

Organisation, information et système d'information

1 Les systèmes d'informations

Dans cette section nous introduisons le concept de développement de systèmes d'information en utilisant l'analogie de la construction immobilière.

Qu'est-ce qu'un système d'information ?

Il s'agit d'une collection de programmes pour répondre à un besoin de traitement d'information.

Un système d'information est un ensemble d'activités qui saisissent, stockent, transforment et diffusent des données sous un ensemble de contraintes appelé l'environnement du système. Des inputs (données entrantes) sont émis par une ou plusieurs sources et traités par le système, lequel utilise aussi des données entreposées préalablement. Les résultats de traitement (outputs) sont transmis à une ou plusieurs destinations ou mettent à jour des données entreposées. Pour sa réalisation, un système d'information utilisera des technologies de l'information plus ou moins sophistiquées jusqu'à des réseaux d'ordinateurs extrêmement puissants, utilisant des interfaces de type multimédia.

Qu'est-ce que le développement de systèmes d'information ?

Il s'agit du processus qui est mis en branle pour produire un système d'information.

Pourquoi utilise-t-on les techniques de développement de systèmes ?

Développement en équipe :

De nos jours, les produits courants font appel la plus part du temps à la participation de nombreuses disciplines scientifiques. L'aspect pluridisciplinaire et la complexité font que le développement n'est plus une œuvre individuelle; il est confié à une équipe qui doit respecter les délais souvent courts de la mise sur le marché. Les difficultés potentielles sont propres au travail d'équipe : les personnes provenant d'horizons divers se heurtent à des problèmes de communication, de vocabulaire, de compétences, de mode de conception de documents de travail...

L'absence d'une méthodologie, même sommaire cause en cascade toutes sortes de problèmes imprévus, en particulier un prolongement de temps de développement, une interprétation variable d'une même solution, un produit non opérationnel, un mauvais suivi dans le développement, une maintenance difficile et finalement à des pertes financières.

Complexité croissante des produits :

Les logiciels sont plus performants car les utilisateurs sont plus exigeants et les environnements plus standardisés. Les clients exigent en outre que le produit ait une durée de vie assez longue, possède un support de maintenance et d'évolution dans le temps. Les exigences de qualités et la

complexité croissante des produits sont telles que les personnes impliquées ne participent réellement qu'à certaines étapes de développement. Cette qualité et cette complexité croissante requièrent une méthodologie efficace adaptée au contexte mouvant du développement.

Mobilité des ressources humaines :

Les compagnies qui développent des produits font face à plusieurs difficultés parmi lesquelles figure le problème de la mobilité des ressources humaines. Quand un individu quitte une compagnie, son départ ne doit pas mettre en danger la vie du produit ou de la compagnie. La formation d'une nouvelle personne sur le produit doit prendre un temps minimal. Or, une documentation incomplète ou non structurée des centaines de milliers de lignes de code rend la tâche impossible. Il est important de disposer d'outils de développement accompagnés de documents synthétiques pertinents.

L'histoire ne s'arrête pas à la fin du développement :

Le développement d'un produit n'est qu'une étape du cycle de vie d'un produit industriel ou commercial. Un produit doit avoir une certaine durée de vie. Sur le plan purement technique, le coût des maintenances corrective ou évolutive est parfois plus élevé que le coût du développement initial. Si le système est bien conçu initialement, l'entretien de celui-ci est alors moins onéreux.

Pour l'apprenti informaticien, il peut être difficile de visualiser ce qu'est un système d'information. En effet, depuis le début de son apprentissage, il n'a pu participer au développement de programmes de grande taille. Toutefois, en tant qu'utilisateur, il a la connaissance de systèmes complexes et même très complexes (exemple : environnement de travail Windows, logiciel de réseau Windows NT 4.0 Server, accès aux différents guichets automatiques, etc.) Pour pouvoir participer au développement de tels systèmes, l'apprenti informaticien doit d'abord maîtriser les techniques de développement de systèmes.

Plus le projet est complexe et plus ces techniques sont nécessaires, comme le démontre, par analogie, l'exemple de la construction immobilière.

Analogie de la construction immobilière.

Si quelqu'un désire construire une niche pour son chien, il n'a qu'à se faire un petit plan sur un bout de feuille, se ravitailler de bois chez le quincaillier et se lancer dans la construction la journée même. Selon l'expérience et le talent de notre constructeur, la niche pourra répondre aux besoins de toutou. Si, toutefois, il y a un défaut majeur de construction (voir figure 1 : la pire éventualité), notre «débrouillard» devra refaire une bonne partie de son travail et peut-être même



Figure 1 - Niche de chien (Code complète, S McConnell, MsPress

retourner chez le quincaillier. Il risquera d'y passer sa fin de semaine si ça lui plaît comme hobby...

(Un apprenti programmeur passe inévitablement ses fins de semaine à se construire de petits programmes, à les refaire, les modifier, les améliorer...)

Par contre, si on désire se construire un petit chalet tout coquet et très rustique (sans électricité, sans eau,...), alors on va passer les fins de semaine de l'hiver à choisir l'emplacement, la grandeur, les angles, etc. Le printemps venu, on invite son beau-frère menuisier et on se lance dans l'aventure.

La construction d'une maison, c'est beaucoup plus sérieux, ça requiert le travail de professionnels tels que les plombiers, les électriciens, menuisiers, architectes, décorateurs parfois, etc. Si certaines personnes prennent en main la construction de leur propre maison, la plupart engageront un entrepreneur qui offrira les services d'une équipe complète de professionnels, chacun œuvrant selon un plan bien défini (architecture à approuver, plan de la maison, choix du site, achat, construction, inspection, décoration,...)

De la même façon, lors du développement de systèmes, les équipes d'informaticiens effectueront toutes les étapes de planification, de conception, de construction et de contrôle. Cette façon de travailler assurera la bonne qualité du produit : on appelle ça le professionnalisme. Ainsi on évitera que le client soit insatisfait parce qu'on lui cause des pertes de temps, d'argent etc.

Lorsqu'on se construit une maison, on indique ses spécifications et on négocie le prix. Si l'équipe de construction ne respecte pas cette entente (en plaçant une fenêtre au mauvais endroit, en livrant la maison en retard ou avec quelques défauts que ce soit), l'entrepreneur et ses employés auront affaire à un client de bien mauvaise humeur. On rencontre ce même type de situation en informatique...

On peut aussi remarquer que les constructeurs de maison font appel à des éléments préfabriqués tels que les lavabos, comptoirs ou armoires. Cette analogie s'applique à l'informatique lorsqu'on fait appel à des librairies pour compléter notre programmation.

La construction d'un édifice nécessitera un très gros budget et s'écoulera sur de longs mois (voir des années). Des équipes complètes de professionnels se succéderont à l'intérieur d'un calendrier très strict (tout délai entraînant des pertes importantes). Comme dans ce dernier cas, le développement de systèmes très complexes devra suivre des normes très strictes.

2 Le processus unifié de développement

Le processus unifié est un processus de développement de logiciel qui vise à transformer les besoins des utilisateurs en un produit logiciel.

Le processus unifié utilise le langage UML (Unified Modeling Language) pour la création des plans d'élaboration et de construction du système logiciel. UML fait partie intégrante du processus unifié.

Le processus unifié est piloté les cas d'utilisation :

L'objectif d'un système logiciel est de rendre service à ces utilisateurs. Pour réussir à mettre en œuvre un système, il est très important de bien cerner les besoins des futurs utilisateurs du système. Le terme utilisateur désigne aussi bien un humain, qu'un autre système dialoguant avec le système en cours de développement. Ce type d'interaction entre le système et l'utilisateur s'appelle CAS D'UTILISATION.

Un cas d'utilisation est une fonctionnalité du système produisant un résultat satisfaisant pour l'utilisateur. (Exemple : cas du retrait d'argent d'un GAB). Les

Les cas d'utilisation décrivent le comportement du système du point de vue de l'utilisateur. Ils permettent de définir les limites du système et les relations entre le système et son environnement. Un cas d'utilisation est une manière spécifique d'utiliser le système. C'est l'image d'une fonctionnalité en réponse à la stimulation d'un acteur externe. Chaque cas d'utilisation décrit donc les besoins fonctionnels du système. L'ensemble des cas d'utilisation des cas d'utilisation constitue **le modèle des cas d'utilisation** et décrit les fonctionnalités complète du système. **Le modèle des cas d'utilisation** remplace la classique spécification fonctionnelle du système (analyse fonctionnelle)

Le processus unifié est centré sur l'architecture :

Le concept d'architecture logiciel représente les aspects statiques (exemples : les données) et dynamiques (exemples : les traitements ou fonctions ou cas d'utilisation) les plus significatifs du système.

- L'architecte crée une ébauche grossière de l'architecture, en partant de l'aspect qui n'est pas propre aux cas d'utilisation (plate forme..). Bien que cette partie de l'architecture soit indépendante des cas d'utilisation. L'architecte doit avoir une compréhension globale de ceux ci avant d'en esquisser l'architecture.
- Il travaille ensuite, sur un sous ensemble des cas d'utilisations identifiés, ceux qui représentent les fonctions essentielles du système en cours de développement. Chaque cas d'utilisation sélectionné est décrit en détail et réalisé sous forme de sous-système.
- L'architecture se dévoile peu à peu, au rythme de la spécification et de la maturation des cas d'utilisation, qui favorisent, à leur tour, le développement d'un nombre croissant de cas d'utilisation.

Ce processus se poursuit jusqu'à ce que l'architecture soit jugée stable.

Le processus unifié est itératif est incrémental :

Contrairement aux méthodes d'analyse traditionnelles, le processus unifié(PU) suggère une approche incrémentale du développement par une succession d'affinement ou d'itération autour de grandes disciplines du développement incluant l'expression des exigences, l'analyse et la conception, l'implémentation, le test et le déploiement. (Voir schéma)



Avantages :

Du fait que c'est une approche itérative

- Raffinement successif du système par de multiples cycles de vie d'analyse, de conception et de réalisation
- Contrôle de la complexité
- Feedback tôt
- Ajustement des besoins

Du fait que ce soit une approche incrémentale

- Ajout de fonctions par plusieurs cycles de livraison
- Facilite les relations avec les utilisateurs

3 Les principaux intervenants

Le nombre d'intervenants dans un projet variera selon l'ampleur et la complexité du projet. Voici une typologie relativement exhaustive, proposée par Y.C. Gagnon. Il faut noter qu'une même personne ne peut, selon les circonstances, appartenir à plus d'un groupe.

Les **décideurs** contrôlent les ressources utilisées et ont le pouvoir d'influencer l'ensemble du projet. Ils interviennent autant dans la sélection des processus et des systèmes que dans la définition des objectifs à poursuivre. Il s'agit souvent de la haute direction d'une organisation.

Les **gestionnaires** supervisent le processus et l'opération du système. Ils sont les représentants, à un niveau hiérarchique inférieur, des décideurs. Ils travaillent en collaboration avec les concepteurs.

Le **groupe des concepteurs** (analystes et concepteurs) analyse, développe et implante le processus et le système d'information en collaboration avec les décideurs et les gestionnaires.

Les **clients** interagissent avec le processus et le système par nécessité ou par choix. Ils utilisent les outputs du système et/ou du processus. Ils sont en contact direct avec le système pour de courtes périodes de temps ; ce sont les usages des guichets automatiques, les voyageurs qui effectuent une réservation de billets d'avion, les gestionnaires pour qui des rapports sont produits etc.

Les **usagers-opérateurs** sont ceux dont le rôle organisationnel est directement associé au processus et au système d'information. Ils produisent les intrants (inputs) et les transforment en extrants pour les clients du processus du système d'information.

Les **programmeurs** travaillent à l'élaboration des détails de la structure du système

Finalement, les **entraîneurs** enseignent aux usagers opérateurs et aux autres groupes comment utiliser le système.

Sources :

Le processus unifié de développement logiciel Par : I. Jacobson, G. Booch et J. Rumbaugh, 2000
Le développement de systèmes d'information de Jean Talbot et Suzanne Rivard

4 Rôle de l'analyste

L'analyste de systèmes n'est généralement pas la seule personne ayant des responsabilités dans un projet de développement de système. Dans le cas d'un très grand système on pourra voir une équipe composée d'un chef de projet, de plusieurs analystes de systèmes, de quelques spécialistes en qualité totale, d'utilisateurs, de plusieurs programmeurs et d'adjoints à l'administration. À l'autre extrême, on pourra avoir une équipe composée d'une seule personne qui jouera à la fois les rôles de chef de projet, d'analyste, de spécialiste en qualité totale, de programmeur et de secrétaire. Pour remplir ses fonctions de façon efficace, l'analyste devra donc posséder des connaissances dans plusieurs domaines, tant en gestion qu'en technologies de l'information.

L'analyste doit donc être en mesure de :

- Comprendre le système à l'étude et les tâches accomplies par les utilisateurs
- Comprendre les difficultés rencontrées par les utilisateurs
- Connaître les différentes méthodes de collecte d'information et de modélisation
- Proposer des solutions aux problèmes et de concevoir les aspects logiques du système correspondant
- Traduire ses solutions logiques en des scénarios concrets et d'évaluer les coûts et les bénéfices (monétaires ou humains)

En outre, l'analyste doit posséder certaines qualités essentielles, étroitement liées au contexte dans lequel se déroulent nombre de projets. En effet, le lancement d'un projet de système ou de transformation des processus engendre souvent des inquiétudes chez les utilisateurs. Certains y voient un moyen pris par leurs supérieurs pour évaluer leur compétence, d'autres ne sont que dérangés dans leurs habitudes, certains craignent une perte de pouvoir, alors que d'autres voient carrément leur emploi menacé. Ces malaises et ces inquiétudes amènent parfois l'utilisateur à résister d'emblée au changement éventuel que pourrait apporter un nouveau système et réduisent la probabilité d'une collaboration efficace à l'étude.

Dans un tel contexte, il est primordial que l'analyste fasse preuve de véritables qualités humaines. Rien n'est plus agressant, pour un utilisateur, que d'avoir affaire à un analyste qui donne l'impression de savoir mieux que lui comment accomplir sa tâche !