoriées sur notre GIT :

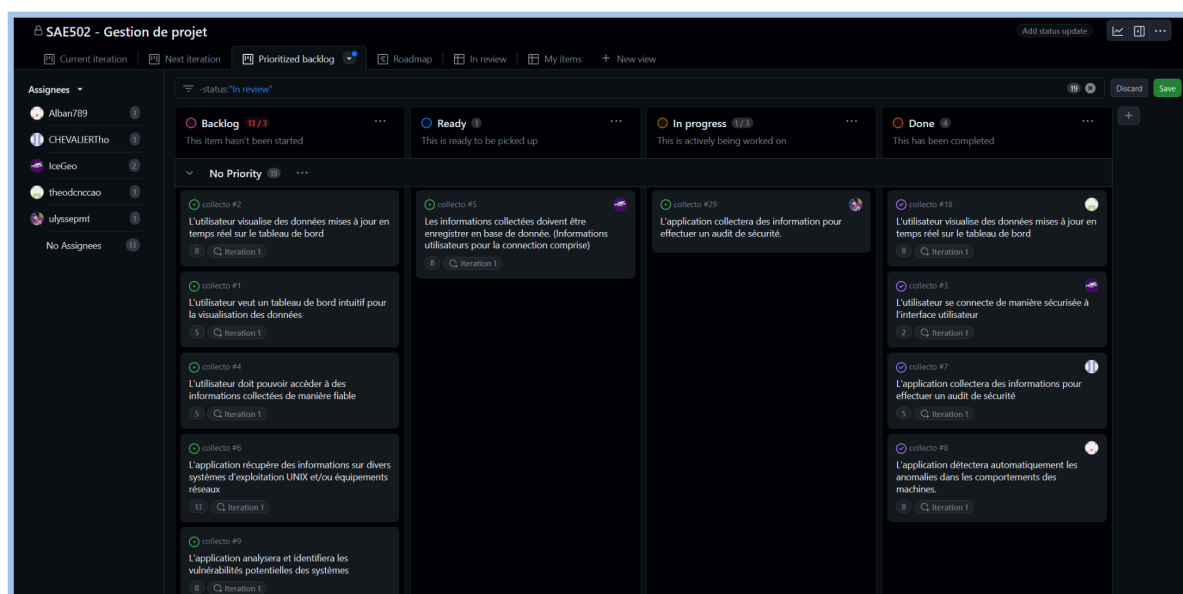
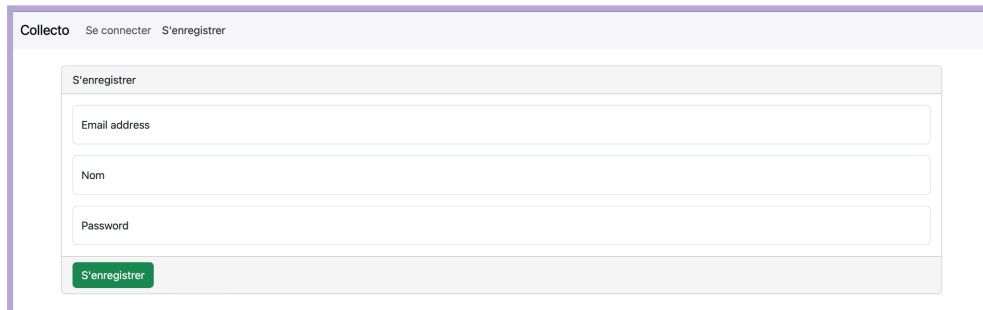


Figure 6 : Présentation de la gestion des tâches

Lorsqu'une issue est déclarée comme terminée, GIT modifie directement notre "Trello" pour signifier que la tâche est terminée.

IV - Présentation de l'application

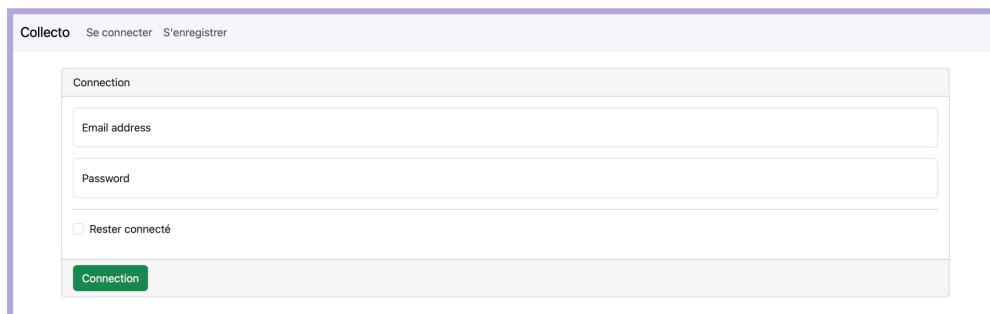
Une fois arrivé sur le site, il est nécessaire de procéder à l'inscription en remplissant le formulaire dédié



The registration form is titled 'S'enregistrer' and is part of the 'Collecto' application. It includes a navigation bar with 'Collecto', 'Se connecter', and 'S'enregistrer'. The form has three input fields: 'Email address', 'Nom', and 'Password'. A green 'S'enregistrer' button is at the bottom.

Figure 7 : Page d'inscription

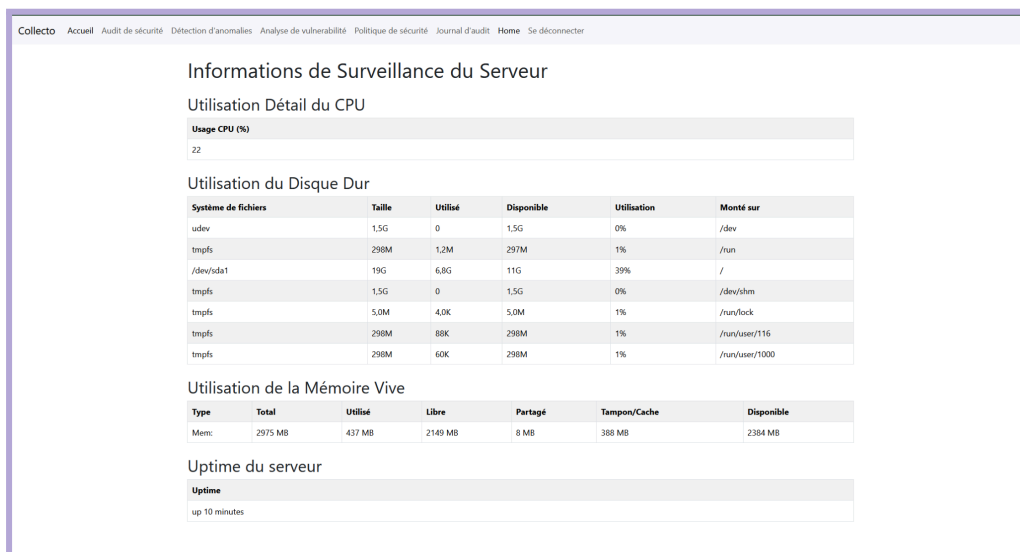
Et par la suite on peut se connecter.



The connection form is titled 'Connection' and is part of the 'Collecto' application. It includes a navigation bar with 'Collecto', 'Se connecter', and 'S'enregistrer'. The form has two input fields: 'Email address' and 'Password'. There is a checkbox for 'Rester connecté' and a green 'Connection' button at the bottom.

Figure 8 : Page de connexion

Une fois connecté sur la page d'accueil du site, nous pouvons observer diverses informations, comme l'utilisation du CPU, du disque, de la mémoire vive et l'uptime du serveur.



The dashboard is titled 'Informations de Surveillance du Serveur' and is part of the 'Collecto' application. It includes a navigation bar with 'Collecto', 'Accueil', 'Audit de sécurité', 'Détection d'anomalies', 'Analyse de vulnérabilité', 'Politique de sécurité', 'Journal d'audit', 'Home', and 'Se déconnecter'. The dashboard displays four sections:

- Utilisation Détail du CPU**: A table showing 'Usage CPU (%)' with a value of 22.
- Utilisation du Disque Dur**: A table showing disk usage for various file systems.
- Utilisation de la Mémoire Vive**: A table showing memory usage for different types of memory.
- Uptime du serveur**: A table showing the server's uptime.

Figure 9 : Page d'accueil

Via la barre de navigation, on peut se rendre sur différente page pour observer les remontées du serveur. Et ainsi détecter une anomalie.

ANNEXE

Annexe 1 : Product Backlog

Projet : Application de collecte d'informations sur les machines d'infrastructure				
Epics	Story	CLXC	Tâches	Temps
Interface utilisateur	L'utilisateur voit un tableau de bord intuitif pour la visualisation des données.	5	Design du tableau de bord	1,5
			Choix des éléments de visualisation	0,5
			Tests d'ergonomie	0,5
	L'utilisateur visualise des données mises à jour en temps réel sur le tableau de bord.	8	Intégration de flux de données en temps réel	1,5
			Mise en place de la mise à jour automatique	0,5
			Tests de performance	0,5
	L'utilisateur se connecte de manière sécurisée à l'interface utilisateur.	2	Mise en place d'un système d'authentification	0,5
			Tests de sécurité	0,5
			Configuration SSL/TLS	0,5
Collecte des informations	L'utilisateur doit pouvoir accéder à des informations collectées de manière fiable.	5	Développement du module de collecte d'informations	1,5
			Tests de fiabilité	0,5
			Vérification des sources de données	0,5
	Les informations collectées doivent être enregistrées en base de données. (Informations utilisateurs pour la connexion complète)	8	Conception de la base de données	2
			Intégration avec le système de collecte	1
			Tests de sauvegarde et récupération	0,5
	L'application récupère des informations sur divers systèmes d'exploitation UNIX.	13	Adaptation du module pour différents OS UNIX	2,5
			Tests multi-plateformes	1
			Documentation spécifique OS	0,5
CyberSecurite	L'application collectera des informations pour effectuer un audit de sécurité	5	Scan des ports ouverte	0,5
			Scan des processus actifs	0,5
			Utilisation du disque	0,5
	L'application détectera automatiquement les anomalies dans les comportements des machines.	8	Récupérer les fichiers de logs de la machine	0,5
			Collecte des données	1
			Utilisation de package pour la détection des anomalies	1
	L'application analysera et identifiera les vulnérabilités potentielles des systèmes	8	Mise en place de seuils d'alerte	0,5
			Notification et rapport d'anomalie	1
			Evolution des systèmes existants	1
	L'application surveillera l'état des certificats SSL/TLS pour assurer une communication sécurisée	5	Analyse des vulnérabilités connues	1
			Simulation d'attaques virtuelles	1
			Recommandations de sécurité	0,5
	L'application évaluera la conformité des systèmes aux politiques de sécurité établies.	5	Récupérer l'état des certificats TLS/SSL	0,5
			Automatiser la vérification de la validité / expiration d'un certif	1
			Notifier l'utilisateur quand un certificat est non valide / expiré	0,5
	L'application enregistrera un journal d'audit pour une traçabilité complète des activités	5	Définition des politiques de sécurité	0,5
			Analyse des configurations système	1
			Identification des écarts de conformité	0,5
Intégration et Compatibilité	L'application s'intègre facilement avec d'autres systèmes via des API.	13	Recommandations de conformité	0,5
			Définition des éléments à auditer	0,5
			Implémentation du journal d'audit	1
	Le programme gère plusieurs sources de données et assure la compatibilité des formats.	3	Stockage sécurisé des journaux d'audit	0,5
			Consultation et analyse des journaux d'audit	0,5
Reporting et Analyse	L'utilisateur profite d'une application testée dans divers environnements.	8	Développement d'API pour l'intégration	2,5
			Documentation des API	1
			Tests d'intégration	0,5
	L'utilisateur génère et exporte des rapports personnalisés pour un partage et une analyse faciles.	3	Gestion des formats de données	1
			Adaptation à différentes sources	1
			Tests de compatibilité des données	0,5
Rédaction des livrables	L'utilisateur utilise des tableaux de bord analytiques pour une compréhension approfondie des données.	5	Tests multi-environnements	1,5
			Optimisation pour différents systèmes	1
			Documentation d'installation	1
	Le Product Owner souhaite des livrables	3	Outils de génération de rapports	0,5
			Fonctionnalités d'exportation	0,5
			Tests d'utilisabilité	0,5
Total	Le Product Owner souhaite des livrables	3	Développement de tableaux de bord analytiques	1
			Intégration de fonctionnalités d'analyse	1
			Formation utilisateur	0,5
	Total	112	Documentation du code source	0,5
			Manuel d'utilisation	0,5
			Rapport de test	0,5
Total	Le Product Owner souhaite des livrables	3	Présentation détaillée du projet	0,5
	Total	112		
Total	Le Product Owner souhaite des livrables	3		
	Total	112		
Total	Le Product Owner souhaite des livrables	3		
	Total	112		
Total	Le Product Owner souhaite des livrables	3		
	Total	112		
Total	Le Product Owner souhaite des livrables	3		
	Total	112		
Total	Le Product Owner souhaite des livrables	3		
	Total	112		
Total	Le Product Owner souhaite des livrables	3		
	Total	112		
Total	Le Product Owner souhaite des livrables	3		
	Total	112		
Total	Le Product Owner souhaite des livrables	3		
	Total	112		
Total	Le Product Owner souhaite des livrables	3		
	Total	112		
Total	Le Product Owner souhaite des livrables	3		
	Total	112		
Total	Le Product Owner souhaite des livrables	3		
	Total	112		
Total	Le Product Owner souhaite des livrables	3		
	Total	112		
Total	Le Product Owner souhaite des livrables	3		
	Total	112		
Total	Le Product Owner souhaite des livrables	3		
	Total	112		
Total	Le Product Owner souhaite des livrables	3		
	Total	112		
Total	Le Product Owner souhaite des livrables	3		
	Total	112		
Total	Le Product Owner souhaite des livrables	3		
	Total	112		
Total	Le Product Owner souhaite des livrables	3		
	Total	112		
Total	Le Product Owner souhaite des livrables	3		
	Total	112		
Total	Le Product Owner souhaite des livrables	3		
	Total	112		
Total	Le Product Owner souhaite des livrables	3		
	Total	112		
Total	Le Product Owner souhaite des livrables	3		
	Total	112		
Total	Le Product Owner souhaite des livrables	3		
	Total	112		
Total	Le Product Owner souhaite des livrables	3		
	Total	112		
Total	Le Product Owner souhaite des livrables	3		
	Total	112		
Total	Le Product Owner souhaite des livrables	3		
	Total	112		
Total	Le Product Owner souhaite des livrables	3		
	Total	112		
Total	Le Product Owner souhaite des livrables	3		
	Total	112		
Total	Le Product Owner souhaite des livrables	3		
	Total	112		
Total	Le Product Owner souhaite des livrables	3		
	Total	112		
Total	Le Product Owner souhaite des livrables	3		
	Total	112		
Total	Le Product Owner souhaite des livrables	3		

Annexe 2 : Sprint

Epics	Story	CLXC	Taches	Temps
Interface utilisateur	L'utilisateur veut un tableau de bord intuitif pour la visualisation des données.	5	Design du tableau de bord	1,5
			Choix des éléments de visualisation	0,5
			Tests d'ergonomie	0,5
	L'utilisateur visualise des données mises à jour en temps réel sur le tableau de bord.	8	Intégration de flux de données en temps réel	1,5
			Mise en place de la mise à jour automatique	0,5
			Tests de performance	0,5
	L'utilisateur se connecte de manière sécurisée à l'interface utilisateur.	2	Mise en place d'un système d'authentification	0,5
			Tests de sécurité	0,5
			Configuration SSL/TLS	0,5
Collecte des informations	L'utilisateur doit pouvoir accéder à des informations collectées de manière fiable.	5	Développement du module de collecte d'informations	1,5
			Tests de fiabilité	0,5
			Vérification des sources de données	0,5
	Les informations collectées doivent être enregistrer en base de donnée. (Informations utilisateurs pour la connection comprise)	8	Conception de la base de données	2
			Intégration avec le système de collecte	1
			Tests de sauvegarde et récupération	0,5
	L'application récupère des informations sur divers systèmes d'exploitation UNIX.	13	Adaptation du module pour différents OS UNIX	2,5
			Tests multi-plateformes	1
			Documentation spécifique OS	1
Rédaction des livrables	Le Product Owner souhaite des livrables	3	Documentation du code source	0,5
			Manuel d'utilisation	0,5
			Rapport de test	0,5
			Présentation détailler du projet	0,5
Vélocité de 40		39		16,5

Epics	Story	CLXC	Taches	Temps
Cybersécurité	L'application collectera des informations pour effectuer un audit de sécurité	5	Scan des ports ouverts	0.5
			Scan des processus actifs	0.5
			Utilisation du disque	0.5
			Récupérer les fichiers de logs de la machine	0.5
	L'application détectera automatiquement les anomalies dans les comportements des machines.	8	Collecte des données	1
			Utilisation de package pour la détection des anomalies	1
			Mise en place de seuils d'alerte	1
	L'application analysera et identifiera les vulnérabilités potentielles des systèmes	8	Notification et rapport d'anomalie	0.5
			Évaluation des systèmes existants	1
Analyse des vulnérabilités connues			1	
Reporting et Analyses	L'utilisateur génère et exporte des rapports personnalisés pour un partage et une analyse faciles	3	Simulation d'attaques virtuelles	1
			Recommandations de sécurité	0.5
			Outils de génération de rapports	1
	L'utilisateur utilise des tableaux de bord analytiques pour une compréhension approfondie des données.	5	Fonctionnalités d'exportation	1
			Tests d'utilisabilité	0.5
			Développement de tableaux de bord analytiques	1
			Intégration de fonctionnalités d'analyse	1
Vélocité de 40		29		14

Epics	Story	CLXC	Taches	Temps	
Intégration et Compatibilité	L'application s'intègre facilement avec d'autres systèmes via des API.	13	Développement d'API pour l'intégration	2.5	
			Documentation des API	1	
			Tests d'intégration	0.5	
	Le programme gère plusieurs sources de données et assure la compatibilité des formats.	3	Gestion des formats de données	1	
			Adaptation à différentes sources	1	
			Tests de compatibilité des données	0.5	
	L'utilisateur profite d'une application testée dans divers environnements.	13	Tests multi-environnements	1.5	
			Optimisation pour différents systèmes	1	
			Documentation d'installation	1	
CyberSecurite	L'application surveillera l'état des certificats SSL/TLS pour assurer une communication sécurisée	8	Récupérer l'état des certificats TLS/SSL	0.5	
			Automatiser la vérification de la validité / expiration d'un certif	1	
			Notifier l'utilisateur quand un certificat est non valide / expiré	0.5	
	L'application évaluera la conformité des systèmes aux politiques de sécurité établies.	8	Définition des politiques de sécurité	0.5	
			Analyse des configurations système	1	
			Identification des écarts de conformité	0.5	
	L'application enregistrera un journal d'audit pour une traçabilité complète des activités	5	Recommandations de conformité	0.5	
			Définition des éléments à auditer	0.5	
			Implémentation du journal d'audit	1	
			Stockage sécurisé des journaux d'audit	0.5	
				Consultation et analyse des journaux d'audit	0.5
	Vélocité de 40		37		13