

# Introduction au génie logiciel



©

Hiver 2017

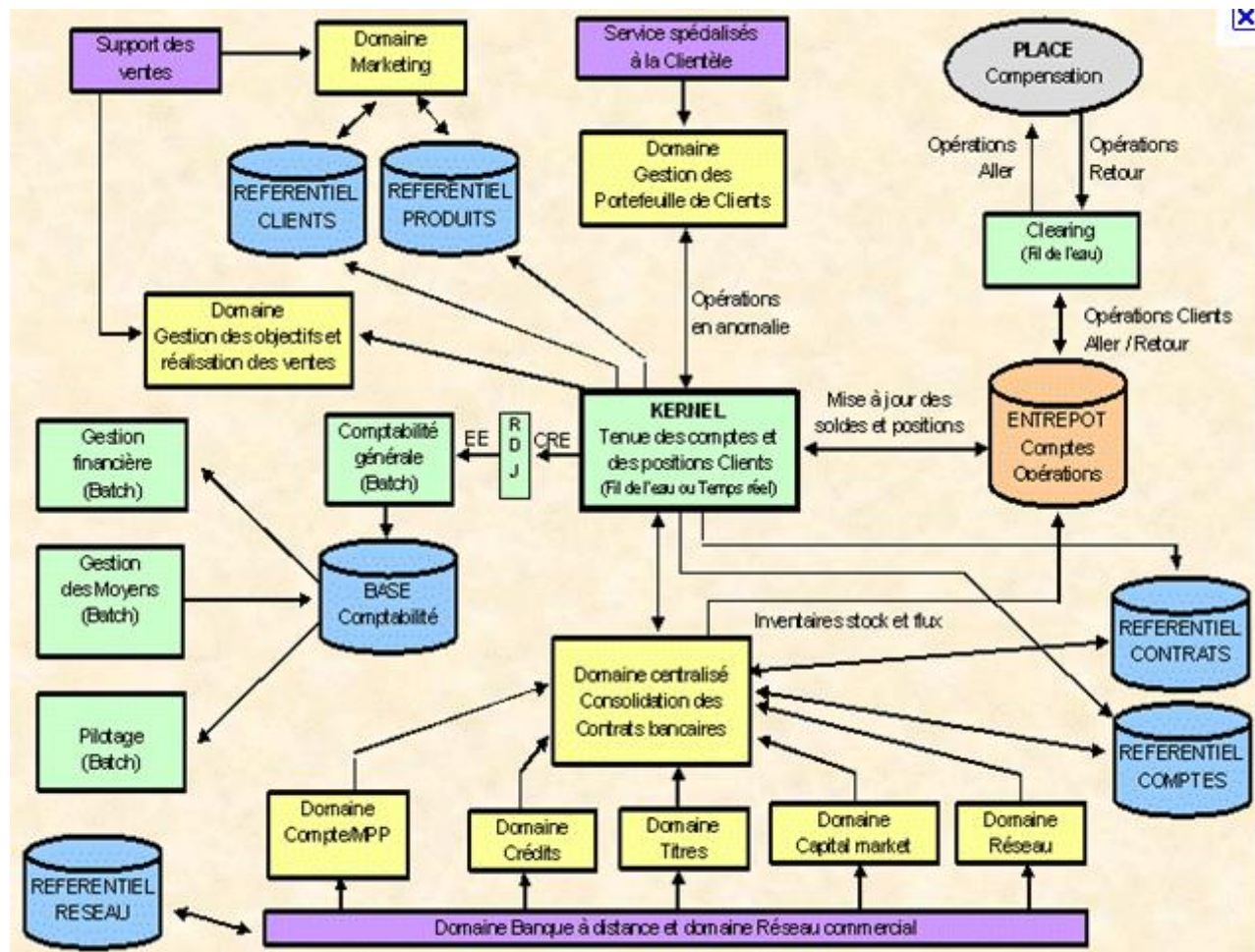
## Table des matières

1	Les systèmes .....	3
1.1	Définitions de système .....	3
1.1.1	Système de traitement des données.....	4
1.1.2	Système d'information .....	4
1.1.3	Système informatisé ou ordonné.....	4
1.1.4	Système informatique .....	4
1.2	Composantes d'un système .....	5
1.3	Caractéristiques d'un système.....	6
1.3.1	Frontière .....	6
1.3.2	Composition .....	7
1.3.3	Contrôle .....	7
1.3.4	Évolution .....	8
1.3.5	Mode de fonctionnement .....	9
1.3.5.1	Traitement par lots (BATCH) ou séquentiel.....	9
1.3.5.2	Traitement en ligne (ON-LINE) ou direct .....	9
1.3.5.3	Traitement en temps réel (REAL TIME) .....	10
1.4	Systèmes d'information .....	10
1.4.1	Systèmes d'information formels versus informels .....	10
1.4.2	Types de systèmes d'information.....	11
1.4.2.1	Systèmes de traitement des transactions ou systèmes de traitement transactionnel .....	11
1.4.2.2	Systèmes d'information de gestion.....	12
1.4.2.3	Tableaux de bord de gestion .....	12
1.4.2.4	Systèmes d'aide à la décision (S.A.D.) ou de support à la prise de décision .....	14
1.4.2.5	Systèmes experts ou systèmes à bases de connaissances (KNOWLEDGE-BASED SYSTEM).....	15
1.4.2.6	Systèmes à avantage compétitif ou concurrentiel .....	16
1.4.2.7	Systèmes auxiliaires de communication.....	17
1.4.2.8	Systèmes de soutien bureautique .....	17
2	Génie logiciel .....	18
2.1	Définitions de génie logiciel .....	19
2.2	Vie d'un système .....	19
2.2.1	Tâches et étapes .....	19
2.2.2	Appellations des étapes et groupes d'étapes .....	23

2.2.3	Approches de développement .....	24
2.2.3.1	Approche structurée .....	24
2.2.3.2	Approche orientée objets .....	26
2.2.3.3	Approche structurée versus approche orientée objets .....	26
2.3	Projets .....	28
2.3.1	Appel d'offres .....	29
2.3.2	Cahier des charges .....	29
2.4	Méthodes et approches .....	30
2.4.1	Outils .....	31
2.4.1.1	Outil GLAO .....	32
2.4.1.2	Atelier de génie logiciel .....	33
2.4.2	Techniques .....	33
2.5	Biens livrables .....	34
2.6	Intervenants dans un projet de développement .....	35
3	Conclusion .....	38

# 1 Les systèmes

Avant d'aborder le génie logiciel, il est de circonstance de s'intéresser à ce qu'est un système et le lien qui existe entre les systèmes et l'informatique. L'informaticien doit posséder une bonne connaissance des concepts des systèmes puisqu'il est amené à analyser, concevoir, implanter ou faire la maintenance des systèmes.



## 1.1 Définitions de système

Le terme «système» provient du grec («*systema*») qui signifie **«ensemble, organisation, mettre en rapport, instituer»**. Voici d'autres définitions provenant de divers dictionnaires qui présentent aussi un intérêt en informatique :

- combinaison d'éléments réunis de manière à former un ensemble (un tout)

- ensemble de méthodes, de procédés destinés à assurer une fonction définie ou à produire un résultat
- ensemble d'éléments (ressources humaines, matérielles, financières, activités, etc.) qui interagissent dans un but donné, selon certains principes ou certaines règles
- ensemble fonctionnel dont les parties sont interconnectées et échangent de la matière, de l'énergie ou de l'information

#### **Exemples**

*Système de chauffage, corps humain, système bancaire, système législatif, système de facturation, système de production, etc.*

Au fil du temps, l'informatique a précisé le terme « système » en lui ajoutant un qualificatif ou un complément du nom et c'est ce qui justifie pourquoi on rencontre parfois les expressions :

#### **1.1.1 Système de traitement des données**

ensemble d'éléments qui permettent d'extraire de l'information ou de produire du savoir à partir de données brutes

#### **1.1.2 Système d'information**

ensemble d'éléments (personnes, procédures, méthodes et techniques, équipements informatiques ou non) qui permettent la saisie, la mémorisation, le traitement et la circulation de données en vue de réaliser les activités nécessaires au bon fonctionnement d'une organisation

#### **1.1.3 Système informatisé ou ordonné**

système d'information qui utilise (partiellement ou complètement) un ordinateur comme moyen de traitement

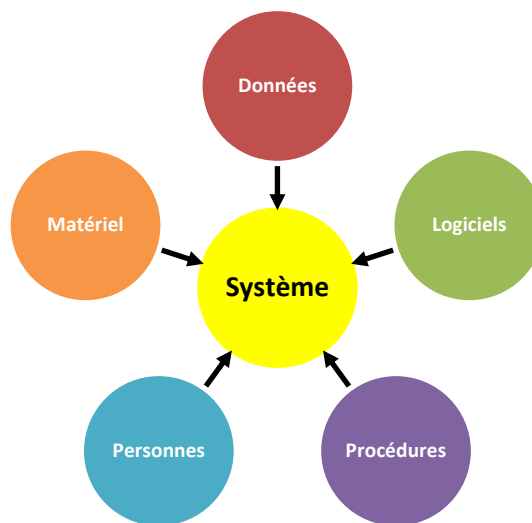
#### **1.1.4 Système informatique**

ensemble du matériel et des logiciels utilisés pour le traitement des données mais plusieurs confondent cette appellation avec les trois autres termes précédents.

## 1.2 Composantes d'un système

En informatique, un système doit être considéré comme un ensemble d'équipements, de méthodes et techniques d'utilisation de ces équipements, de procédures de travail et de personnel engagé dans le traitement des données. Par équipement, il faut comprendre tous les moyens de traitement (calculatrice, photocopieur, ordinateur, périphérique, réseau, etc.) et les supports (papier, cédérom, mémoire vive, etc.). L'utilisation de l'équipement ne peut se faire au hasard car les actions sont ordonnées et doivent suivre les méthodes et techniques d'utilisation. Ceci exige des personnes de connaître le fonctionnement et les limites de ces équipements. De plus, ce personnel est soumis à des procédures de travail ayant pour but de garantir l'efficacité et l'exactitude du travail. Ainsi, un système peut comporter jusqu'à cinq (5) composantes :

- 1) le matériel (ordinateur ou non mais aussi calculatrice, téléphone, etc.)
- 2) les logiciels (système d'exploitation, SGBD, application personnalisée, etc.)
- 3) les données (informations textuelles et multimédia que le système doit mémoriser une certaine période de temps)
- 4) les gens (les utilisateurs du système, personnel du support, etc.)
- 5) les procédures (politiques et instructions pour opérer le système).



### Exemple

*Un étudiant répond de la façon suivante à un examen de mathématiques distribué par le professeur.*

- 1) l'étudiant lit un problème ;
- 2) il note au crayon, sur des feuilles de papier, les données du problème ;
- 3) il détermine les formules à utiliser ;
- 4) il inscrit les différentes étapes de son raisonnement ;
- 5) il effectue les calculs à l'aide d'une calculatrice ;
- 6) il inscrit les résultats.

*À la fin de l'examen, il remet sa copie au professeur.*

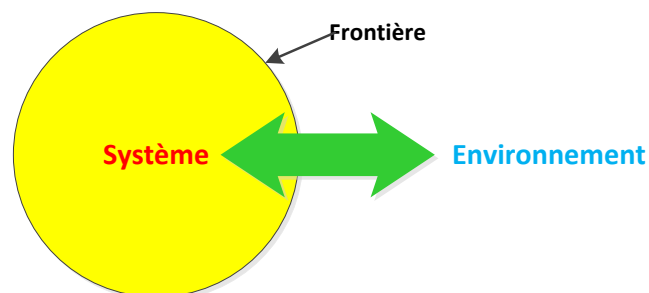
<u>Système</u> :	Évaluation d'un étudiant en mathématiques
<u>Matériel</u> :	Crayon, questionnaire, calculatrice et feuilles de papier
<u>Logiciel</u> :	Aucun ou en forçant, programme de la calculatrice
<u>Données</u> :	Questions, réponses, identification de l'étudiant, date et heure de début et fin de l'examen, etc.
<u>Personnes</u> :	Étudiant et professeur
<u>Procédures</u> :	L'étudiant doit savoir se servir de la calculatrice (méthode, technique d'utilisation) Il doit connaître l'heure du début et de la fin de l'examen Il doit répondre en respectant les directives du professeur

## 1.3 Caractéristiques d'un système

Les systèmes possèdent diverses propriétés qui permettent de les catégoriser. Voyons quelques caractéristiques principales des systèmes, à considérer en informatique.

### 1.3.1 Frontière

Le concept de frontière ou borne d'un système exige de différencier les systèmes ouverts des systèmes fermés. Un **système ouvert** est actif et complet en soi, reçoit des entrées et produit des sorties ; ainsi, il est influencé et influence son environnement. À l'opposé, un **système fermé** ne communique pas avec ce qui lui est extérieur. Dans la réalité, tous les systèmes sont ouverts. Toutefois, le concept de « système fermé » facilite l'étude des systèmes. Il permet de réduire le plus possible les échanges avec l'extérieur et ainsi d'établir de façon plus ou moins artificielle, une frontière qui limite l'étendue du système étudié. L'étude de l'environnement du système conduit à l'identification de ses intervenants externes (personnes ou autres systèmes) et des échanges avec eux. Ainsi, la frontière du système est tracée et il est alors possible de se concentrer sur ce qu'il y a à l'intérieur du système.



### Exemple

*Un système de paie est un système ouvert. Les employés fournissent au système des feuilles de temps; selon les entreprises, le système transmet aux employés leurs chèques ou effectue un dépôt directement auprès de leur institution financière. Le ministère du revenu fournit au système les tables d'impôt et autres paramètres pour calculer la paie. Le service des ressources humaines fournit les échelles salariales au système. Etc.*

## 1.3.2 Composition

En général, un système fait partie de plus gros systèmes ou peut être subdivisé en plus petits systèmes. Voir un système de ce point de vue facilite l'étude de systèmes complexes et l'identification des autres systèmes impliqués.



### Exemple

*Le système d'évaluation des apprentissages dans une école fait partie du système éducatif de l'école où l'on retrouve beaucoup d'autres systèmes tels que le système de l'organisation scolaire, le système de gestion des programmes et de l'enseignement, etc. Le système d'évaluation des apprentissages peut se subdiviser en d'autres systèmes comme la gestion des examens, le processus de révision de notes, etc.*

## 1.3.3 Contrôle

Pour qu'un système fonctionne bien, il doit être sous contrôle c'est-à-dire qu'il fonctionne à l'intérieur de niveaux de performance tolérables. Au-delà de ces niveaux, le système peut devenir en piètre état et même mourir. Le contrôle implique :

- ♦ qu'un standard ou une norme (niveaux de performance acceptables) ait été établi;
- ♦ qu'un moyen de vérifier la performance actuelle soit en place;
- ♦ qu'on puisse comparer la performance actuelle avec le standard établi afin de détecter, d'étudier et de corriger tout écart;
- ♦ qu'un moyen d'obtenir de la rétroaction soit prévu (un système fonctionne tant qu'il reçoit des informations régulatrices adéquates).





**Exemple**

*D'habitude, le système de paie d'une entreprise produit la paie périodiquement (à la semaine ou aux deux semaines). Plusieurs contrôles comme par exemple celui des absences et celui du temps supplémentaire permettent au système de s'ajuster pour produire une rémunération cohérente et adéquate. Les conventions collectives, les politiques de l'entreprise, les lois et autres (par exemple, les tables des d'impôts) fixent les niveaux. Des contrôles sont exercés comme par exemple, la validation de la feuille de temps de l'employé par son supérieur.*

L'informaticien doit garder à l'esprit de produire des systèmes qui puissent s'auto-adapter. Cet objectif peut être atteint en paramétrant les systèmes et en les produisant pour qu'ils nécessitent le moins d'interventions extérieures.

**Exemple**

*Un bon système de paie doit permettre d'ajuster facilement des paramètres tels que les taux d'impôt, sans que l'on ait à reprogrammer son logiciel.*

### 1.3.4 Évolution

Comme les besoins évoluent dans le temps, un système nécessitera des modifications, des entretiens et des améliorations tout au long de son existence. En principe, plus un système est spécialisé et moins il peut s'adapter à différentes circonstances. Très souvent, plus la taille d'un système augmente et plus il faudra attribuer de ressources pour sa maintenance ou son opération.

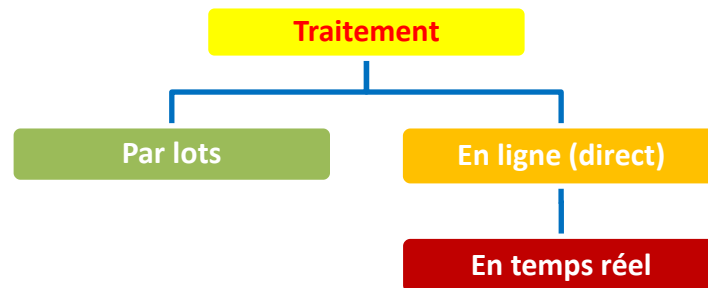


**Exemple**

*On effectue la maintenance d'un logiciel afin de tenir compte de nouveaux besoins et de prolonger sa durée de vie. Ce principe est à la base des versions du logiciel.*

### 1.3.5 Mode de fonctionnement

Divers types de traitements contribuent au fonctionnement d'un système. En informatique, on retrouve :



#### 1.3.5.1 Traitement par lots (BATCH) ou séquentiel

- Le traitement et la mise à jour des données y sont effectués de façon séquentielle, tout comme la recherche des informations.
- Les modifications à apporter sont habituellement accumulées durant une certaine période (jour, semaine, mois, ...) en vue d'effectuer la mise à jour à un moment donné ; durant ce temps, les données peuvent perdre de leur actualité.

**Exemples**

*Production d'états de compte, production de bulletins scolaires, préparation de la paie, etc.*

#### 1.3.5.2 Traitement en ligne (ON-LINE) ou direct

- Le système accepte les entrées directement de l'endroit où elles sont créées ou à distance ; il retourne les sorties là où elles sont requises ;
- Les données sont organisées de façon à pouvoir les retrouver et les modifier rapidement sans nécessairement avoir à accéder à d'autres données du système (accès direct).

**Exemples**

*Gestion des dossiers des clients, gestion des fournisseurs, guichet automatique, etc.*

### 1.3.5.3 Traitement en temps réel (REAL TIME)

- Cette variante des systèmes en ligne s'en distingue par le fait qu'il y a **contrôle d'un environnement.**
- Le système reçoit des données, les traite et retourne suffisamment rapidement (millisecondes ou microsecondes) les résultats afin d'affecter l'environnement à ce moment.

#### Exemples

*Systèmes de contrôle de processus (raffinerie de pétrole, etc.), systèmes d'acquisition de données à haute vitesse (satellites, etc.), systèmes de contrôle de missiles, systèmes d'aiguillage téléphonique, systèmes de surveillance des patients, système de sécurité, etc.*

## 1.4 Systèmes d'information

Un système d'information est un ensemble d'éléments (personnes, procédures, méthodes et techniques, équipements informatiques ou non) qui saisissent, stockent, transforment et diffusent des données. Des données sont émises par une ou plusieurs sources et traitées par le système, lequel utilise aussi des données entreposées préalablement. Les résultats du traitement sont transmis à une ou plusieurs destinations ou mettent à jour des données entreposées. Pour la réalisation des traitements, le système utilisera ou non des technologies de l'information plus ou moins sophistiquées.

Le système d'information devrait viser à **fournir la bonne information (pertinente, précise, complète), au bon moment, à la bonne personne.**

### 1.4.1 Systèmes d'information formels versus informels

Les systèmes d'information sont constitués de l'ensemble des activités de traitement d'information que sont l'envoi et la réception de lettres et de notes de service, les conversations téléphoniques, les discussions imprromptues, les messages de courrier électronique, les notes aux tableaux d'affichage, l'information recueillie en consultant divers sites du World Wide Web ou des articles de journaux et de magazines, etc. Dans une entreprise ou organisation, on distingue deux types de systèmes d'information qui se côtoient : formels et informels. Un **système d'information formel** comporte généralement un ensemble de règles et de méthodes de travail dûment documentées ou à tout le moins établies selon une tradition; un **système d'information informel** est tout le contraire (l'information y est obtenue sans respect des voies officielles).

### Exemples

#### Système d'information **formel**

*Un système classique de paie saisit les données au sujet du temps travaillé, les transforme à l'aide de données entreposées dans des bases de données, produit des chèques de paie ou effectue des dépôts automatiques aux comptes de banque des employés et transmet de l'information au sujet des montants versés. Bien qu'il existe sans doute encore des systèmes de paie manuels, la grande majorité des systèmes de paie modernes utilisent des technologies de l'information incluant bien sûr des ordinateurs, mais aussi des bases de données et des réseaux de télécommunications.*

*Autres : système de paiement aux fournisseurs, système comptes-clients, analyse des ventes et établissement de budgets, système permettant d'évaluer les aspects financiers de diverses possibilités d'acquisition et même système expert permettant le diagnostic organisationnel*

#### Système d'information **informel**

*Un président d'entreprise prend, dans son agenda personnel, des notes sur le comportement de ses proches collaborateurs, leur efficacité, leur degré d'autonomie, et les utilise au moment des promotions, des offres de participation aux bénéfices ou des augmentations de salaires. Dans un tel cas, le président est à la fois le producteur de l'information et sa principale destination. La « technologie de l'information » utilisée est par contre relativement primitive, puisqu'elle consiste essentiellement en un agenda personnel (à moins qu'il ne s'agisse d'un agenda électronique).*

## 1.4.2 Types de systèmes d'information

Comme les organisations réalisent diverses activités et que celles-ci varient grandement d'une organisation à l'autre, il existe divers types de systèmes d'information. Examinons brièvement les principaux.

### 1.4.2.1 Systèmes de traitement des transactions ou systèmes de traitement transactionnel

- assistent ou **améliorent les activités opérationnelles** liées à la raison d'être de l'organisation ;
- sont responsables de **l'emmagasiner des données** (gros volume) nécessaires au suivi des activités de l'organisation ;
- **traitent les données qui proviennent des transactions** souvent semblables que l'organisation effectue avec les clients, les fournisseurs, les créanciers, les employés, etc. ;
- **produisent les documents et pièces justificatives** qui témoignent de ces transactions ;
- tendent à **remplacer leurs procédures manuelles par un traitement automatisé** afin que ces activités répétitives soient plus efficaces, plus rapides et plus exactes ; ils ont été parmi les premiers systèmes à être

informatisés ; ils utilisent aujourd'hui des technologies de pointe et présentent des défis importants aux développeurs ;

- fonctionnent très souvent en mode interactif ;
- possèdent des procédures bien connues et pouvant être décrites clairement ;
- comportent peu d'exceptions à la procédure normale ;
- sont utilisés par les employés effectuant les opérations ou les clients directement.

#### Exemples

*Systèmes de comptabilité, de gestion des stocks, de paie, de prise de commandes, de facturation, de comptes-fournisseurs, de comptes-clients, de mise à jour de comptes bancaires, d'inscription d'étudiants, de prêt de livres et de documents dans une bibliothèque, de calcul des impôts dus par un contribuable, etc.*

*Parmi les plus récents, on retrouve les systèmes de traitement transactionnel en ligne comme ceux de commerce électronique.*

### 1.4.2.2 Systèmes d'information de gestion

- ont pour objectif de **soutenir les activités des gestionnaires de l'organisation**, qu'elles se situent au niveau du contrôle des opérations, du contrôle de gestion ou de la planification stratégique ;
- consistent généralement en des **rapports remis aux gestionnaires**, de façon périodique ou sur demande, qui résument un aspect particulier de l'organisation ; **rapports souvent comparatifs qui mettent en contraste une situation actuelle avec une prévision ou un standard**, des données présentes avec des données historiques et des données à propos d'entreprises du même secteur d'activité ;
- **reposent souvent sur les bases de données des systèmes de traitement des transactions**, bien qu'ils aient aussi des sources externes à l'organisation (données bien identifiées et connues à l'avance) ; la qualité de l'information qu'ils produisent est largement tributaire du bon fonctionnement des systèmes de traitement des transactions ;
- impliquent généralement des traitements en différé ;
- sont employés par les gestionnaires de l'organisation.

#### Exemples

*Systèmes d'analyse de performance des vendeurs, de suivi budgétaire, de suivi de la productivité ou de l'absentéisme, les études de marché, etc.*

### 1.4.2.3 Tableaux de bord de gestion

- partagent le même objectif que le système d'information de gestion ;

- mettent l'accent sur :
  - 1) la surveillance de base des résultats, du fonctionnement et des activités,
  - 2) le repérage et la localisation des problèmes correspondants,
  - 3) l'analyse des écarts entre les résultats prévus et ceux obtenus,
  - 4) la comparaison dans le temps et le balisage (comparaison avec d'autres organisations ou unités organisationnelles),
  - 5) l'analyse d'options de solution et la prise de décision,
  - 6) la simulation et l'optimisation,
  - 7) la prévision et la planification
- conçus pour fournir de l'information «de façon sommaire et ciblée», en général sous forme de « flash » accompagnés de reportage ventilé ou synoptique ; offrent une vue d'ensemble avec des détails, mais au besoin seulement ;
- constitués d'un certain nombre d'indicateurs essentiels et pertinents ;
- mettent en évidence les résultats significatifs, les exceptions, les écarts et les tendances ;
- fournissent à son utilisateur un modèle mental cohérent en regroupant les indicateurs de façon à les placer dans son esprit ;
- présentent les indicateurs sous une forme compréhensible, évocatrice et attrayante, pour en faciliter la visualisation ;
- outils de travail plutôt que simples producteurs d'information, la convivialité de leur interface usager est primordiale.

#### Analogie

*Un tableau de bord peut être comparé à un document de 1000 pages qui contient toute l'information dont le gestionnaire a besoin pour faire le suivi des processus dont il a la responsabilité. Cependant, si on lui fournit quotidiennement un rapport de 1 000 pages, il ne le consultera sans doute pas, faute de temps ! Pour être utile, le tableau de bord doit pouvoir être facilement consulté.*

*Les pages 101 à 1 000 représentent la base de données volumineuse, qui contient les données détaillées parmi lesquelles on peut extraire un élément particulier et qui constitue surtout une source de données «de base» servant de cumul de statistiques.*

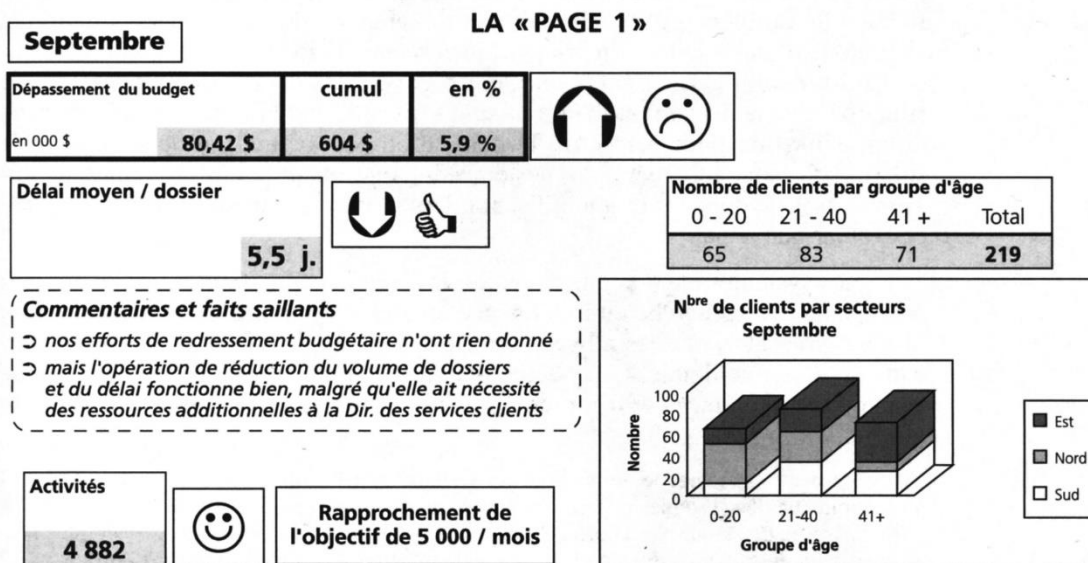
*Les pages 11 à 100 sont dix fois moins volumineuses et représentent les «statistiques» que l'on peut fournir, par exemple, en annexe d'un rapport annuel. Ce ne sont plus les données brutes, mais elles sont encore très détaillées et elles le sont probablement trop pour fournir une perspective d'ensemble cohérente.*

*Les pages 2 à 10, elles aussi, dix fois moins volumineuses que les précédentes, constituent la partie dynamique plus détaillée ou ventilée du tableau de bord. Nous y retrouvons des tableaux croisés, des graphiques pouvant s'apparenter aux rapports statistiques, mais qui ont la particularité d'être concis et de porter sur l'essentiel. Les «pages 2 à 10» peuvent parfois être constituées et porter le nom de «reportage synoptique », surtout si elles sont une extension des « sorties » de systèmes d'information existants, ou un amalgame de statistiques.*

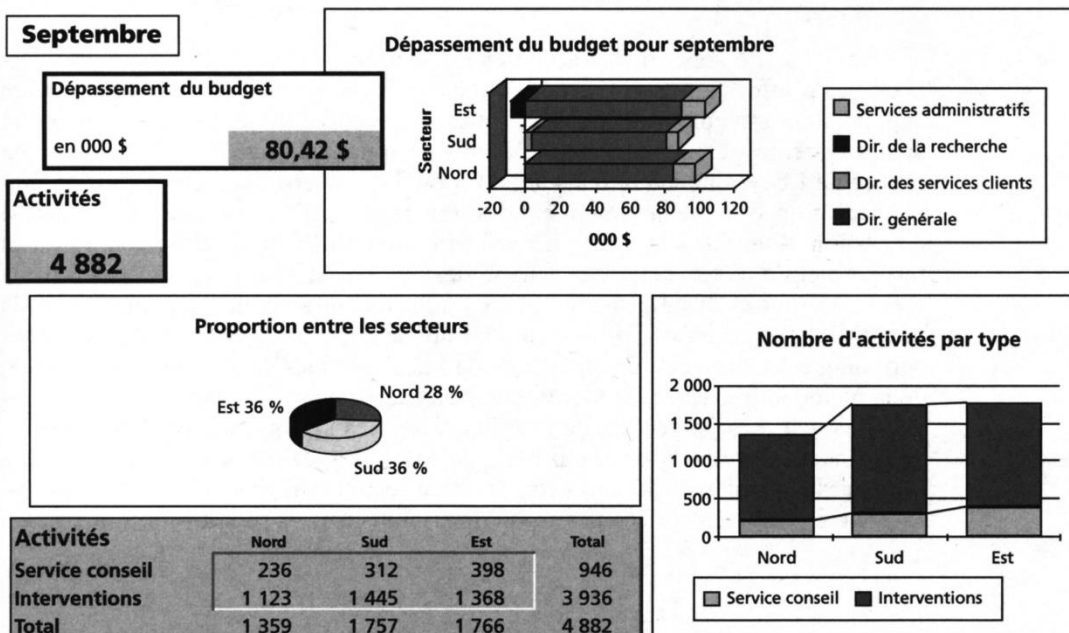


La page 1, ou page éclair, donne une vue d'ensemble des indicateurs essentiels : des résultats globaux, des graphiques simples, des pictogrammes, des clignotants avertissant d'un résultat ou d'un écart anormal, etc.

### Un exemple fictif de tableau de bord



### CERTAINS INDICATEURS VENTILÉS DES « PAGES 2 À 10 »



#### 1.4.2.4 Systèmes d'aide à la décision (S.A.D.) ou de support à la prise de décision

- conçus dans l'objectif explicite de **soutenir les activités de prise de décision** (Le processus de prise de décision est composé de trois

- grandes phases : 1- l'identification du problème, 2- l'élaboration et l'évaluation de scénarios de solution et 3- le choix d'une solution.); au lieu de prendre des décisions, ils aident le gestionnaire à prendre des décisions éclairées sur des aspects variés d'une opération ;
- fournissent l'information permettant à ceux qui prennent des décisions, d'identifier une situation où une décision doit être prise ;
  - pourvus de capacités de modélisation afin de permettre la génération et l'évaluation de scénarios de solutions ; ils permettent de simuler des situations selon le principe du WHAT-IF (que se passe-t-il si ...?) ;
  - utilisés de façon interactive et ad hoc (pas sur une base régulière) ;
  - retrouvent et affichent des données, font une variété d'analyses statistiques et mathématiques sur les données ; utilisent un ou plusieurs modèles pour représenter et évaluer une situation ;
  - possèdent très souvent des facilités pour présenter l'information sous forme de graphiques, de tableaux, de rapports ;
  - doivent être très flexibles car les procédures ne sont pas claires ou complètement identifiables à l'avance ;
  - utilisent des données dont la provenance est éparpillée ; ont accès à une ou plusieurs bases de données ;
  - sont employés par les dirigeants qui prennent les décisions.



#### Exemples

*Chiffriers électroniques, les systèmes d'analyse statistique, les programmes d'étude de marché, prévisions des ventes, etc.*

#### 1.4.2.5 Systèmes experts ou systèmes à bases de connaissances (KNOWLEDGE-BASED SYSTEM)

- originent du domaine de l'intelligence artificielle qui vise à produire des systèmes qui imitent le comportement humain dans des tâches «intelligentes» ;



- renferment les connaissances d'experts dans un domaine donné et sont capables d'expliquer les lignes du raisonnement qui conduit à leurs décisions ;
- sont composés essentiellement d'une base de connaissances (faits et règles utilisés par les experts) et d'un moteur d'inférence ;
- sont une extension des systèmes interactifs d'aide à la décision lorsque l'expertise vise la prise de décision ou un prolongement des systèmes d'aide au travail intellectuel.



#### Exemples

*Jeux d'échecs, systèmes experts médicaux, systèmes d'intervention en psychologie, etc.*

#### 1.4.2.6 Systèmes à avantage compétitif ou concurrentiel

- sont en fait des systèmes de traitement des transactions, des systèmes d'information de gestion, des systèmes d'aide à la décision ou des systèmes experts mais s'en distinguent par l'utilisation qu'on en fait ;
- une partie de l'information origine des bases de données de l'organisation mais la plus grande part provient de sources extérieures ;
- sont utilisés par les cadres pour surveiller l'environnement concurrentiel et faire de la planification stratégique ;
- leurs utilisateurs peuvent aussi être un client, un fournisseur et même une autre organisation de la même industrie, contrairement aux autres systèmes qui sont conçus pour des utilisateurs qui font partie de l'organisation (gestionnaires ou autres) ;
- peuvent permettre à l'organisation de réussir face aux forces concurrentielles représentées par les clients, les fournisseurs, les

nouveaux venus dans l'industrie, les produits de substitution et les autres organisations du même secteur d'activité.

#### Exemples

*L'installation de terminaux qui permettent aux clients d'un distributeur de produits pharmaceutiques de transmettre directement leurs commandes à l'ordinateur de leur fournisseur faisant des clients quasi captifs.*

*Un fabricant d'armoires de cuisine met à la disposition de sa clientèle un système d'aide à la décision au sujet du design d'une cuisine afin d'augmenter fortement sa part de marché.*

*Systèmes qui donnent des nouvelles sur les concurrents, fournissent des rapports boursiers, des prévisions économiques, etc.*

### 1.4.2.7 Systèmes auxiliaires de communication

- Permettent aux employés d'une organisation de communiquer entre eux et avec des personnes externes (clients, fournisseurs, etc.).

#### Exemples

*Courriel, télécopie, accès à Internet, vidéoconférence, etc.*

### 1.4.2.8 Systèmes de soutien bureautique

- Aident les employés à créer et partager des documents ;
- Permettent de gérer l'information sur les horaires de travail, les réunions, les projets, etc. ;
- Peuvent utiliser des outils informatiques (numériseurs, bases de données, etc.) et des logiciels pour effectuer la gestion électronique de documents (GED) ; très souvent, les documents sont numérisés, référencés et archivés pour en faciliter la recherche.

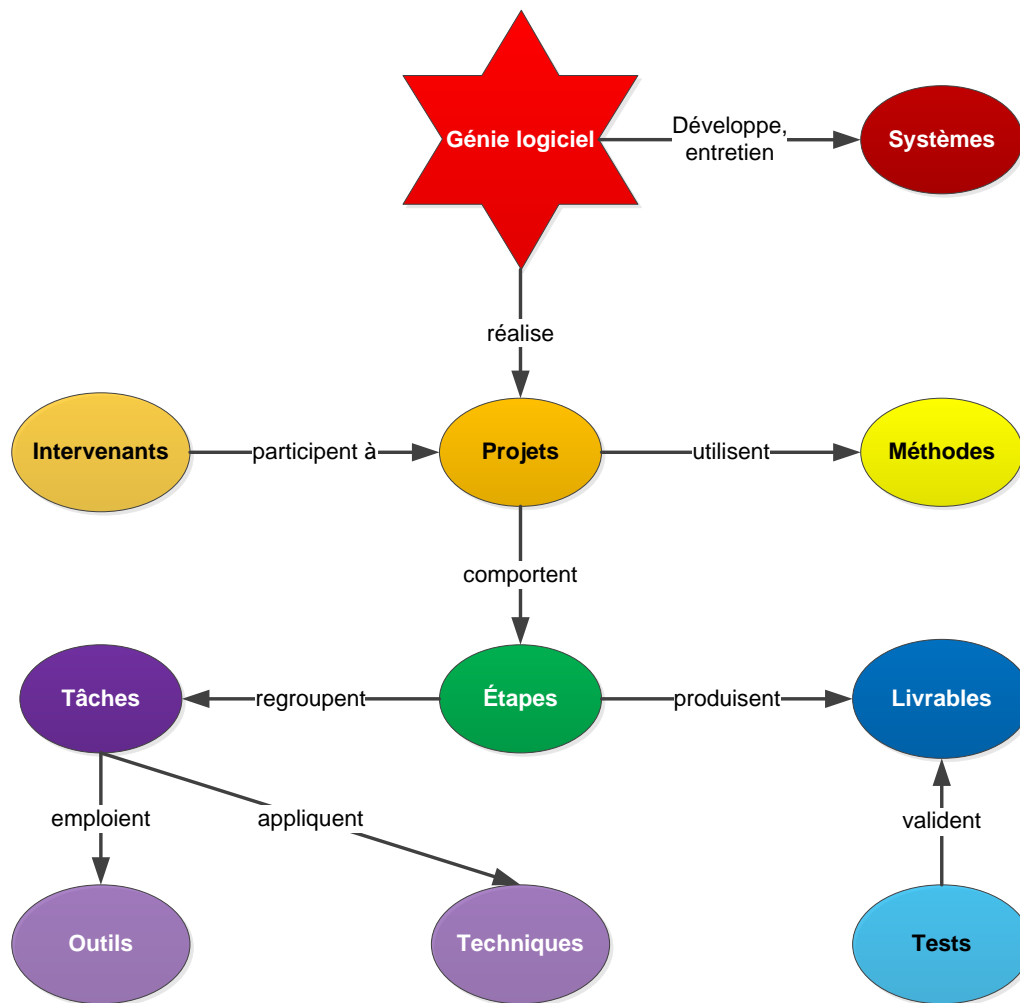


#### Exemples

*Systèmes de gestion de documents (rapports, soumissions, notes de service, etc.), agenda collectif, portail, etc.*

## 2 Génie logiciel

Dans cette section, vous découvrirez ce qu'est le génie logiciel et son implication au sein de la vie d'un système. Vous poursuivrez en comparant deux approches de développement utilisées en génie logiciel et qui sont à la base de la programmation d'une application logicielle. Vous terminerez en vous familiarisant aux concepts de base des projets réalisés en génie logiciel.



Le **génie logiciel** est un domaine qui s'intéresse au développement et à la maintenance de systèmes. En génie logiciel, un **système** est un ensemble de matériel, de logiciels, de données, de personnes et de procédures qui interagissent dans un but donné.

Le génie logiciel procède par **projets** pour réaliser le développement et la maintenance des systèmes. Au sein d'un projet, on retrouve habituellement des équipes multidisciplinaires et divers **intervenants** plus ou moins impliqués, informaticiens ou non. Question de productivité, on fait appel à des **méthodes** pour réaliser le projet de manière organisée, cohérente et contrôlable.

Un projet est subdivisé en **étapes** afin de mieux en assurer la gestion. Les étapes regroupent des **tâches** à réaliser où des **outils** sont employés et des **techniques** appliquées. Au cours de chaque étape, la réalisation des tâches conduit à la production de **livrables** qui sont validés par des **tests**.

L'explication de la plupart de ces termes suit.

## 2.1 Définitions de génie logiciel

Le génie logiciel (en anglais, *software engineering*) est une discipline qui porte sur les connaissances et **les méthodes de développement et d'entretien des systèmes logiciels**. Le génie logiciel est donc l'ensemble des méthodes et des procédures mises en œuvre lors des différentes phases de production d'un logiciel afin d'en améliorer la qualité et la maintenance.

De nos jours, l'appellation « génie logiciel » remplace de plus en plus les termes : « développement de systèmes », « informatisation » ou « analyse ». Toutefois, l'Office québécois de la langue française accepte les synonymes : « génie du logiciel », « ingénierie logicielle » et « ingénierie du logiciel ».

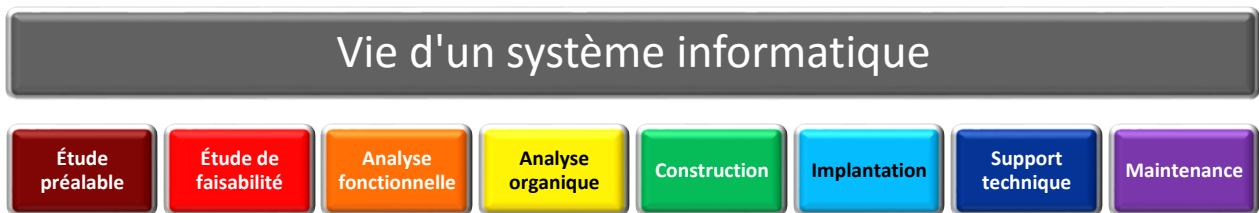
## 2.2 Vie d'un système

### 2.2.1 Tâches et étapes

En génie logiciel, on retrouve le cycle de vie d'un logiciel (en anglais *software lifecycle*) qui désigne l'ensemble des activités, de son développement jusqu'à sa disparition ; certains auteurs préfèrent « cycle de vie d'un système » parce que cette expression englobe le cycle de vie du logiciel et bien plus.

La vie d'un système est complexe et exige beaucoup de travail ; la réussite est tributaire d'une série de tâches organisées méthodiquement et dont l'aboutissement est un système d'information fiable, robuste et efficace. Même s'il existe plusieurs méthodes de génie logiciel, toutes découpent le cycle de vie ; ainsi, on peut retrouver des phases ou des étapes qui regroupent des tâches ou activités constituées d'opérations. Comme ce découpage varie d'une méthode à l'autre, les subdivisions et leurs appellations peuvent différer grandement ; toutefois, dans l'ensemble, on devrait retrouver des équivalents en termes de tâches. Ce découpage permet de valider la qualité du travail et sa conformité avec les besoins en plus de vérifier l'adéquation du processus de gestion des tâches.

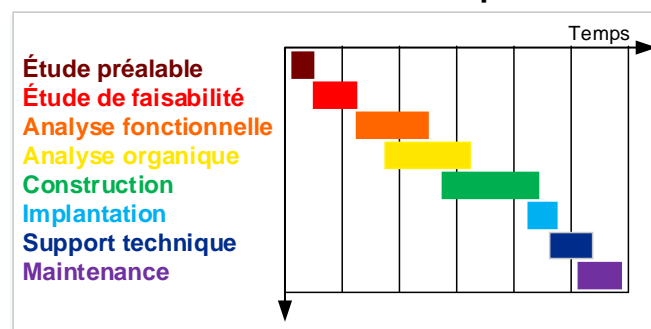
Un autre avantage à découper le cycle de vie est de permettre de détecter les erreurs au plus tôt pour ainsi maîtriser la qualité, les délais et les coûts associés.



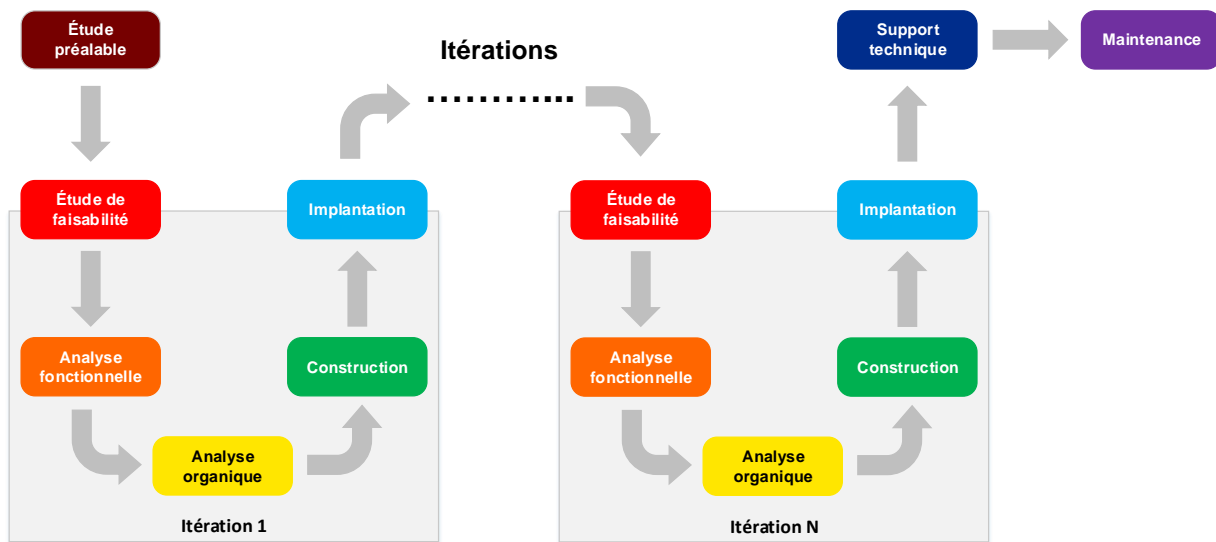
Le schéma précédent présente les étapes « génériques » que l'on retrouve dans plusieurs méthodes. Le **support technique** et la **maintenance** devraient en temps normal avoir lieu lors de l'étape d'exploitation (utilisation) du système mais ils ont été représentés distinctement afin de leur accorder plus d'importance. La **mise au rancart** devrait se retrouver à la fin mais elle n'a pas été considérée comme étant une étape parce que les tâches qui y sont effectuées (archivage et protection des données, débranchement du matériel, etc.) sont peu nombreuses et relèvent du gros bon sens.

Le diagramme indique la chronologie entre les étapes en recourant à une progression au niveau des couleurs, un peu à la manière d'un arc-en-ciel. Contrairement à ce que pourrait laisser croire le diagramme, la vie d'un système n'est pas un processus linéaire. Pour un système, certaines étapes peuvent se chevaucher partiellement ou s'effectuer en parallèle afin d'obtenir des résultats plus rapidement.

### Chevauchement d'étapes



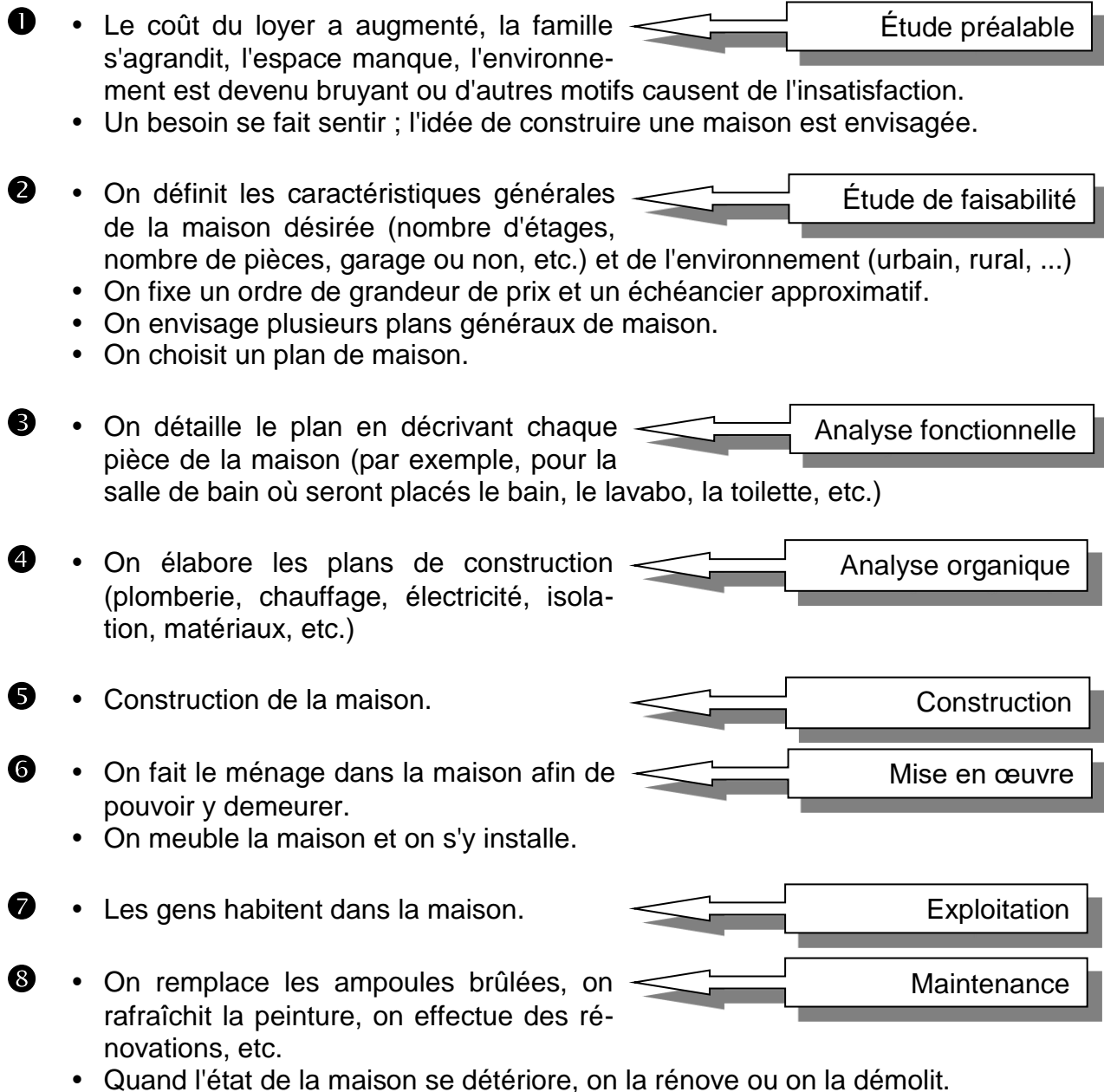
La vie d'un système est plutôt un **processus itératif** car bien souvent, il faut retourner en arrière et corriger les travaux des étapes antérieures. Lorsqu'un système devient désuet, il peut arriver que l'on décide de recommencer à zéro depuis l'étape d'étude de faisabilité pour concevoir son remplaçant. Certaines méthodes comme celles « agiles » préfèrent effectuer une partie de plusieurs étapes et de procéder de la sorte au cours d'itérations successives. Dans le diagramme suivant, les étapes d'étude de faisabilité et d'implantation sont représentées à mi-chemin sur la frontière d'une itération pour signifier qu'elles peuvent faire partie ou non de certaines itérations.



Le tableau ci-après tente de regrouper les étapes en phases et de décrire sommairement les étapes. Les regroupements en étapes peuvent différer d'une méthode à l'autre.

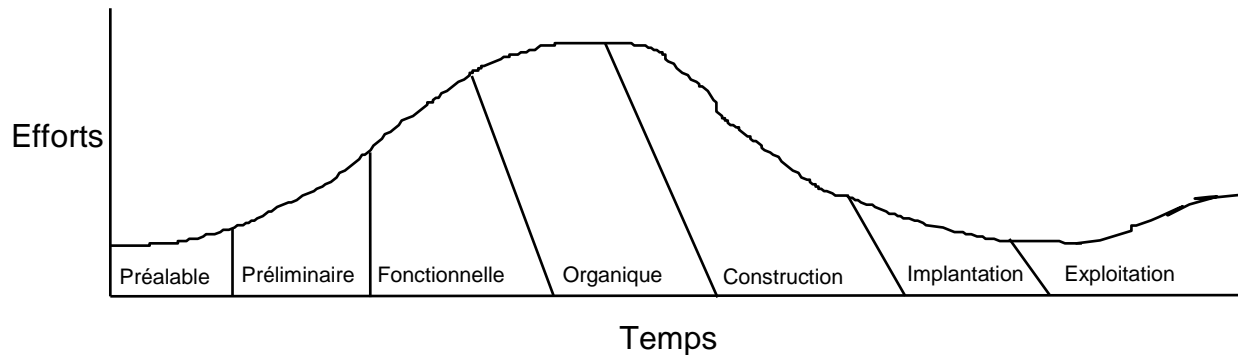
Phases		Étapes		
n°	Nom	n°	Nom	Description
1	Analyse	1.1	Étude préalable	Vérification <u>rapide</u> de la recevabilité de la demande et de la faisabilité du projet.
		1.2	Étude de faisabilité	Étude de la situation actuelle, proposition de solutions qui répondent aux besoins/ problèmes, recommandations.
2	Conception	2.1	Analyse fonctionnelle	Définition du <u>quoi</u> (traitements et données) de la solution choisie par le client (analyse fonctionnelle = <u>fonctionnement</u> du système = grandes fonctionnalités du système).
		2.2	Analyse organique	Description du <u>comment</u> de la solution à développer (analyse organique = <u>organisation</u> du système = spécifications pour la construction).
3	Construction	3.1	Construction	Réalisation technique de la solution (fichiers ou bases de données, programmes, procédures, documentation).
4	Implantation	4.1	Implantation	Installation du système (aménagement des lieux, préparatifs, installations matérielles et logicielles, configuration, entrée des données, formation) et son opérationnalisation dans un contexte réel.
5	Exploitation	5.1	Support technique	Assistance et dépannage des utilisateurs du système.
		5.2	Maintenance	Amélioration du système pour corriger des problèmes et prolonger sa durée de vie.

Afin de mieux comprendre, voici une analogie entre la vie d'un système informatique et la construction d'une maison.



Le graphique qui suit présente la séquence des étapes et un aperçu des efforts à y consacrer habituellement. On y remarque que les étapes d'analyse fonctionnelle, d'analyse organique et de construction sont celles qui nécessitent le plus d'efforts. Le graphique est plus ou moins conforme à la réalité puisque la phase d'exploitation peut à elle seule, durer plus longtemps que toutes les autres étapes réunies. Les séparations obliques dans le graphique indiquent que certaines étapes peuvent se chevaucher dans le temps alors que d'autres étapes (aux séparations verticales) doivent être entièrement complétées avant de poursuivre

avec les étapes suivantes ; dans certains projets, les séparations obliques peuvent être plus inclinées à l'horizontale afin de représenter un plus grand chevauchement des étapes.



## 2.2.2 Appellations des étapes et groupes d'étapes

On rencontre plusieurs **synonymes** pour les noms des étapes et même certaines appellations pour des groupes d'étapes. Aussi certains termes n'ont pas la même signification d'une personne à l'autre. Voici quelques équivalences.

<i>Étude préalable</i> =	démarrage = planification du projet = exploration
<i>Étude de faisabilité</i> =	étude préliminaire = étude d'opportunité = analyse des besoins = analyse et conception générale
<i>Analyse fonctionnelle</i> =	analyse et conception détaillée
<i>Construction</i> =	codage et débogage = programmation = recette = réalisation technique
<i>Implantation</i> =	mise en œuvre = installation = mise en production = déploiement
<i>Mise en œuvre</i> =	construction + implantation
<i>Maintenance</i> =	entretien
<i>Réalisation technique</i> =	analyse organique + construction
<i>Conception</i> =	étude préliminaire + analyse fonctionnelle + analyse organique + construction
<i>Développement</i> =	étude préliminaire + analyse fonctionnelle + analyse organique + construction + mise en œuvre



Certains auteurs font aussi une distinction entre analyse et conception :

#### Analyse

- étude faite en vue de discerner les différentes parties d'un tout, de déterminer ou d'expliquer les rapports qu'elles entretiennent les unes avec les autres ;
- vise à comprendre le fonctionnement du système actuel, à en diagnostiquer le problème et/ou à en identifier les tâches réalisées par les divers intervenants, en vue de proposer des améliorations ;
- fait une représentation (modèle descriptif) du problème soumis par les utilisateurs, circonscrit le domaine du problème ;
- spécifie le **QUOI** (c'est-à-dire le problème).

#### Conception

- résultat ou action d'élaborer quelque chose ;
- vise à planifier et mettre en place de nouvelles façons de faire pour remplacer ou pour compléter le système existant ;
- produit une représentation (modèle prescriptif) de la solution au problème ;
- spécifie le **COMMENT** (c'est-à-dire la solution au problème).

### 2.2.3 Approches de développement

Le développement d'un système peut s'effectuer de différentes façons ; cette diversité peut dérouter les développeurs de systèmes. Parfois, chaque société possède sa propre approche de développement. D'autres fois, différents groupes de développement qui œuvrent au sein d'une même entreprise utilisent diverses démarches et chaque employé de l'entreprise peut aussi avoir sa manière personnelle de développer des systèmes.

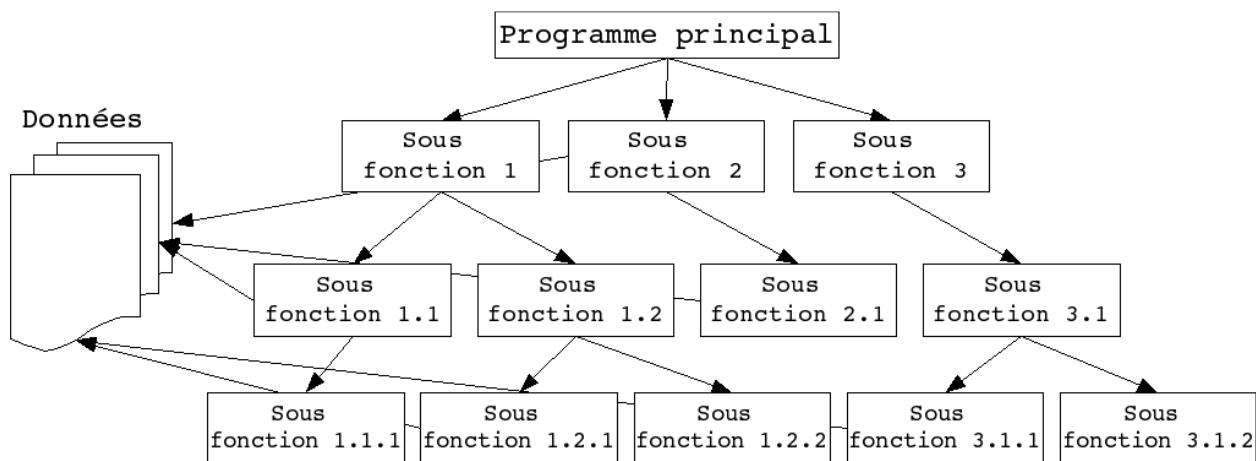
Cependant, il existe de nombreux concepts communs. À peu près tous les groupes de développement de systèmes utilisent une variante du cycle de vie du logiciel (voir la section 2.2 Vie d'un système). De plus, pratiquement toutes les approches emploient des modèles, des outils et des techniques qui, ensemble, constituent une méthode de développement. Ainsi, tous les développeurs devraient connaître les deux approches générales de développement de systèmes : l'approche structurée et l'approche orientée objets.

#### 2.2.3.1 Approche structurée

L'**approche structurée** est aussi appelée approche **classique** et approche **fonctionnelle**. Elle est constituée de l'**analyse structurée** (structured analysis), la **conception structurée** (structured design) et la **programmation structurée** (structured programming). Les méthodes

structurées trouvent leur origine dans les langages procéduraux (C, Cobol, Pascal, etc.).

Ces méthodes mettent en évidence les fonctions à assurer et proposent une approche hiérarchique descendante (TOP-DOWN) et modulaire. Elles utilisent intensivement les raffinements successifs pour produire des spécifications dont l'essentiel est sous forme de notation graphique en diagrammes de flux de données. Le plus haut niveau représente l'ensemble du problème (sous forme d'activité, de données ou de processus, selon la méthode). Chaque niveau est ensuite décomposé en respectant les entrées/sorties du niveau supérieur. La décomposition se poursuit jusqu'à arriver à des composants maîtrisables.



L'approche structurée dissocie le problème de la représentation des données, du problème du traitement de ces données. Dans le diagramme précédent, les données du problème sont représentées sur la gauche. Des flèches transversales matérialisent la manipulation de ces données par les sous-fonctions. Cet accès peut être direct (c'est parfois le cas quand les données sont regroupées dans une base de données), ou peut être réalisé par le passage de paramètre depuis le programme principal.

La SADT (Structured Analysis Design Technique) est probablement la méthode d'analyse structurée et de gestion de projets la plus connue. Elle permet non seulement de décrire les tâches du projet et leurs interactions, mais aussi de décrire le système que le projet vise à étudier, créer ou modifier, en mettant notamment en évidence les parties qui constituent le système, la finalité et le fonctionnement de chacune, ainsi que les interfaces entre ces diverses parties. Le système ainsi modélisé n'est pas une simple collection d'éléments indépendants, mais une organisation structurée de ceux-ci dans une finalité précise.

### 2.2.3.2 Approche orientée objets

L'approche orientée objets est composée de : l'**analyse orientée objets** (**OOA = Object-Oriented Analysis**), la **conception orientée objets** (**OOD = Object-Oriented Design**) et la **programmation orientée objets** (**OOP = Object-Oriented Programming**). Cette approche a commencé en Norvège dans les années 1960 avec le développement du langage de programmation Simula qui était utilisé pour créer des simulations automatisées. Dans les années 1970, le développement du langage Smalltalk a permis de résoudre les problèmes de création d'interfaces utilisateur impliquant des objets (menus déroulants, boutons, cases à cocher, boîtes de dialogue, etc.). De nos jours, on retrouve les langages C++, Java, C# et autres qui servent à écrire les définitions des types d'objets nécessaires à un système.

L'approche orientée objets considère un système d'information comme une collection d'objets interdépendants qui fonctionnent de concert pour exécuter les tâches. Il n'y a ni processus, ni programmes, ni entités de données, ni fichiers mais uniquement des objets qui répondent à des messages. La fonctionnalité du logiciel émerge alors de l'interaction entre les différents objets qui le constituent.

Un objet possède une identité, des attributs et des méthodes. L'identité distingue l'objet des autres objets et prend généralement la forme d'un identifiant (par exemple un code pour un produit, un numéro de série pour un véhicule automobile, etc.). L'attribut est une donnée qui décrit l'objet ou son état. La méthode correspond à un comportement de l'objet c'est-à-dire une opération qu'il peut réaliser ; la méthode permet de faire réagir l'objet aux sollicitations extérieures ou d'agir sur d'autres objets. Une des particularités de cette approche est qu'elle regroupe les données et leurs traitements associés au sein de l'objet ; les méthodes sont étroitement liées aux attributs, car leurs actions peuvent dépendre des valeurs des attributs ou les modifier.

Identifier des types d'objets consiste à utiliser des « classes » qui représentent des collections d'objets semblables ; un objet est une occurrence d'une classe. Aussi, chaque classe peut se subdiviser en sous-classes spécialisées qui « héritent » des caractéristiques (attributs et/ou méthodes) de leur classe supérieure. Pour représenter toutes les classes d'objets existant dans un système, l'approche orientée objets a recours au diagramme de classes.

### 2.2.3.3 Approche structurée versus approche orientée objets

L'approche structurée privilégie la fonction comme moyen d'organisation du logiciel : l'approche objet n'est pas pour autant *non fonctionnelle* puisque les méthodes d'un objet sont des fonctions. Ce qui différencie ces deux

approches, c'est que les fonctions obtenues à l'issue de la mise en œuvre de l'une ou l'autre méthode sont distinctes. L'approche objet est une approche orientée donnée : les fonctions sont déduites à partir d'un regroupement de données formant une entité cohérente, logique, tangible et surtout stable quant au problème traité. Pour sa part, l'approche structurée privilégie une organisation des données postérieure à la découverte des fonctions qui constituent les services qui répondent aux besoins.

En approche objet, l'évolution des besoins a le plus souvent tendance à se présenter comme un changement d'interaction des objets. S'il faut apporter une modification aux données, seul l'objet concerné (encapsulant ces données) sera modifié. Toutes les fonctions à modifier sont bien identifiées : elles se trouvent dans ce même objet : ce sont ses méthodes. Dans une approche structurée, l'évolution des besoins entraîne souvent une dégénérescence, ou une profonde remise en question, car la décomposition des unités de traitement (du programme principal aux sous-fonctions) est directement dictée par ces besoins ; d'autre part, une modification des données entraîne généralement une modification d'un nombre important de fonctions éparpillées et difficiles à identifier dans la hiérarchie de cette décomposition.

La modularité n'est pas contraire à l'approche structurée. Les unités de traitement, et surtout leur dépendance dans la hiérarchie sont initialement bonnes. C'est leur résistance au temps, contrairement aux modules objet, qui est source de problème. La structure d'un logiciel issue d'une approche structurée est beaucoup moins malléable, adaptable, que celle issue d'une approche objet.

Selon un grand principe de Church Turing, tout programme qu'il est possible d'écrire dans un langage peut également être écrit dans n'importe quel autre langage. Ainsi, tout ce que l'on fait avec un langage de programmation par objets peut être fait en programmation structurée. La différence entre l'approche structurée et l'approche orientée objets est purement d'ordre pratique. Comme de nombreux systèmes au sein d'une organisation utilisent les mêmes objets, les classes d'objets mises en œuvre par l'approche orientée objets peuvent être réutilisées, évitant d'avoir à les recréer.

L'approche orientée objets est la conséquence de la modularisation du logiciel, démarche qui vise à maîtriser sa production et son évolution. Malgré cette continuité logique, les langages objet ont apporté un profond changement dans l'art de la programmation : ils impliquent un changement d'attitude mentale du programmeur. Certaines personnes trouvent l'approche orientée objets difficile à comprendre ; la difficulté réside dans la création d'objets abstraits pour représenter des entités réelles (personne, véhicule, animal, etc.) ou virtuelles (activité, transaction, etc.). Pourtant, l'approche orientée objets est plus naturelle (intuitive) puisque l'être humain tend à

penser le monde en termes d'objets tangibles ; il est moins naturel de penser aux procédures complexes développées avec les langages de programmation procéduraux.

L'approche orientée objets est clairement différente de l'approche structurée ; par contre, sous certains aspects, un bon nombre de concepts traditionnels y sont simplement réaménagés. Aujourd'hui, de nombreux systèmes développés combinent les deux approches. Certains environnements de développement intégré (IDE = Integrated Development Environment) proposent aussi les deux approches dans le même outil ; par exemple, on utilise la programmation orientée objets pour l'interface utilisateur et la programmation procédurale pour le reste. Par ailleurs, de nombreux projets utilisent exclusivement l'une ou l'autre de ces approches. Ces raisons et bien d'autres expliquent l'importance pour un développeur de connaître les concepts fondamentaux de ces deux approches.

## 2.3 Projets

Comme d'autres secteurs d'activités, le génie logiciel procède par projets. Voici quelques définitions du mot « projet » extraites de divers dictionnaires qui peuvent s'appliquer en génie logiciel même si leur contexte premier est autre :

- Intention souvent floue, dont la réalisation peut être lointaine ;

### Exemples

*Projet de voyage, projet d'expansion d'une entreprise, etc.*

- Étude préparatoire, parfois exhaustive, qui sera soumise à décision ;

### Exemples

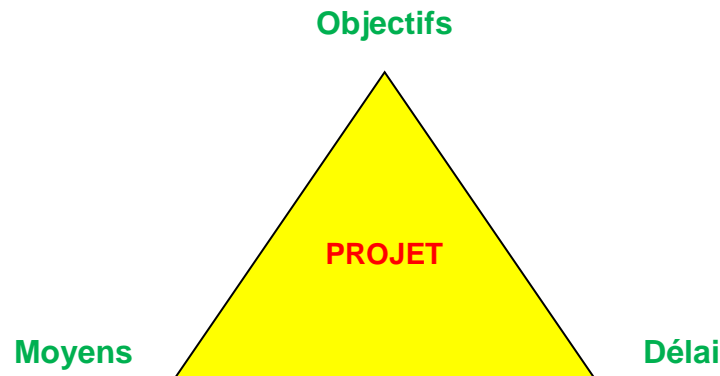
*Projet de loi, projet d'urbanisme, etc.*

- Situation qui précède une réalisation ou un état définitif.

### Exemples

*Projet de construction d'un édifice, projet de création d'un club social, etc.*

Dans un projet, on souhaite **atteindre des objectifs avec des moyens (ressources), dans un délai donné**. Les projets en génie logiciel sont influencés par divers facteurs (budgets, échéances, contraintes organisationnelles, etc.) qui doivent être considérés afin de produire des systèmes de qualité qui répondent adéquatement aux besoins.



S'inspirant du principe du **PODC (Planification, Organisation, Direction, Contrôle)** utilisé en gestion (administration), un projet en informatique comporte des activités pour :

- **Préparer les plans des activités à réaliser ;**
- **Mettre en place les ressources nécessaires ;**
- **Réaliser les activités et produire les biens livrables (voir la section 2.5) ;**
- **Contrôler l'adéquation des biens livrables et l'avancement des travaux.**

Souvent, les entreprises vont en soumission pour des projets. Cette situation donne lieu à des procédures et au moins deux documents importants :

### 2.3.1 Appel d'offres

- Procédure qui **permet à une organisation (entreprise, etc.) de sélectionner un fournisseur (soumissionnaire) pour obtenir un bien ou un service, après l'avoir mis en concurrence avec d'autres ;**
- Concrètement, un appel d'offres peut prendre la forme d'une annonce dans un journal qui invite à soumissionner sur un projet. En plus de présenter l'organisation, l'appel d'offres indique succinctement les biens ou services exigés, les coordonnées et délais, et s'il y a lieu, la façon de se procurer le cahier des charges.

### 2.3.2 Cahier des charges

- **Document écrit qui définit les caractéristiques que doivent présenter un bien ou un service à fournir pour une réalisation à l'étude ou en cours ;** entre autres, il indique les **objectifs**, les modalités d'exécution sans imposer de solutions, les **critères d'évaluation**, les **contraintes et les exclusions ;**

- Très souvent, il **fait partie d'un contrat administratif** où sont déterminées les obligations réciproques de l'organisation (entreprise, etc.) et de son contractant.

## 2.4 Méthodes et approches

Au fil du temps, l'informatique s'est intégrée à la plupart des sphères de l'activité humaine et la demande n'a cessé de croître. Les informaticiens ont été contraints de développer des moyens de formalisation et d'organisation des connaissances pour manipuler, traiter et conserver l'information sous toutes ses formes, en vue de produire des systèmes. C'est ce qui a conduit à l'apparition de méthodes de génie logiciel.

Le mot **méthode** vient du grec « methodos » qui signifie poursuite ou recherche d'une voie à réaliser quelque chose. Ce mot est formé des racines grecques : « métis » qui signifie « **raisonnement rusé** ou **ruses de l'intelligence** » et « hodos » qui veut dire « **chemin, voie** ».

Par abus de langage, le terme « méthodologie » (ensemble des méthodes et des techniques d'un domaine particulier) est parfois utilisé à la place de méthode.

Une méthode est un ensemble de principes, de règles et d'étapes, ordonné de manière logique et qui constitue un moyen pour parvenir à un résultat. C'est une manière de mener, selon une démarche raisonnée, une action, un travail, une activité ou une technique. En génie logiciel, une méthode est un ensemble de règles et de définitions qui décrit une démarche organisée, cohérente et contrôlable pour produire un système et ses biens livrables.

### Exemples

SADT (**S**tructured **A**nalysis and **D**esign **T**echnique), Merise, RUP (**R**ational **U**nified **P**rocess, **XP** (**eX**treme **P**rogramming), Scrum, etc.



L'utilisation d'une méthode comporte des **avantages** :

- **Augmente la productivité des développeurs ;**



- Favorise un même savoir-faire (techniques, outils) qui assure une certaine cohérence et continuité du développement ;
- **Produit des biens livrables normalisés ;**
- **Contribue à la qualité du produit ;**

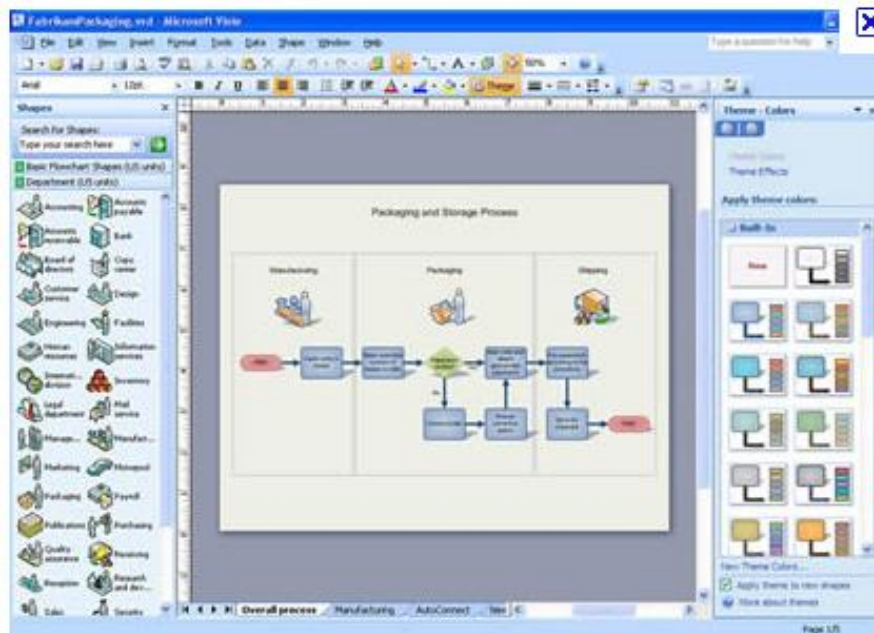
et des **inconvénients** :

- Alourdit parfois le processus et **peut entraîner des coûts ;**
- Peut porter à **surévaluer la capacité à développer un système ;**
- **Affecte parfois la motivation** en limitant la créativité et les initiatives personnelles ;
- Crée un **manque de discipline** pour appliquer correctement la méthode ;
- **Empêche le changement rapide de méthode.**

Une méthode n'est pas une recette que l'on peut appliquer les yeux fermés et qui garantit le succès d'un projet ; il faut définir une **démarche** (adaptation d'une méthode dans un contexte particulier) pour rendre la méthode utile et efficace. En fonction de l'organisation, de sa culture, de ses objectifs, du contexte humain, budgétaire et technologique du projet, on choisira les tâches pertinentes d'une méthode. L'utilisation d'une méthode fait appel à des outils et des techniques.

### 2.4.1 Outils

Un outil est un instrument utilisé par un être vivant directement ou par le truchement d'une machine afin d'exercer une action ou de réaliser une activité. Dans le contexte du génie logiciel, l'outil est souvent un logiciel.





### Exemples

*Questionnaire pour une entrevue*

*Application de dessin et de graphisme (création de diagrammes, images)*

*Traitement de texte ou éditeur de texte*

*Environnement de développement intégré (en anglais **IDE** = **I**ntegrated **D**evelopment **E**nvironment) qui comprend de nombreux outils d'aide à la programmation (éditeur intelligent, aide contextuelle, outil de débogage)*

*Logiciel de gestion de projet (planification et suivi du projet [tâches, ressources, calendrier])*

*Application de base de données (qui peut stocker l'information relative au projet)*

*Générateur de code (produit des programmes à partir de spécifications ou de diagrammes)*

*Outil de rétroingénierie (en anglais **reverse engineering**, pour décompiler un programme sans documentation et obtenir un modèle permettant au développeur de connaître les fonctionnalités)*

#### 2.4.1.1 Outil GLAO

Les **outils GLAO** (GLAO = **G**énie **L**ogiciel **A**ssisté par **O**rdinateur [en anglais CASE = **C**omputer **A**ided **S**oftware **E**ngineering]) sont des logiciels qui ont été développés pour assister les informaticiens lors de la fabrication de logiciels. Ils constituent des moyens pour produire des composantes du système ou réaliser des tâches du projet.

Un outil GLAO peut couvrir une ou plusieurs étapes de la vie d'un système. Plusieurs appellations permettent de le préciser un peu plus :

- **GLAO intégré** : couvre une grande partie de la vie d'un système sans être vraiment complet ;
- **GLAO en modules** : le module se concentre sur une fonction pointue et peut s'interfacer avec d'autres GLAOs ;
- **GLAO supérieur** (Upper CASE) : porte sur les premières étapes de la vie d'un système (planification, conception, architecture et analyse) ;
- **GLAO inférieur** (Lower CASE) : porte principalement sur la construction (programmation) ou la maintenance ;
- **GLAO horizontal** : sert pour une tâche spécifique (par exemple, pour programmer) au niveau d'une étape de la vie du système ;
- **GLAO vertical** : utilisé tout au long des projets (par exemple, pour produire des documents, pour gérer les activités du projet) ;

Voir le document **GLAO.pdf** pour plus de détails.

### 2.4.1.2 Atelier de génie logiciel

L'« atelier de génie logiciel » (AGL) peut être défini comme :

- un ensemble d'outils logiciels pour l'informatisation des systèmes d'information. Chaque outil sert de support pour une ou plusieurs tâches dans le cycle de vie. L'objectif final est de construire des systèmes opérationnels. Certains incluent aussi les matériels comme faisant partie de l'AGL.
- un ensemble cohérent de composants (outils) qui au sein d'une même structure technique, possèdent des interfaces de dialogue semblables et souvent une base de données commune ou référentiel, et qui communiquent entre eux.

L'A.G.L. repose sur une ou plusieurs méthodes qu'il automatise. C'est ce qui le distingue des outils de génie logiciel épars et isolés et qui explique pourquoi la formation à un AGL doit être précédée d'une formation aux méthodes.

Certaines entreprises constituent leur AGL « maison » en regroupant divers outils GLAO comme un simple éditeur de textes ou de graphiques, un générateur de code ou de bases de données, etc. La mise en place de l'AGL permet d'obtenir les avantages qu'offre l'utilisation des méthodes :

- la diminution des temps de conception et de réalisation,
- la coordination des équipes (partage d'informations, etc.),
- l'amélioration de la qualité des travaux (rigueur, formalismes, etc.),
- le fait de rendre la maintenance plus facile.

Voir le document **GLAO.pdf** pour plus de détails.

### 2.4.2 Techniques

Une technique réfère au savoir-faire appliqué dans la réalisation d'une tâche ou d'une activité.

Dans le cadre du génie logiciel, une technique est un ensemble de lignes directrices conçues pour aider au développement ; la technique, comprend des instructions pas à pas ou des conseils plus généraux. Il arrive qu'une technique s'applique à une étape entière du cycle de vie d'un système ou à plusieurs d'entre elles (voir la section 2.2 Vie d'un système).



#### Exemples

*Techniques de planification stratégique  
Techniques de gestion de projets  
Techniques d'interview  
Techniques de modélisation de données  
Techniques d'analyse structurée  
Techniques de conception structurée  
Techniques de programmation structurée  
Techniques de tests de logiciels  
Techniques d'analyse et de conception par objets*

## 2.5 Biens livrables

Le « bien livrable » (deliverable en anglais) est aussi appelé « livraison » ou « livrable » dans certaines méthodes. Il correspond à un produit ou service réalisé à un certain moment au cours de la vie d'un système et qui est remis au client (propriétaire du système ou autre personne qu'il a mandaté), bien souvent, en contrepartie d'une rémunération. Contrairement à ce qu'on pourrait croire, les biens livrables ne se limitent pas seulement à des rapports et par conséquent peuvent prendre d'autres formes.

#### Exemples

*Lors de la phase d'analyse ce sont surtout des rapports qui sont produits. Par contre, la phase de construction livre le logiciel, la phase d'implantation offre de la formation aux utilisateurs et la phase d'exploitation fournit des interventions de support technique.*

La notion de « livrable » peut s'appliquer au sein de plusieurs activités de la vie courante. Un rapport de laboratoire, une présentation orale, un examen écrit d'un étudiant peuvent être considérés comme des livrables.

Dans le domaine du génie logiciel il existe quatre types de biens livrables :

- les rapports qui rendent compte à la fin d'une étape et présentent le plan de la suite ;

#### Exemples

*Rapport de l'étude préalable, rapport de mise en œuvre, etc.*

- les descriptions qui présentent un ensemble d'éléments pour approbation ;

**Exemples**

*Budget, échéancier, description des interfaces, spécifications des données, etc.*

- les devis qui spécifient le détail d'éléments (structurogrammes, algorithmes, etc.) pour la poursuite des travaux ;

**Exemples**

*Dossiers de programmation*

- les produits qui sont des composantes complètes et utilisables du système.

**Exemples**

*Programmes, fichiers, documentation, formation, etc.*

## 2.6 Intervenants dans un projet de développement

Un projet de développement de système, selon sa nature, la culture de l'organisation, etc. implique la participation de divers types d'intervenants. Le nombre d'intervenants dans un projet varie selon l'ampleur et la complexité du projet, etc. Il faut noter qu'une même personne peut, selon les circonstances, appartenir à plus d'un groupe.

### Direction - administrateurs – décideurs

- responsables du financement du projet, de l'attribution de ressources au développement et à l'exploitation ;
- ont le pouvoir d'influencer le développement (l'ensemble du projet) ;
- interviennent autant dans la sélection des processus et des systèmes que dans la définition des objectifs à poursuivre ;
- souvent, haute direction d'une organisation ;

### Direction du projet – gestionnaires

- responsables des orientations à prendre concernant le développement des systèmes d'information dans l'organisation (plan directeur, stratégies de développement, dépenses et contrôles dans les projets) ;
- émettent la demande ;
- établissent l'information de base du système ;
- supervisent le processus de développement et/ou l'opération du système ;

- représentants des décideurs, à un niveau hiérarchique inférieur ;
- travaillent en collaboration avec les concepteurs ;

### Utilisateurs - clients

- personnes qui bénéficient du système pour la réalisation de leurs tâches ; interagissent avec le processus et le système par nécessité ou par choix ;
- définissent et expriment leurs besoins ;
- valident les biens livrés par les développeurs ;
- peuvent dans certains cas être les clients de l'entreprise au lieu des employés ;
- peuvent être usagers-opérateurs (produisent les entrées ou les sorties du système, sont directement associé au processus et au système d'information) ou usagers indirects (subissent les effets du système, en utilisent les entrées/sorties, sont en contact direct avec le système pour de courtes périodes de temps) ;
- usagers de guichets automatiques, voyageurs effectuant une réservation de billets d'avion, gestionnaires pour qui des rapports sont produits, etc. ;

### Informaticiens – concepteurs

- personnel plus ou moins technique qui analyse, conçoit, réalise et gère le développement, développe et implante le processus et le système d'information en collaboration avec les décideurs et les gestionnaires ;
- souvent, ils voient aussi à l'exploitation et à la maintenance du système ;
- parfois employés de l'entreprise, parfois conseillers provenant de firmes externes ;
- la plupart des équipes comptent parmi leurs membres des représentants de la population utilisatrice (gestionnaires ou usagers-opérateurs) qui agissent à titre d'experts du processus à l'étude ;
- analystes, programmeurs/analystes, programmeurs, techniciens en informatique, administrateurs de bases de données, administrateurs de réseau, chef de projet, etc. ;
- au cours des dernières années, **intégrateurs** (nouveau rôle); possèdent une expertise fonctionnelle des progiciels à adapter et à implanter (à partir d'un besoin d'affaire identifié par le client, suggèrent la façon la plus simple et la plus efficace de le supporter avec le progiciel); possèdent beaucoup d'expérience en implantation (méthodes, phases, activités, livrables, outils, approches de documentation, etc.); «challengent» les clients sur leurs façons de faire en se basant sur des projets antérieurs menés chez d'autres clients ;

### Spécialistes ad hoc

- règlent les problèmes imprévus, complexes ou exigeant une grande expertise dans un secteur d'activité (spécialiste de réseaux, notaire, avocat, comptable, chimiste, etc.) ;

### **Fournisseurs de services et de biens – formateurs (entraîneurs)**

- distributeurs de matériel informatique, de logiciels, de fournitures et autres produits nécessaires au développement et à l'exploitation ;
- enseignent aux usagers comment utiliser et gérer le système (la formation est un service) ;
- peuvent offrir du support technique aux utilisateurs ou au personnel responsable du bon fonctionnement du système.

De nos jours, très souvent, l'équipe qui intervient au niveau d'une étape est multidisciplinaire ; fini le temps où une personne voulait tout faire. De plus, certaines méthodes de génie logiciel telles que les méthodes agiles, accordent une grande importance à l'équipe et la considère comme le cœur du projet.

### 3 Conclusion

On peut établir une analogie entre le génie logiciel et l'art. En génie logiciel, il faut amalgamer divers éléments pour réaliser un système (œuvre). Dans la palette de ces artistes on peut retrouver les budgets, les échéanciers, les personnes, les idées, les procédures et la documentation, le matériel et les logiciels et tout l'aspect administratif de gestion de projet.

