

Gérard NGUYEN

Etude d'une solution de suivi d'incident

DUT INFORMATIQUE

Sommaire

Introduction générale.....	3
Cahier des charges	3
Contexte et définition du problème	3
Objectif	4
Cycle de vie.....	5
Risques.....	6
Analyses et gestion des risques.....	6
Analyse des besoins	7
Spécifications fonctionnelles.....	7
Spécifications techniques.....	8
Conception de l'application	8
Sécurité.....	10

Introduction générale

Notre-Dame « Les Oiseaux » fondé en 1929 est un grand établissement scolaire sous la tutelle de la Congrégation Notre-Dame – Chanoinesses de Saint-Augustine. Cette école est implantée dans un parc de 14 hectares à Verneuil-sur-Seine dans les Yvelines. Regroupant des élèves de la maternelle jusqu'à l'enseignement supérieur, elle les accompagne tout au long de leur scolarité et a pour but leur réussite dans leurs projets. Avec une superficie aussi importante, une population d'élèves de 3300 ainsi que plus de 200 utilisateurs (professeurs et administration), il est primordial de gérer un parc informatique de grande envergure afin d'améliorer le cadre et les outils de travail pour les élèves et les professeurs. C'est pourquoi, des informaticiens sont à dispositions pour répondre aux besoins des utilisateurs, mais aussi des équipements tels que des ordinateurs, bornes wifi, Ipad etc... sont mis en place pour leur confort.

Cahier des charges

Contexte et définition du problème

Au sein d'une organisation aussi grande que Notre-Dame, d'un point de vue informatique, il est primordial de pouvoir répondre aux besoins des utilisateurs. Actuellement, la solution permettant de faire la relation entre l'utilisateur et le service informatique passe par 2 modes de communications :

- Formelle : passant par les mails afin de procéder à l'ouverture d'un ticket sur une messagerie commune relative au service informatique ;
- Informelle : passant par le bouche à oreille, au détour d'un couloir, par téléphone, etc...

Aussi, plusieurs problèmes se posent :

- Il y a une difficulté pour attribuer les problèmes pour chaque informaticien ;
- Le suivi des tickets reste difficile, les mails peuvent être perdus avec l'afflux d'incident ;
- Difficulté pour retracer des incidents antérieurs, via un historique qui n'est pas présent ;

Ils subsistent par ailleurs plusieurs inconvénients concernant ces 2 modes de communication. Premièrement, la communication formelle, bien qu'officielle et permettant une trace informatique et donc historique, ne nous permet pas un réel suivi sur le long terme de chaque incident.

En effet, au fur et à mesure du temps, les mails se perdent à la réception de nouveaux incidents. Il est possible dans une certaine mesure de garder les mails importants. Cependant dans une autre mesure, à long terme, il n'est pas possible d'avoir un réel historique sur l'incident.

Deuxièmement, le problème se posant quant à la communication informelle est qu'elle ne permet aucune traçabilité si la communication dans une entreprise est défaillante. Bien qu'elle permette une certaine flexibilité au niveau de la communication. La résolution se fera directement entre le technicien et l'utilisateur. Cela peut être difficile pour faire remonter les informations exprimées par l'utilisateur aux autres techniciens par exemple.

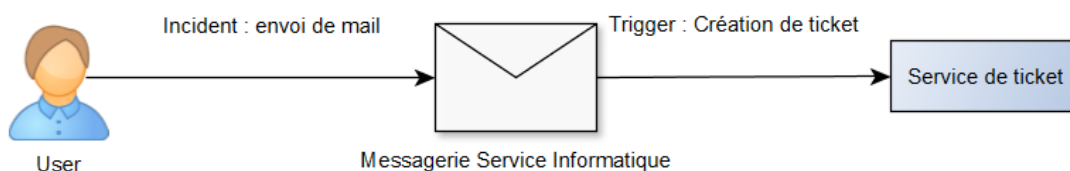
Aussi, nous nous concentrerons ici sur le cas de la communication formelle et proposer une solution aux différents problèmes rencontrés.

Objectif

L'objectif de la solution de suivi d'incident serait de pouvoir avant tout attribuer pour chaque technicien le ticket demandé par l'utilisateur. Grâce à cela, il n'y aura pas de double travail, pas de perte de communication, et donc pas de perte de temps. A long terme, cela nous permettrait d'améliorer notre système d'information et donc le confort de nos utilisateurs.

Il faut aussi savoir que nous sommes dans un contexte où les utilisateurs sont pour la plupart néophytes dans l'informatique.

Il faudra alors simplifier le plus possible le travail pour l'utilisateur, essayer de trouver une méthode ne changeant pas totalement les systèmes d'informations. Actuellement, l'envoi de ticket par utilisateur passe par la messagerie, il faudrait donc dans l'optique faire en sorte que ces modifications ne touchent pas l'utilisateur et que cela soit donc invisible pour celui-ci. Dans l'idéal il faudrait un système ou application permettant de faire une liaison avec notre messagerie sur notre application avec un trigger sur la création de ticket pour chaque mail.



Ci-dessus un schéma de l'objectif idéal

Pour faire évoluer et avoir une meilleure gestion de notre système d'information, il serait judicieux de s'inspirer des normes ITIL. Ce dernier étant un référentiel décrivant

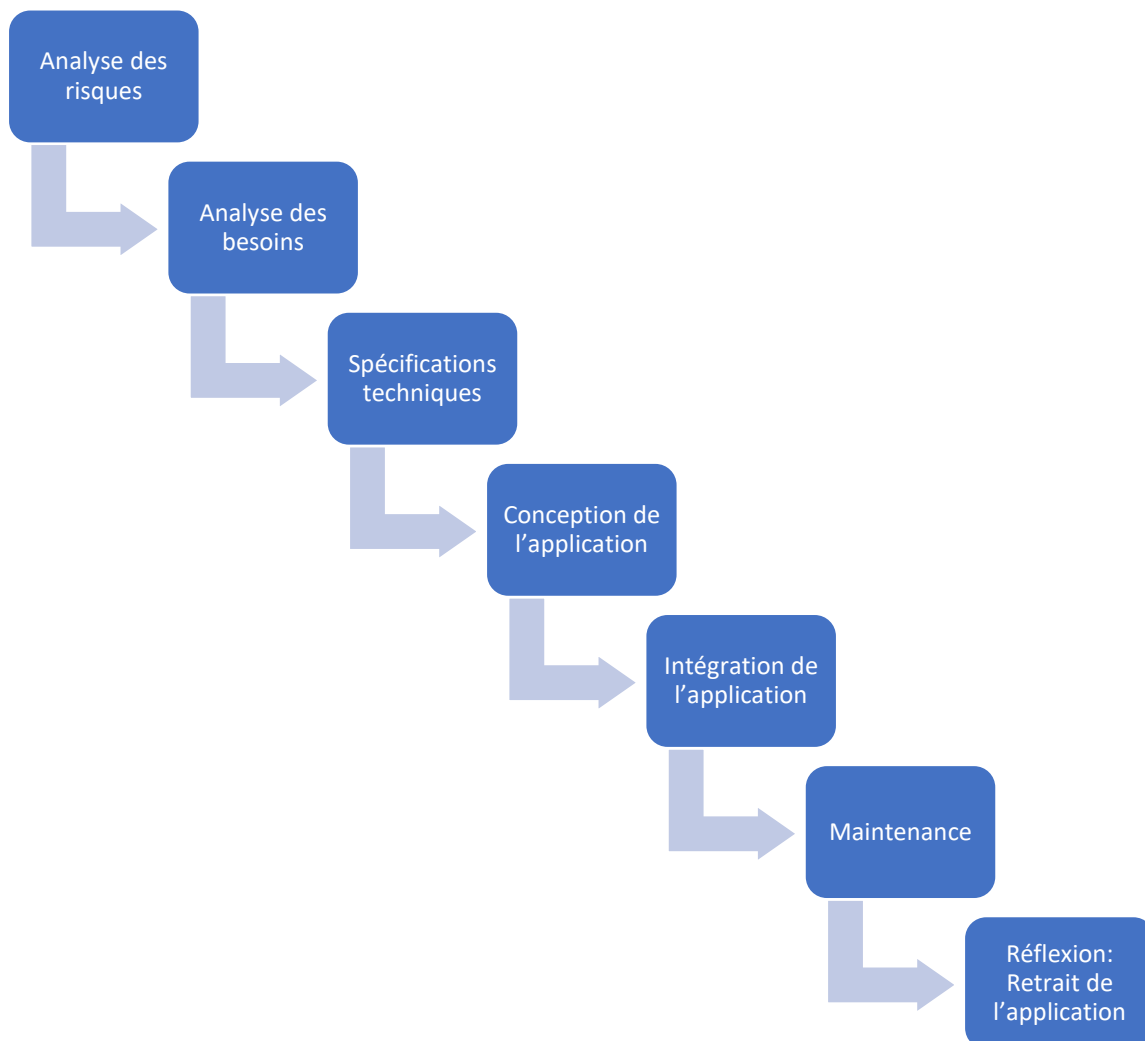
l'ensemble des processus de gestion de services technologiques utilisé par le métier. Il définit donc le best practices du management des systèmes d'informations.

A long terme, cela nous permettra d'utiliser les ressources humaines de manière efficaces, d'augmenter la qualité et adapter certains processus de l'entreprise, améliorer des processus déjà existants au sein de l'entreprise.

Cycle de vie

Le cycle de vie qui correspondrait le mieux selon moi à ce projet serait un cycle de vie en cascade. En effet, plusieurs indices pourraient indiquer cela :

- La définition des besoins au niveau de ce projet n'est pas évolutive, c'est-à-dire, que cette application aura seulement pour but de résoudre les problèmes et atteindre les objectifs dit précédemment, bien qu'il faille voir cette application sur le long terme et donc prendre du recul pour voir nos besoins futurs.
- La qualité est primordiale pour chaque étape du cycle de vie, nous n'aurons pas de réelles restrictions de temps sur chaque étape si ce n'est le respect des délais du projet,
- Enfin, il est obligatoire ici de valider chaque étape pour passer à la suivante.



Ci-dessus le cycle de vie en cascade pour notre projet. Il faut aussi spécifier que pour chaque étape, il est possible de retourner en arrière.

Risques

Analyses et gestion des risques

Avant de faire l'analyse des besoins pour notre système ou application, il est primordial de faire une analyse de risque. Cela nous permettra alors de pouvoir faire un plan dit de gestion des risques et donc de pouvoir obtenir un plan d'actions correctrices et préventives pour chaque problème susceptible d'être rencontré au cours de notre projet.

Le tableau suivant aura pour caractéristique de pouvoir prioriser chaque risque. Les paramètres seront alors :

- Le risque identifié en amont ;
- La probabilité de l'évènement ;
- La gravité du risque ;
- La criticité, qui nous permettra de prioriser le risque ;
- Et l'action préventive/corrective.

L'évaluation de la probabilité et la gravité de chaque risque se fera sur une échelle de 0 à 5.

Risques	Probabilité	Gravité	Criticité (=P*G)	Action préventive/corrective
Cahier des charges incomplet	3	4	12	Finalisation du cahier des charges
Dépassement du délai	2	4	4	Fixation d'une date limite pour chaque étape de notre cycle de vie
Difficulté matérielle	1	5	5	Proposition des solutions peu coûteuses ou déjà existantes dans notre organisme

Sous-estimation de l'intégration de l'application/système	3	4	12	Documentation, décrire un objectif à atteindre pour chaque étape du cycle de vie, méthode COCOMO ou points de fonctions
Pas de prise en compte de l'évolution	2	1	2	Réflexion sur la pérennité et l'utilisation à long terme de l'application

On peut observer que pour notre futur projet, la problématique se posera avant tout sur le cahier des charges qui pourraient être incomplet mais aussi sur la sous-estimation de l'intégration de l'application. Aussi puisque la probabilité de ces événements est forte, on devra y donner une attention particulière.

Cette liste de risques n'est pas exhaustive, au cours du projet, il sera toujours possible d'évaluer de nouveau risque non étudié en amont.

Analyse des besoins

Les besoins se feront sur différents plans. D'un point de vue utilisateur, il faudrait un moyen de communication simple, et une intervention rapide pour résoudre et clore leurs problèmes.

Il convient aussi de faire une estimation du coût d'investissement et de fonctionnement de ce projet en termes de moyens humains mais aussi matériels.

Le besoin actuel se fait surtout en interne, au sein de notre service informatique. Il faudra alors concevoir en priorité une application permettant l'attribution de ticket en fonction du technicien.

Spécifications fonctionnelles

Notre application devra avoir principalement une fonction de suivi avec notamment un historique, pouvoir modifier le statut du ticket (nouveau, en cours, en attente, résolu, clos), la priorisation du ticket (améliore le système d'information pour le confort de

l'utilisateur), le lieu de l'évènement, l'attribution du technicien, un titre et la description du ticket.

Il faudra aussi avant tout permettre un système de notifications qui sera lié à notre service de messagerie afin de créer automatiquement un ticket lors de la réception d'un message sur notre boîte mail.

Spécifications techniques

Dans cette partie nous nous intéresserons à la technique et aux moyens utilisés pour arriver à nos fins.

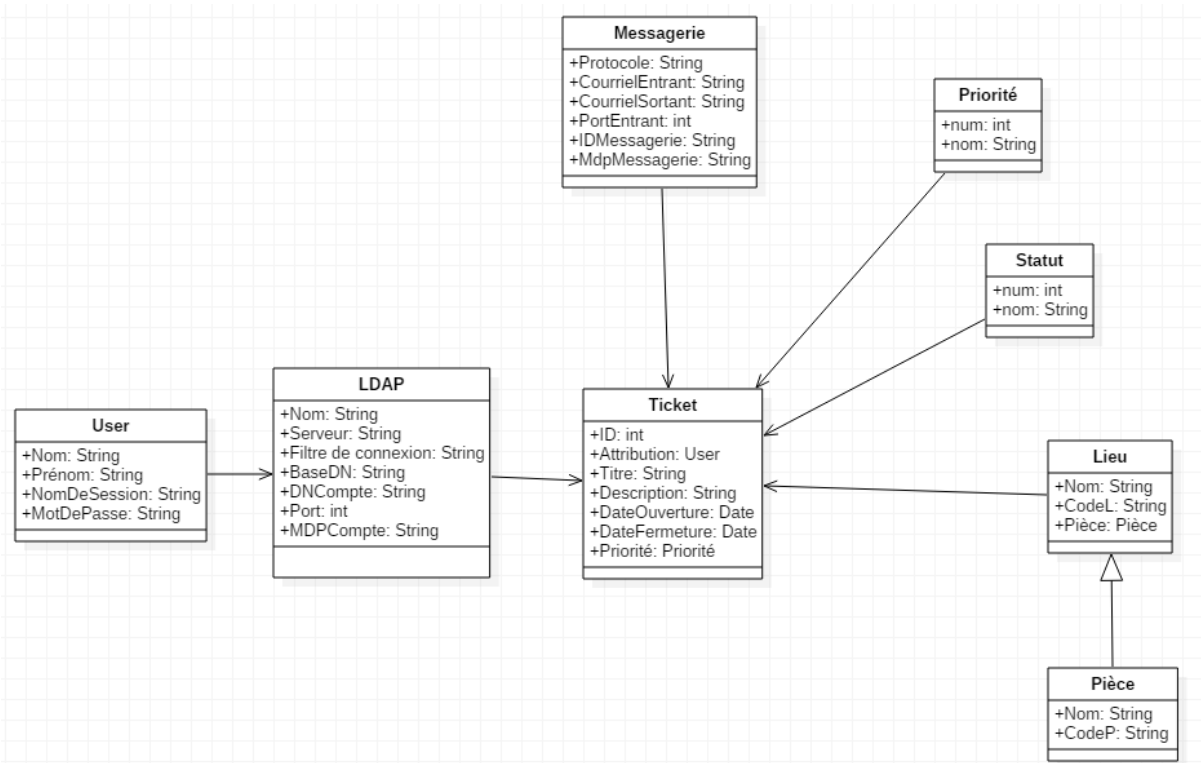
Du point de vue matériel, l'application que l'on proposera sera une application web. Il faudra donc un serveur web mais aussi un serveur de base de données pour garder nos données. On utilisera aussi le langage de programmation PHP et PDO pour la connexion à la base de données. Ce choix a été fait car le PHP est un langage natif à l'application Web, beaucoup de frameworks existent et la compilation et la recherche d'erreurs se trouvent simple avec ce langage. Aussi pour un site de notre envergure et qui sera utilisé seulement par notre équipe informatique, l'utilisation du langage PHP semble être incontournable.

Pour minimiser les coûts de matériel et l'efficacité, on utilisera pour nos deux serveurs, web et de BDD, des machines virtuelles qui auront pour but de simuler des machines physiques. Cela nous permettra aussi dans un long terme de plus facilement procéder au retrait de l'application. On utilisera Apache pour notre serveur web (le plus répandu sur internet) et MariaDB ou MySQL pour notre serveur de BDD.

Notre application devra avoir un système permettant des notifications grâce à notre serveur de messagerie. Mais aussi, une authentification LDAP, ce qui nous évitera de devoir créer des comptes et permettra donc de réutiliser les comptes de notre Active Directory (permet une certaine facilité d'utilisation par la suite).

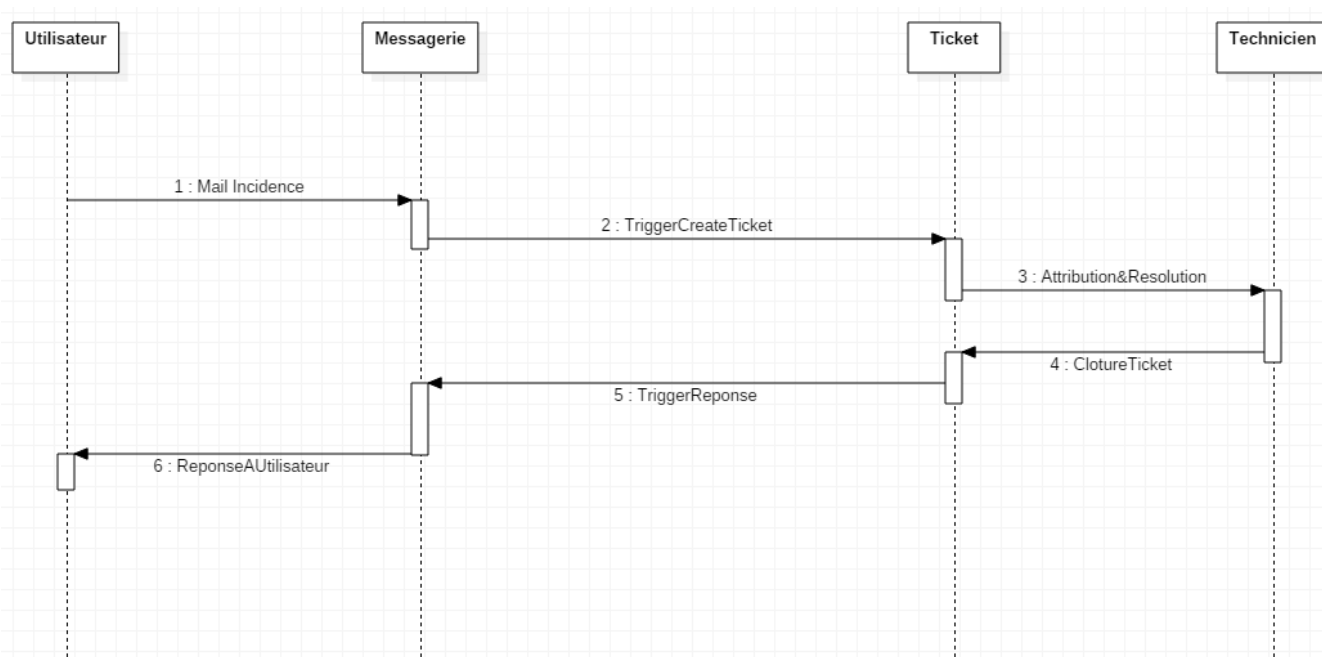
Conception de l'application

Nous allons ici élaborer notre diagramme de classe. Ce dernier représentera les classes de notre système ainsi que les différentes relations entre chaque classe. Cela nous permettra un meilleur point de vue quant à notre future application.



Ci-dessus le diagramme de classe non exhaustif de notre application de ticket.

On synchronisera alors notre application de ticket avec notre serveur LDAP. Cela nous permettra une facilité au niveau des connexions et ne pas devoir recréer des comptes pour chaque technicien ou personne. Le ticket sera aussi créé via notre système de messagerie grâce à une action automatique ou trigger.



Ci-dessus le diagramme de séquence pour la création d'un ticket et sa résolution.

On peut voir ici que pour chaque message envoyé par l'utilisateur sur notre serveur de messagerie. Un ticket se crée automatiquement grâce à un trigger sur action. Le technicien intervient et résout le souci rencontré par l'utilisateur, clôt le ticket avec obligatoirement une réponse avec la solution du problème. Cette réponse est donc renvoyée à l'utilisateur sous forme de mail afin qu'il soit notifié que le problème est bien résolu.

Sécurité

D'un point de vue de la sécurité, l'incontournable pour une application WEB/HTTP serait d'obtenir un certificat HTTPS qui permettrait à nos données d'utiliser un protocole dit de sécurité en utilisant une couche de chiffrement comme le SSL (Secure Sockets Layer, les données transiteront dans des sockets chiffrées). Aussi cette méthode efficace, ne pourra pas être mis en œuvre, en effet, l'utilisation d'un certificat pour la sécurisation a un coût. La création d'un certificat auto-signé peut être une solution cependant, cela ne nous garantira jamais à long terme une sécurisation de nos données. Il faudra donc peser le pour et le contre de ces solutions.

En ce qui concerne nos moyens pour communiquer les informations de compte et message entre notre service de ticket et notre LDAP/serveur de messagerie, on utilisera une délégation de contrôle pour chaque moyen.

La délégation de contrôle permettrait ici de faire office de proxy, on créera un compte « proxy » qui aura pour rôle de faire l'intermédiaire entre notre service de ticket et notre LDAP/messagerie. Ce « proxy » aura seulement des droits en lectures sur nos différents systèmes. Dans le cas où notre système est corrompu et qu'un utilisateur malintentionné à accès à ce « proxy », il aura seulement des droits en lecture et non écriture.