

Table des matières

I	Étude Préalable	iv
I.1	Présentation de l'organisme d'accueil	iv
I.1.1	Introduction	iv
I.1.2	Spécialité	iv
I.1.2.1	Développement Application web :	iv
I.1.2.2	Tierce Maintenance Applicative :	iv
I.2	Présentation du Projet :	iv
I.2.1	Intitulé :	iv
I.2.2	Monitoring System :	v
I.2.2.1	Définition :	v
I.2.2.2	Protocole de supervision :	v
I.2.2.3	Utilisation de supervision :	vi
I.2.2.4	Logiciels Existants :	vi
	Nagios :	vii
	Centreon :	ix
	Zabbix :	xi
I.2.3	Analyse des besoins :	xiv
I.2.3.1	Problématique :	xiv
I.2.3.2	Solution :	xiv
I.2.3.3	Identification des acteurs :	xiv
I.2.3.4	Identifiant des besoins :	xiv
	Besoins fonctionnels	xiv
	Besoins non fonctionnels	xv
	Diagramme de cas d'utilisations	xv
I.3	Méthodologie de développement :	xvi
I.3.1	Méthodes Agiles :	xvi
I.3.1.1	Définition :	xvi
I.3.1.2	Scrum :	xvii
	Conclusion	xix

Table des figures

I.1	Nagios	viii
I.2	centreon	x
I.3	Zabbix	xii
I.4	useCase	xv
I.5	useCase	xvi
I.6	Scrum	xix

Liste des tableaux

I.1	Tableau comparative	xiii
-----	-------------------------------	------

Chapitre I

Étude Préalable

Introduction :

Ce chapitre introductif a comme objectif de mettre notre travail dans son contexte général. Tout d'abord, nous commençons par faire une présentation de l'organisme "Bepro Consulting" ou nous avons effectué notre stage, ensuite nous présentons le sujet de ce projet en détaillant son cadre et ses fonctionnalités.

I.1 Présentation de l'organisme d'accueil

I.1.1 Introduction

"Bepro Consulting" est une société tunisienne "off-shore" créée en 2007 spécialisée dans le développement des applications web, mobiles, conception de sites Web et diagnostics de dysfonctionnements

I.1.2 Spécialité

I.1.2.1 Développement Application web :

"Bepro Consulting" offre des applications autour de métier pour leur client, afin de rendre l'utilisateur plus performant sur ses tâches. Il est spécialisé dans les technologies web telles que Asp.net, c#, le cloud azure, SQL server, JQuery, JavaScript. . . se ressemble des ingénieurs expérimentés dans le développement d'applications web complexes et ont apporté le savoir-faire codé selon les bonnes pratiques, on respecte les standards du web et la sécurité des données et utilisateurs.

I.1.2.2 Tierce Maintenance Applicative :

"Bepro Consulting" garantit la réactivité d'un support technique en cas de bug logiciel ou d'actions à effectuer à distance, faisant évoluer les systèmes selon ces nouvelles versions logicielles, effectue la gestion des bases de données et apporte leur expertise en diagnostiquant les anomalies. Un contrat d'assistance technique permet également de développer des nouvelles fonctionnalités du logiciel de monitoring.

I.2 Présentation du Projet :

I.2.1 Intitulé :

Le sujet consiste à concevoir et développer une application de « Monitoring System » permettant à l'administrateur de contrôler l'état des serveurs pour détecter à

tout moment une défaillance ou panne et alerter immédiatement l'administrateur en lui envoyant un mail, aussi elle permettra de gérer les équipements, de planifier les tâches qui devront être par la suite exécutées sur les différents équipements, de stocker les alertes envoyées, les tâches exécutées (les événements déroulés). En plus elle devra pouvoir gérer les différents utilisateurs qui vont la manipuler.

I.2.2 Monitoring System :

I.2.2.1 Définition :

"**Monitoring**" ou supervision désigne l'ensemble des systèmes et technologies mis en place dans le but de veiller au bon fonctionnement de votre système informatique d'entreprise. C'est une mesure préventive de cybersécurité qui permet d'éviter l'interruption de votre activité due aux pannes informatiques ou aux attaques.

Un tel processus est réalisé à plusieurs niveaux d'un parc de machines : Au niveau interconnexions (Réseau), au niveau de la machine elle-même (Système) et au niveau des services offerts par cette machine (Applications).

- **Supervision réseau** Par le terme réseau on entend ici l'aspect communication entre les machines. Le rôle est de s'assurer du bon fonctionnement des communications et de la performance des liens (débit, latence, taux d'erreurs). C'est dans ce cadre que l'on va vérifier par exemple si une adresse IP est toujours joignable, ou si tel port est ouvert sur telle machine, ou faire des statistiques sur la latence du lien réseau.
- **Supervision système** La surveillance se cantonne dans ce cas à la machine elle-même et en particulier ses ressources. Si l'on souhaite par exemple contrôler la mémoire utilisée ou la charge processeur sur le serveur voire analyser les fichiers de logs système.
- **Supervision applicative** Cette technique est plus subtile, c'est elle qui va nous permettre de vérifier le fonctionnement d'une application lancée sur une machine. Cela peut être par exemple une tentative de connexion sur le port de l'application pour voir si elle retourne ou demande bien les bonnes informations, mais aussi de l'analyse de logs applicatifs. En effet rien ne garantit qu'un port X ouvert veut dire que l'application qui tourne derrière n'est pas "plantée".

I.2.2.2 Protocole de supervision :

Il existe des protocoles réseau qui permettent de récupérer des informations sur le parc informatique. Nous allons en étudier deux particulièrement importants qui possèdent des rôles très différents mais qui ont un point en commun : Ils sont tout deux largement utilisés par les logiciels de supervision.

- **ICMP (Internet Control Message Protocol)** : ICMP est un protocole de couche réseau (couche 3 du modèle OSI) qui vient palier à l'absence de message d'erreur du protocole IP (Internet Protocol). En effet si il y a un incident de transmission les équipements intermédiaires vont utiliser ce protocole pour prévenir la machine émettrice. Les paquets ICMP sont encapsulés dans des paquets IP (malgré qu'ils soient au même niveau OSI), et peuvent contenir des bouts de paquets IP pour citer celui ayant généré l'erreur. Afin de catégoriser les erreurs, elles sont divisées

en types eux-mêmes parfois redécouverts en codes. Par exemple le type 3 représente un destinataire inaccessible : Il existe 16 codes différents en fonction de la raison pour laquelle le destinataire n'est pas joignable. C'est un protocole très simple, qui n'a pas pour fonction directe la supervision d'un réseau mais qui est utilisé comme source d'information sur la qualité du réseau ou sur la présence d'une machine.

- SNMP (Simple Network Management Protocol) :SNMP est un protocole de couche applicative qui a pour but de superviser les réseaux. Il a été conçu en 1988 par l'IETF (Internet Engineering Task Force) avec pour idée directrice de créer un protocole simple qui ne vienne pas gêner le trafic du réseau qu'il supervise. Depuis sa création, le protocole a évolué par soucis de sécurité : La version 2 qui est pour l'instant la plus utilisée possède une notion de communauté qui est utilisée comme un mot de passe, la version 3 durcit un peu plus le protocole en y ajoutant le chiffrement.

I.2.2.3 Utilisation de supervision :

on peut utiliser la supervision dans des différentes spécialités,dans :

- les hôpitaux et autres centres de recherche pour surveiller les malades, et étudier les réactions suite à la prise de certains médicaments ou suite à innovation sur certains traitements. On pose un monitoring à la maman, qui va accoucher pour surveiller et agir en cas d'urgence.
- le monde de la musique : pour séparer et isoler les sons, (entre graves, aigus, médiums) et bien faire le delta entre les enceintes actives et les enceintes passives ou bien les enceintes de proximité et les enceintes pour écouter à distance, mais le fond (ou le but visé) est de contrôler le flux, pour améliorer l'écoute de la musique enregistrée
- l'écologie : pour superviser certains systèmes informatifs ou systèmes d'observations, aussi bien dans le cadre animalier (migrations, évolutions des espèces) ou des écosystèmes tels que les grandes forêts, les océans, les pôles, le climat.
- L'électronique, et toutes les nouvelles technologies, pour apprécier les réactions, amplifiant et commandant les différents paramètres pour faire évoluer les tests et autres essais en grandeur nature. On fait du monitoring, lorsque certains essais répétitifs sont demandés, pour savoir, combien de temps cela va tenir.
- l'informatique, et là c'est probablement le domaine où le terme est le plus utilisé, pour contrôler, superviser, détecter, analyser les différents éléments qui composent par exemple un parc informatique. On fait du monitoring ou de la supervision, pour contrôler un parc informatique, ou un réseau de serveurs et être alerté, en cas de dysfonctionnement, panne ou anomalie, autrement dit en cas d'alerte qu'elle soit mineure ou majeure.

I.2.2.4 Logiciels Existants :

Dans cette partie nous allons représenter les principaux outils de supervision en dégageant leur fonctionnalités de base,avantage,inconvénient,nous nous intéressons à étudier les systèmes les plus utilisés :Nagios,Zabbix,Centreon...

Nagios :

(Anciennement Net saint) est un logiciel de supervision de réseaux créé en 1999 par Ethan Galstad. Il est considéré comme étant la référence des solutions de supervision Open Source. C'est un outil très complet pouvant s'adapter à n'importe quel type d'utilisation avec des possibilités de configuration très poussées. La modularité et la forte communauté (> 250 000) qui gravite autour de Nagios (en participant au développement de nombreux plugins et add-ons) offrent des possibilités en terme de supervision qui permettent aujourd'hui de pouvoir superviser pratiquement n'importe quelle ressource. C'est un programme modulaire qui se décompose en trois parties :

- Le moteur de l'application en charge d'ordonnancer les tâches de supervision l'interface web, qui permet d'avoir une vue d'ensemble du système d'information et des possibles anomalies les plugins, un ensemble de programmes que l'on peut compléter ou modifier en fonction des besoins de chacun pour superviser chaque service ou ressource disponible sur l'ensemble des ordinateurs ou éléments réseaux du si Offrant les possibilités suivantes :
- Superviser des protocoles réseaux : (SMTP, POP3, HTTP, NNTP, ICMP, SNMP, LDAP, etc.) superviser les ressources des serveurs (charge du processeur, occupation des disques durs, utilisation de la mémoire paginée) et ceci sur les systèmes d'exploitations les plus répandus superviser via le protocole SNMP, notamment pour des équipements réseaux (Switch, Firewall), la supervision à distance en utilisant par exemple SSH ou un tunnel SSL.
- Les plugins sont écrits dans les langages de programmation les plus adaptés à leur tâche : Scripts Shell (Bash, ksh, etc.) C++, Perl, Python, Ruby, PHP, C#, etc.
- La vérification des services se fait en parallèle, possibilité de définir une hiérarchie réseau pour différencier une panne serveur et un serveur injoignable, la notification d'alertes est entièrement paramétrable grâce à l'utilisation de plugins (alerte par email, SMS, etc.).
- Acquiescement des alertes par les exploitants de la supervision
- Gestion des escalades pour les alertes.
- Gestion des escalades pour les alertes.
 1. OK (tout va bien)
 2. WARNING(le premier seuil d'alerte est dépassé)
 3. CRITICAL(le second seuil d'alerte est dépassé ou alors le service a un problème)
 4. UNKNOWN (impossible de connaître l'état du service)

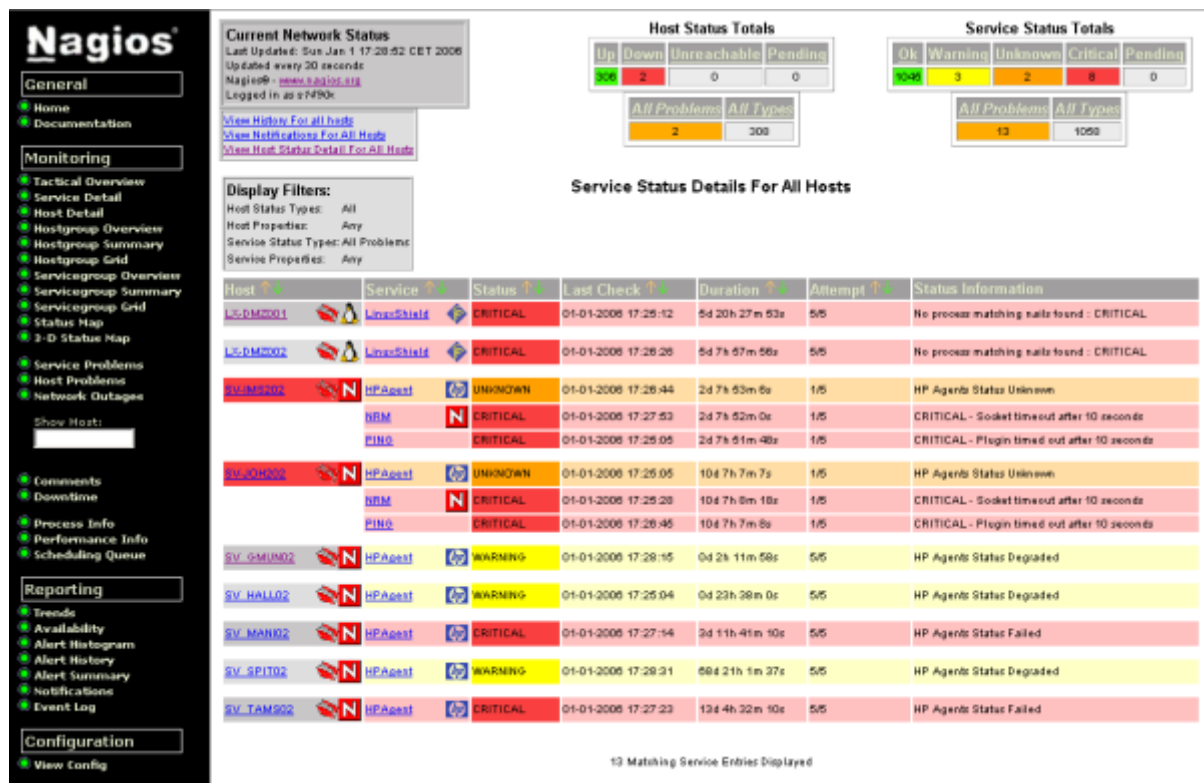


FIGURE I.1 – Nagios

● Avantages :

- Une très grande communauté qui participe activement au développement.
- Des plugins qui étendent les possibilités de Nagios.
- Solution complète permettant le reporting, la gestion des pannes et d'alarmes, gestion des utilisateurs
- Offre la possibilité de développer ses propres modules

● Inconvénients :

- Difficile à installer et à configurer
- Nagios dispose d'une interface austère
- Nagios n'affiche pas de graphes en natif
- Configuration fastidieuse via beaucoup de fichiers
- Pour avoir toute les fonctionnalités il faut installer des plugins, de base c'est assez limité.

Centreon :

(Anciennement Oréon), Créé en 2003 par des français souhaitant améliorer Nagios, il a été repris par une nouvelle entreprise nommée Merethis il se présente comme une évolution de celui-ci pour tout d’abord son interface mais aussi ses fonctionnalités. Il s’appuie également sur les technologies Apache et PHP pour l’interface web, MySQL pour le stockage des données de configuration et de supervision.

Les principales fonctionnalités de Centreon :

- Consultation de la métrologie et des informations d’état des services/hôtes supervisés
- Présentation mode “Dashboard” des informations de supervision à l’aide de widgets
- Gestion avancée des modèles de supervision
- Possibilité de définir la hiérarchie du réseau (relation parent/enfant) pour déterminer les hôtes arrêtés des hôtes non-joignables
- Mécanismes de surveillance de types actifs et passifs grâce au support de nombreux protocoles et agents de supervision
- Support du chiffrement des communications avec possibilité d’authentifier les flux par l’utilisation de certificats
- Configuration en ligne de commande via une API
- Support des architectures distribuées pour les parcs importants
- Déclenchement d’actions correctrices suivant les changements d’état des services hôtes
- Contrôle des droits d’accès utilisateurs via une gestion avancée des contrôles d’accès
- Gestion avancée des notifications (définition des périodes de surveillance et de remontés d’alertes, escalades, périodes de maintenances, prise en compte des dépendances de services pour limiter le nombre d’alertes)

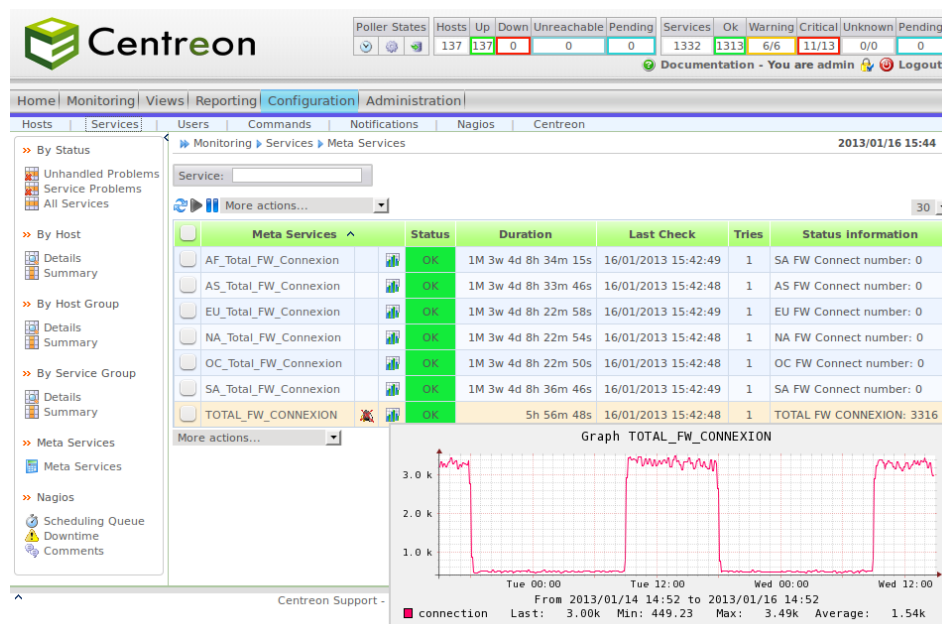


FIGURE I.2 – centreon

● Avantage :

- Une installation complète et automatique des packages nécessaires à l'utilisation de NAGIOS.
- Facilite la configuration de Nagios
- Une découverte automatique du réseau via NMAP.¹
- Graphe le résultat des alertes, système de reporting

● Inconvénient :

- Requier plus de ressources matérielles que Nagios
- L'interface peut paraître complexe car il existe beaucoup d'options
- Un développement qui n'est pas encore en phase avec celui de Nagios : Parfois des problèmes de compatibilité.
- Un peu plus lourd que du Nagios pur.

1. NMAP est un scanner de ports libre créé par Fyodor et distribué par Insecure.org. Il est conçu pour détecter les ports ouverts, identifier les services hébergés et obtenir des informations sur le système d'exploitation d'un ordinateur distant. Ce logiciel est devenu une référence pour les administrateurs réseaux car l'audit des résultats de Nmap fournit des indications sur la sécurité d'un réseau. Il est disponible sous Windows, Mac OS X, Linux, BSD et Solaris.

Zabbix :

C'est un outil de supervision, ambitionnant de concurrencer Nagios et MRTG². Il fait la supervision technique et applicative, offre des vues graphiques (générés par RRDtool) et des alertes sur seuil. C'est une solution de monitoring complète embarquant un front-end web, un ou plusieurs serveurs distribués, et des agents multiplateformes précompilés (Windows, Linux) il est également capable de faire du monitoring SNMP et IPMI ainsi que de la découverte de réseau. Il repose sur du C/C++, PHP pour la partie front end et MySQL/PostgreSQL/Oracle pour la partie BDD.

Principale fonctionnalités :

- une supervision répartie avec une administration web centralisée, afin que la récolte des données ne soit pas interrompu en cas de problème réseau
- une supervision de site web avec recherche de motif et scénarios de navigation
- un fonctionnement en actif et/ou passif, le serveur Zabbix va récupérer les données via des agents sur les éléments supervisés, et/ou, ce sont les éléments supervisés qui vont envoyer au serveur Zabbix les données
- une interface web pour une visualisation efficace de l'états des éléments réseaux et des données récoltées
- la notification par e-mail, messagerie instantanée, sms, et pratiquement n'importe quel autre moyen, pour être informé rapidement lors qu'un problème apparaît
- le calcul de SLA³, pour avoir une vue « commerciale » des services
- un agent haute performance en natif (Logiciel client pour Linux, Windows 2000, Windows 2003, Windows XP, Windows Vista)
- une supervision proactive, pour une interaction forte avec vos équipements (relance de services, extinction, reboot ...)
- la découverte automatique des serveurs et périphériques réseaux, pour surveiller les nouveaux serveurs dès que ceux-ci sont présents sur le réseau,

2. MRTG :Multi Router Traffic Grapher, Ce logiciel permet de surveiller et mesurer le trafic réseau. Il utilise le protocole SNMP pour interroger des équipements réseaux tels que des routeurs, commutateurs, ou bien encore serveurs, disposant d'une MIB.

3. SLA :La disponibilité est la durée pendant laquelle un service est disponible et opérationnel. La disponibilité est généralement la mesure la plus importante pour un site Web, un service en ligne ou un fournisseur basé sur le Web. Elle est exprimée en pourcentage tel que "99.9%". Par exemple, une disponibilité de 99,9% équivaut à 43 minutes et 50 secondes de temps mort.

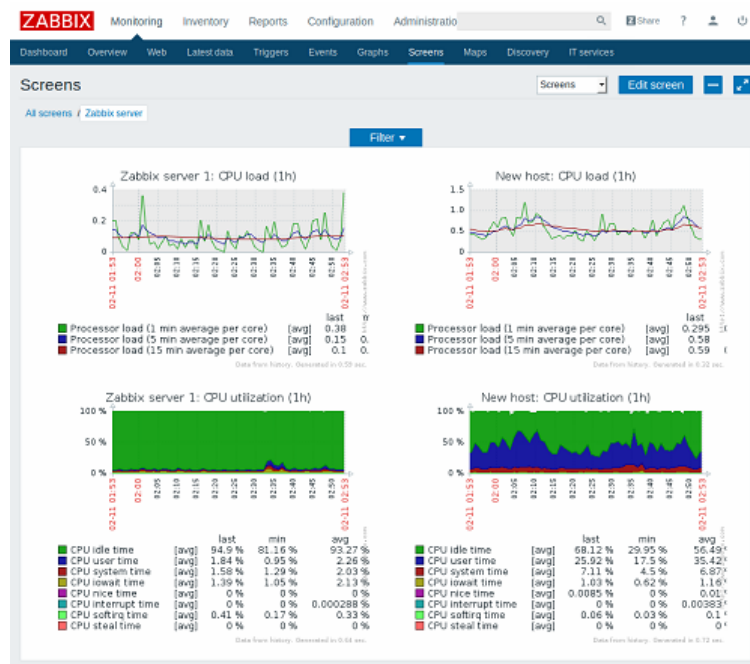


FIGURE I.3 – Zabbix

- **Avantages :**
 - Richesse des sondes et tests possibles (supervision d'applications Web, par exemple).
 - Réalisation de graphiques, cartes ou screens, Configuration par une interface graphique
 - Mise à jour de la configuration via l'interface Web de Zabbix.
 - Serveur Proxy Zabbix, Surveillances des sites web : temps de réponse, vitesse de transfert...
- **Inconvénient :**
 - Interface est un peu vaste, la mise en place des templates n'est pas évidente au début : petit temps de formation nécessaire.
 - L'agent zabbix communique par défaut en clair les informations, nécessité de sécuriser ces données (via VPN par exemple).
 - Peu d'interfaçage avec d'autres solutions commerciales.

Tableau comparative :

Critères fonctionnels	Nagios	Centreon	Zabbix
Environnement d'installation	Unix	Unix	Unis
Base de donnée	C++	PHP	PHP,C++,C
Protocole	SNMP,SMTP,ICMP	SNMP,SMTP,ICMP	HTTP,FTP,SMT
Gestion d'une authentification	Oui	Oui	Oui
Création des graphe simple à partir des mesure	Non	Oui	Oui
Utilisation d'agent sur machine cible	NSclient	NSclient	agent unix/windows
Utilisation et Configuration simple	Non	Oui	Oui
Intégration simple d'une nouvel host dans un système de centralise	Nom	Oui	Oui
Création des paragraphe complexe avec mise en relation des métriques des services supervisés	Nom	Nom	Oui
Possibilité de mettre en place simplement une supervision centralisé entre plusieurs sous-réseaux ou site	Non	Oui	Nom
Utilisation de gestion de base de données(RRD)	Non	Oui	Non

Tableau I.1 – Tableau comparative

I.2.3 Analyse des besoins :

I.2.3.1 Problématique :

Pour toute entreprise la stabilité de son système informatique est importante et obligatoire pour son développement. Cette stabilité est imposée à l'entreprise d'importants investissements matériels et humains ce qui n'est disponible pour toute société. Dès lors l'utilisation d'une solution logicielle permettant d'optimiser la réactivité des équipes techniques de maintenance devient très demandée.

I.2.3.2 Solution :

Pour répondre à cette problématique et malgré l'existence des logiciels libres (Nagios, Centreon, Zabbix...) qui sont difficiles à configurer et avec des interfaces complexes à comprendre. Nous proposons de développer une application web qui permet de superviser le système et qui dispose d'une interface claire permettant d'être réactif sur les indicateurs sélectionnés aussi d'être informé en temps réel grâce aux alertes personnalisées et pouvoir intervenir sur l'infrastructure à distance, en instantané, et réaliser la maintenance nécessaire sans le déranger.

I.2.3.3 Identification des acteurs :

- Administrateur : Un employé de la société ayant tous les droits d'accès.
- Technicien : Personne qui connaît et maîtrise une technique

I.2.3.4 Identifiant des besoins :

Besoins fonctionnels

- **Gestion d'utilisateur :**
 - Enregistrer des utilisateurs du système de surveillance (administrateur, technicien).
 - Donner les droits à l'utilisateur à utiliser certaines fonctionnalités ou un accès total au système.
 - Consulter la liste des utilisateurs du système par l'administrateur.
- **Gestion des équipements :**
 - Enregistrer des équipements et leurs caractéristiques.
 - Consulter la liste des équipements et leurs états.
- **Gestion des tâches :**
 - Lancer une nouvelle tâche planifiée ou non planifiée en utilisant un script.
 - Arrêter, annuler ou modifier une tâche existante.
 - Consulter la liste des tâches.
- **Gestion des alertes :**
 - Notification par mail et sms en cas de dysfonctionnement ou panne.
 - Enregistrer les rapports des anomalies du système

Besoins non fonctionnels

- **Fiabilité** :Il faut garantir la qualité de continu et le produit doit fonctionner correctement.
- **Facilité d'utilisation** :Il faut que l'interface du système simple et facile à utiliser.
- **Rapidité** :Le déplacement entre les pages doit être rapide et facile
- **Sécurité** :l'accès aux données doit être authentifier et autorisé par des moyens de sécurité.
- **Performance** :une application doit être avant tout performant c'est à dire à travers ses fonctionnalités, elle répond à toutes les exigences des usagers d'une manière optimale
- **Maintenabilité** :Le code de l'application est lisible et compréhensible afin d'assurer son état évolutif

Diagramme de cas d'utilisations

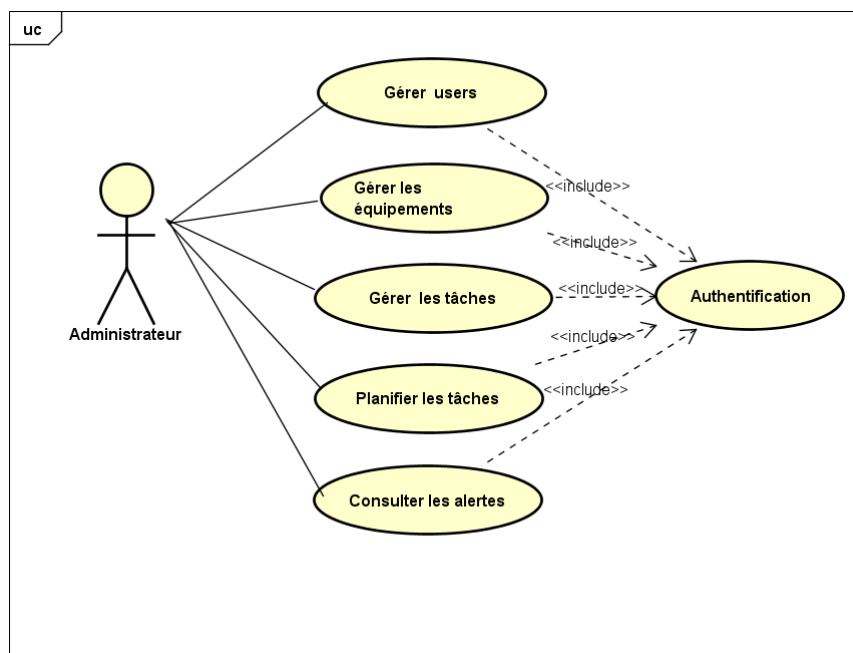


FIGURE I.4 – useCase

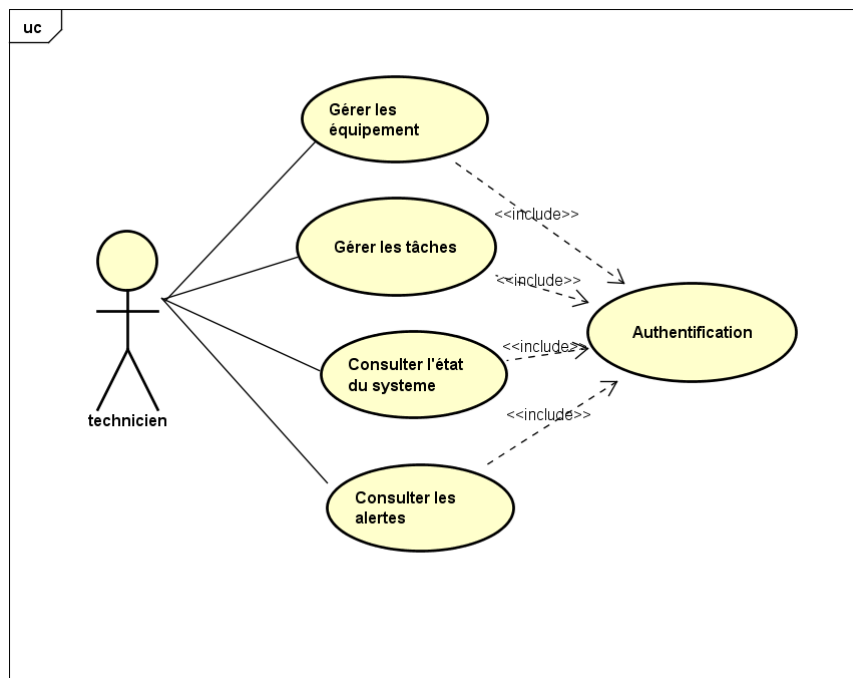


FIGURE I.5 – useCase

I.3 Méthodologie de développement :

I.3.1 Méthodes Agiles :

I.3.1.1 Définition :

Une méthode Agile est une approche itérative et collaborative, capable de prendre en compte les besoins initiaux du client et ceux liés aux évolutions.

La méthode Agile se base sur un cycle de développement qui porte le client au centre. Le client est impliqué dans la réalisation du début à la fin du projet. Grâce à la méthode agile le demandeur obtient une meilleure visibilité de la gestion des travaux qu'avec une méthode classique. L'implication du client dans le processus permet à l'équipe d'obtenir un feedback régulier afin d'appliquer directement les changements nécessaires. Cette méthode vise à accélérer le développement d'un logiciel. De plus, elle assure la réalisation d'un logiciel fonctionnel tout au long de la durée de sa création.

Le principe de base consiste à proposer une version minimale du logiciel puis à intégrer des fonctionnalités supplémentaires à cette base, par processus itératif. Le processus itératif regroupe une séquence d'instructions à répéter autant de fois que possible, selon le besoin. En ce qui concerne la réalisation d'un logiciel, les instructions à répéter sont les suivantes :

- Les tests unitaires à la fin de chaque itération
- Le développement de l'application web
- L'intégration
- La relecture et amélioration des codes

Les deux méthodes agiles les plus utilisées :

- Scrum
- la méthode Extreme Programming, qu'on appelle aussi méthode XP ⁴

Tout au long de notre projet, nous avons adopté la méthode "Scrum" que nous la présentons ci-dessous

I.3.1.2 Scrum :

Scrum est un schéma d'organisation de développement de produits complexes. Il est défini par ses créateurs comme un « cadre de travail holistique itératif qui se concentre sur les buts communs en livrant de manière productive et créative des produits de la plus grande valeur possible ». Scrum est considéré comme un groupe de pratiques répondant pour la plupart aux préconisations du Manifeste Agile.

Parmi les entreprises utilisatrices de cette méthode, de grands noms apparaissent comme Microsoft, Yahoo, Google, Electronic Arts, Nokia...

cette méthode nécessite 4 types de réunions :

- **Les réunions quotidiennes** : chaque jour, toute l'équipe se réunit, généralement debout, pendant 15 minutes environ pour répondre aux 3 questions suivantes : qu'ai-je fait hier ?, Que vais-je faire aujourd'hui ? Y a-t-il un obstacle gênant aujourd'hui ?
- **Les réunions de planifications** : toute l'équipe se réunit pour décider des fonctionnalités qui vont composer le sprint suivant et mettre à jour la liste générale.
- **Les réunions de revue de travail** : lors de cette réunion, chacun présente ce qu'il a fait pendant la durée du sprint. Une démonstration des nouvelles fonctionnalités ou de présentation de l'architecture est organisée. Il s'agit d'une réunion informelle de 2 heures environ à laquelle participe toute l'équipe.
- **Les réunions de rétrospectives** : à chaque fin de sprint, l'équipe fait le point sur ce qui a bien fonctionné et sur ce qui a moins bien fonctionné. Lors de cette réunion d'une durée de 15 à 30 minutes où chacun est invité et parle en son nom, un vote de confiance est organisé pour décider des améliorations à apporter.

La méthode Agile Scrum fait intervenir 3 rôles principaux qui sont :

- **Product owner** : Dans la majorité des projets, le responsable produit (product owner) est le responsable de l'équipe projet client. C'est lui qui va définir et prioriser la liste des fonctionnalités du produit et choisir la date et le contenu de chaque sprint sur la base des valeurs (charges) qui lui sont communiquées par l'équipe.
- **ScrumMaster** : Véritable facilitateur sur le projet, il veille à ce que chacun puisse travailler au maximum de ses capacités en éliminant les obstacles et en protégeant l'équipe des perturbations extérieures. Il porte également une attention particulière au respect des différentes phases de Scrum.

4. Extreme Programming (XP) est une méthode agile plus particulièrement orientée sur l'aspect réalisation d'une application, sans pour autant négliger l'aspect gestion de projet. XP est adapté aux équipes réduites avec des besoins changeants.

- **Équipe** : l'équipe regroupe tous les rôles habituellement nécessaires à un projet, à savoir l'architecte, le concepteur, le développeur, le testeur, etc. L'équipe s'organise elle-même et elle reste inchangée pendant toute la durée d'un sprint.

Fonctionnement :

Les fonctionnalités à implémenter sont contenues dans le backlog du produit. Au début de chaque sprint, des fonctionnalités sont ajoutées dans le backlog du sprint, fonctions qui devront être réalisées dans une durée fixe allant de 2 à 4 semaines. Tous les jours auront lieu des mêlées quotidiennes afin de voir l'état d'avancement de chacun. À la fin de chaque sprint, une version du produit, utilisable peut-être fournie au client. Toutes ces notions sont expliquées plus en détail par la suite :

- **Sprint** : Les sprints sont des itérations de durées fixes de une à quatre semaines. Chaque sprint représentant des fonctions à réaliser. L'esprit de Scrum est la participation forte du client (le responsable du produit) afin qu'il définisse la priorité de chaque fonctionnalité et choisisse celles qui seront réalisées durant un sprint. De plus, il peut compléter ou modifier la liste des fonctionnalités à venir mais pas celles en cours. Ceci signifie qu'à la fin de chaque sprint, une version de l'application peut-être montrée au client. Il voit ainsi l'évolution de l'application et ne la découvre pas à quelques jours de la deadline. Le produit fourni à la fin de chaque Sprint est codé et testé ! Les releases sont en fait un regroupement d'itérations afin de mieux identifier les versions du produit. Une release permet d'aboutir à une version livrable et dont la mise en exploitation est possible.
- **Backlog du produit** : Le backlog produit est un catalogue de toutes les fonctionnalités. Chaque item de ce catalogue possède deux attributs. Le premier étant une estimation en Points Relatifs, ce qui signifie que l'estimation n'est pas faite en nombre de jours mais en termes de complexité. La complexité a généralement comme valeur un nombre de la suite de Fibonacci (1 2 3 5 8 13 21 34 55 89 144 233 377 610...). C'est l'équipe qui détermine la complexité de l'item. Pour cela, l'équipe se réunit lors de ce que l'on appelle le Planning Poker, et chaque personne écrit la complexité qu'il juge sur un bout de papier que tout le monde abat en même temps. Suite à cela, l'équipe débat sur ces valeurs et chaque personne argumente. L'estimation se reproduit jusqu'à ce que la plupart soit d'accord. Les items sont bien quant à eux estimés en points.
- **Backlog du sprint** : Lors du démarrage d'un sprint, des items du backlog du produit sont choisis afin de constituer ce sprint. Chaque item est alors divisé en petites tâches constituant ainsi le backlog du sprint. Chaque tâche étant estimée en nombres d'heures et ne pouvant pas dépasser 2 jours. Celles-ci sont prises au fur et à mesure par les membres de l'équipe, elles ne sont pas attribuées dès le départ du sprint. Le backlog est ainsi mis régulièrement à jour avec les éléments restant à faire.

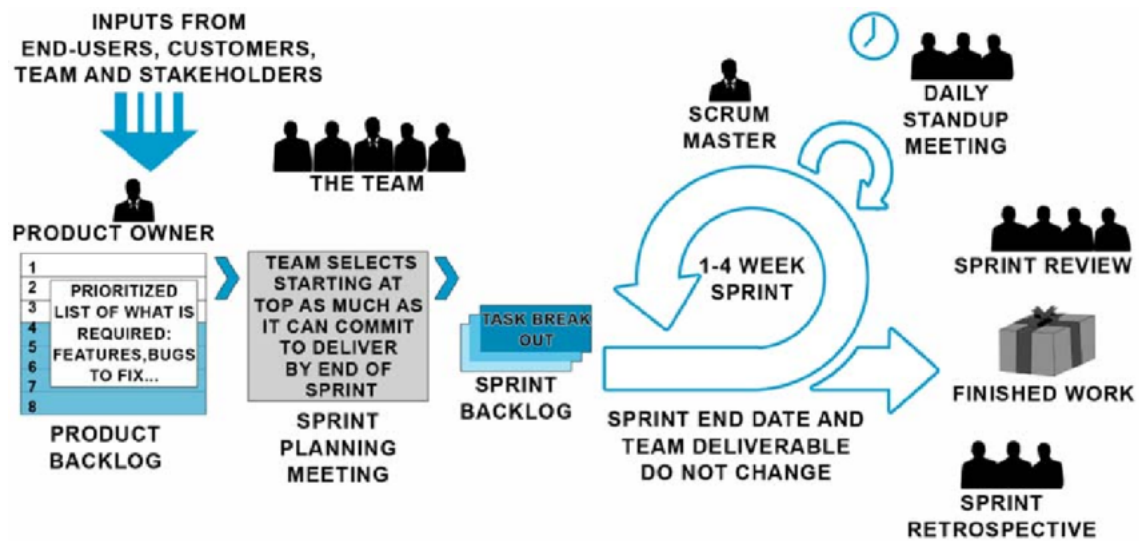


FIGURE I.6 – Scrum

● **Avantage :**

Scrum se différencie des autres méthodes de développement par ses avantages qui font de ce procédé une réponse pragmatique aux contraintes actuelles des chefs de produits

- Une démarche d'amélioration continue visant à augmenter la qualité et la productivité
- Une construction itérative et incrémentale du logiciel
- Plus grande réactivité, Flexibilité aux changements
- Une organisation favorisant la communication entre les équipes projet et les utilisateurs métiers
- Le pilotage par les tests pour assurer la non régression au fil des évolutions
- Maîtriser les coûts de développement

Conclusion :

Tout au long de ce chapitre nous avons présenté l'organisme d'accueil par la suite nous avons pu dégager le contexte générale du projet et présenter le choix de méthodologie de développement.