

## I 5

# Qualité logicielle Système d'Information

- **Introduction**
  - Définitions normalisées
  - La démarche par le verbe
  - Les coûts
  - Le projet "Iceberg"
  - Suite et caricature !
- **La mise en œuvre : une vision !**
- **Les principales normes**
- **Qualité logicielle**



Une fine équipe ....



# 1-1 Introduction

---

*Stricte satisfaction du besoin exprimé dans le respect :*

- + *Des délais*
- + *Et du budget*

*La définition de l'ISO 9001:2000 de la qualité est :*

- + *"Aptitude d'un ensemble de caractéristiques intrinsèques à satisfaire des exigences."*



Dans un premier temps, et pour rester "pratique", la qualité peut se résumer à des principes relativement simples et pragmatiques :

- + *Dire ce que l'on va faire,*
- + *Faire ce que l'on a dit,*
- + *Vérifier que ce que l'on a fait correspond à ce que l'on avait dit.*

*Faire bien du premier coup !*

*Ensemble des caractéristiques d'une entité qui lui confère l'aptitude à satisfaire des besoins exprimés ou implicites*

# 1-1 Introduction : Définition normalisées

---

## **Qualité :**

La qualité est définie comme l'ensemble des traits et caractéristiques d'un produit ou d'un service qui leur confèrent l'aptitude à satisfaire les besoins exprimés ou implicites (NF X50-120)

*La politique qualité définit les orientations et objectifs généraux d'une entreprise en ce qui concerne la qualité, tels qu'ils exprimés par la direction générale.*

## **Assurance Qualité :**

L'assurance qualité est définie comme l'ensemble des actions préétablies et systématiques nécessaires pour donner la confiance appropriée en ce qu'un produit ou service satisfera aux exigences relatives à la qualité. (NF X50-120)

*En fait les moyens que l'entreprise juge nécessaire pour parvenir à la qualité souhaitée. En matière de SI, ces dispositions s'attachent à définir un cadre de production, de mise en œuvre et d'exploitation des applications.*

## **Système Qualité :**

Le système qualité est défini comme l'ensemble de la structure organisationnelle, des responsabilités, des procédures, des procédés et des ressources pour mettre en œuvre la gestion de la qualité. (NF X50-120)

# 1-1 Introduction : Définition normalisées

---

## **Surveillance de la Qualité :**

La surveillance de la qualité est définie comme la vérification et le suivi permanents de l'état des procédures, méthodes, conditions d'exécution, procédés, produits et services et l'analyse des résultats engendrés par comparaison au référentiel, en vue de s'assurer que les exigences spécifiées pour la qualité sont en voie d'être remplies.

La surveillance de la qualité peut être exercée par le client ou en son nom pour garantir que les exigences contractuelles sont en voie d'être satisfaites. Dans certains cas, la surveillance doit prendre en considération les facteurs qui peuvent conduire à une détérioration ou à une dégradation avec le temps (NF X50-120).

## **Audit Qualité :**

L'audit qualité est défini comme un examen méthodique et indépendant en vue de déterminer si les activités et résultats relatifs à la qualité, satisfont aux dispositions préétablies et si ces dispositions sont mises en œuvre de façon effective et sont aptes à atteindre les objectifs. (NF X50-120)

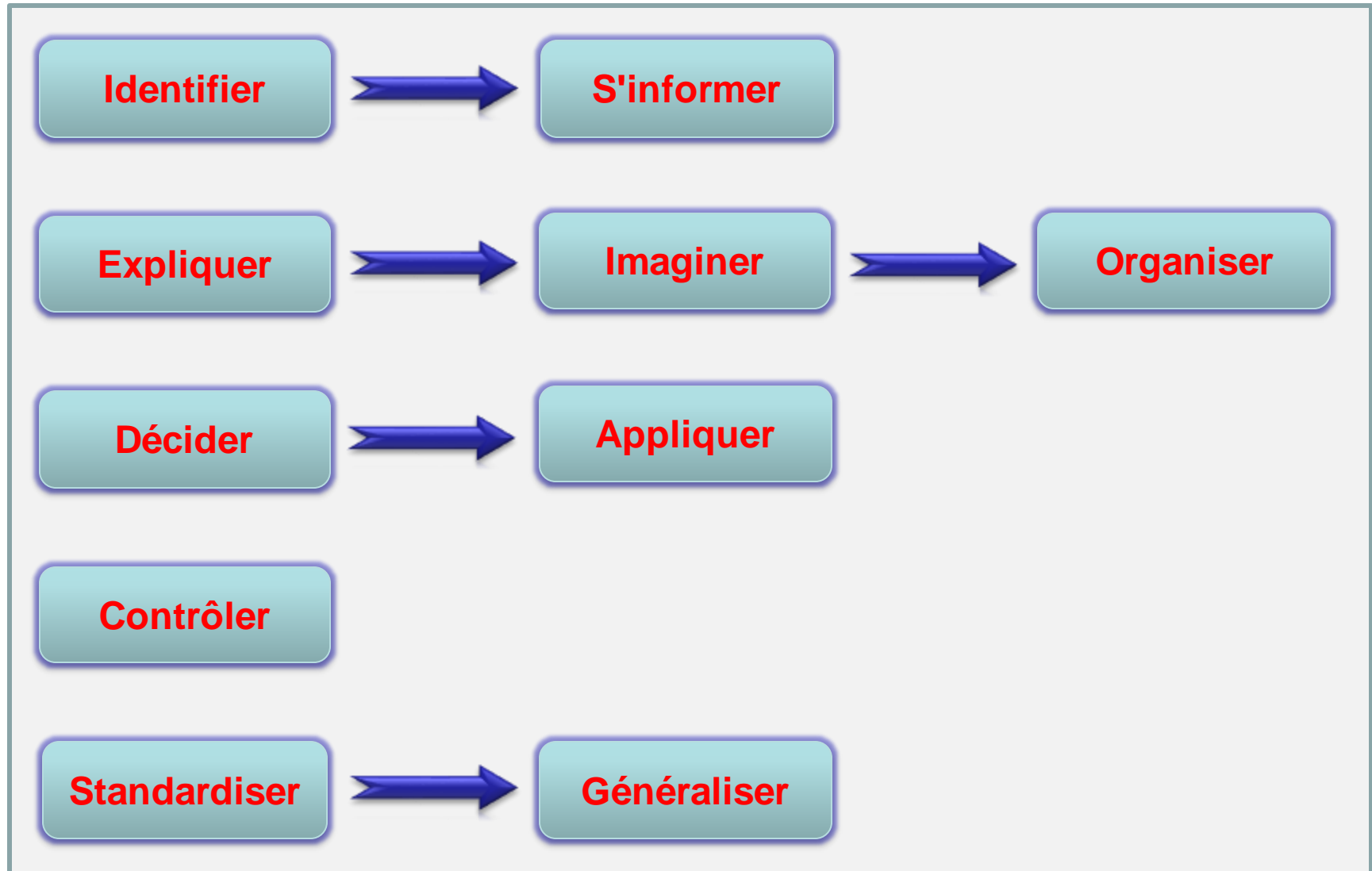
## **Non-conformité :**

Une non-conformité correspond à la non satisfaction des exigences spécifiées. La définition s'applique à l'écart ou à l'inexistence d'une ou plusieurs caractéristiques qualité ou d'éléments d'un système par rapport aux exigences spécifiées. (NF X50-120)

# 1-1 Introduction – La démarche par le verbe

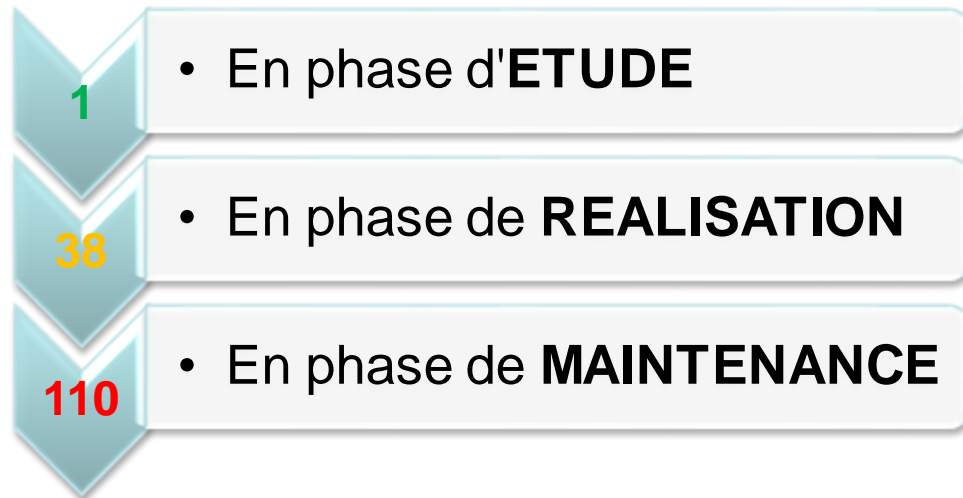
---

Les principes de la démarche qualité : Les verbes !



# 1-1 Introduction – Les coûts

## Une erreur coûte !



<b>Correction Erreur (*)</b>	<b>Coût</b>
Etude préalable	<b>1</b>
Etude détaillée	<b>3</b>
Etude technique	<b>10</b>
Réalisation	<b>20</b>
Mise en œuvre	<b>50</b>
Maintenance	<b>100</b>

L'arlésienne du développeur ! : **Les excuses ...**

- ✚ J'ai pas le temps,
- ✚ J'ai pas les moyens,
- ✚ Je ne suis pas aidé,
- ✚ Y a pas d'argent,
- ✚ De toutes façons, j'ai pas la formation !

(\*) *C'est assez curieux : on sait ce que coûte les tests lorsqu'ils sont réalisés, mais on ne cherche pas ce que coûterait l'absence de tests et ses conséquences dans un projet !*



# 1-1 Introduction – Les coûts

## Toujours l'arlésienne : le programmeur !

<b>Pervers</b>	1 Je code tout de suite 2 Je jette mon travail et je recommence en 1
<b>Médiocre</b>	Pas le temps pour la conception Logiciel fini à 90%
<b>Bûcheur</b>	Planning + la conception Pas le temps pour les tests
<b>Planificateur</b>	Planning + planning Pas le temps pour le codage
<b>Optimal</b>	Il faut un juste équilibre !

## Un élément souvent oublié mais d'importance ; la documentation !

### Au minimum

- Définition des besoins
- Spécification logicielle
- Description des interfaces
- Sources
- Tests (procédures + résultats)
- Manuel utilisateur



La règle ne doit pas se substituer à l'intelligence. C'est-à-dire :  
**Volume de la documentation ~ Volume du projet**



# 1-1 Introduction – Les coûts

---

La qualité est un compromis ! (07)

Quelques chiffres que l'on rencontre et qui donne à réfléchir !

## Un taux d'échec :

**50%** des projets qui n'aboutissent jamais !

**25%** des projets qui n'apportent aucune valeur ajoutée aux utilisateurs

## Des retards :

**2/3** des projets dépassent largement les délais

**25% à 50%** représente le retard des grands projets

**50%** des cas, le planning est établi avant les spécifications

## Des défauts :

Représente **50%** des projets qui n'aboutissent jamais !

Sont à l'origine de **50%** des abandons de projets

## De l'incompréhension :

**65%** des projets ont des mésententes avec le client

**10%** des projets sont annulés suite à des attentes irréalistes

## Manque d'efficacité :

**40%** des erreurs sont générées par le stress

**65%** du temps à des activités contre-productives

# 1-1 Introduction – Le projet "Iceberg"

---

## Approche traditionnelle : Projet ICEBERG et ses étapes ....

- 1 Fixer la date d'achèvement des travaux
- 2 Baptiser le projet (optionnel)
- 3 Trouver un DINDON (en général le Chef Projet)
- 4 Trouver un analyste
- 5 Effectuer un ordonnancement des travaux
- 6 Décrire et commander les documents d'entrée
- 7 Embaucher les programmeurs
- 8 Bâtir les programmes de contrôle
- 9 Faire les programmes de conversion
- 10 Dessiner les fichiers

# 1-1 Introduction – Le projet "Iceberg"

---

## Approche traditionnelle : Projet ICEBERG et ses étapes .... (suite et fin ...)

11

Ecrire les programmes de mise à jour

12

Embaucher des programmeurs supplémentaires

13

Corriger le planning

14

Commencer les états de sortie

15

Lâcher prise

16

Paniquer, s'agiter, s'affoler ...

17

Chercher un coupable (et vite ....)

18

Punir un innocent

19

Promouvoir les sceptiques du ... début ...

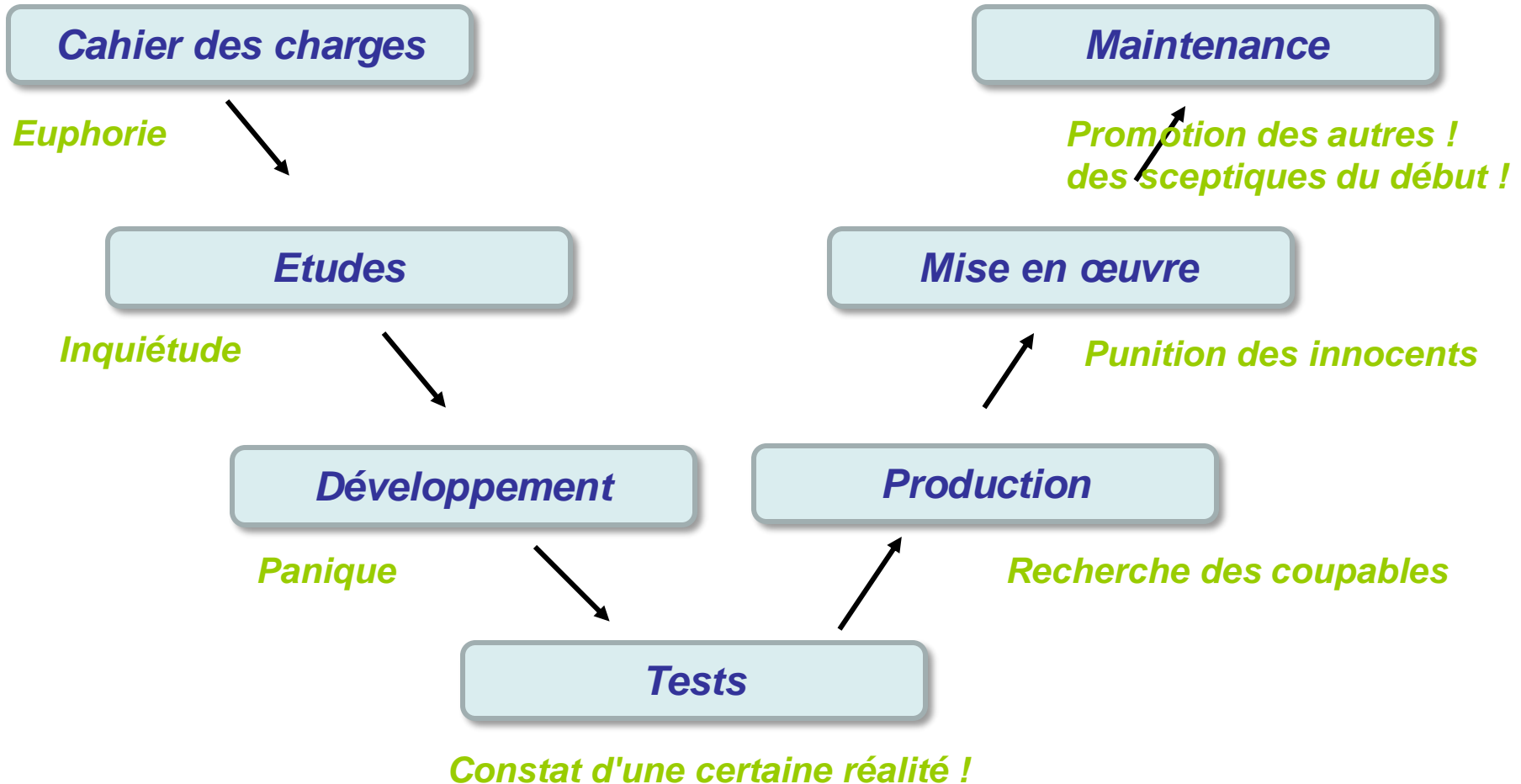
20

Revenir en position 1 et .... recommencer

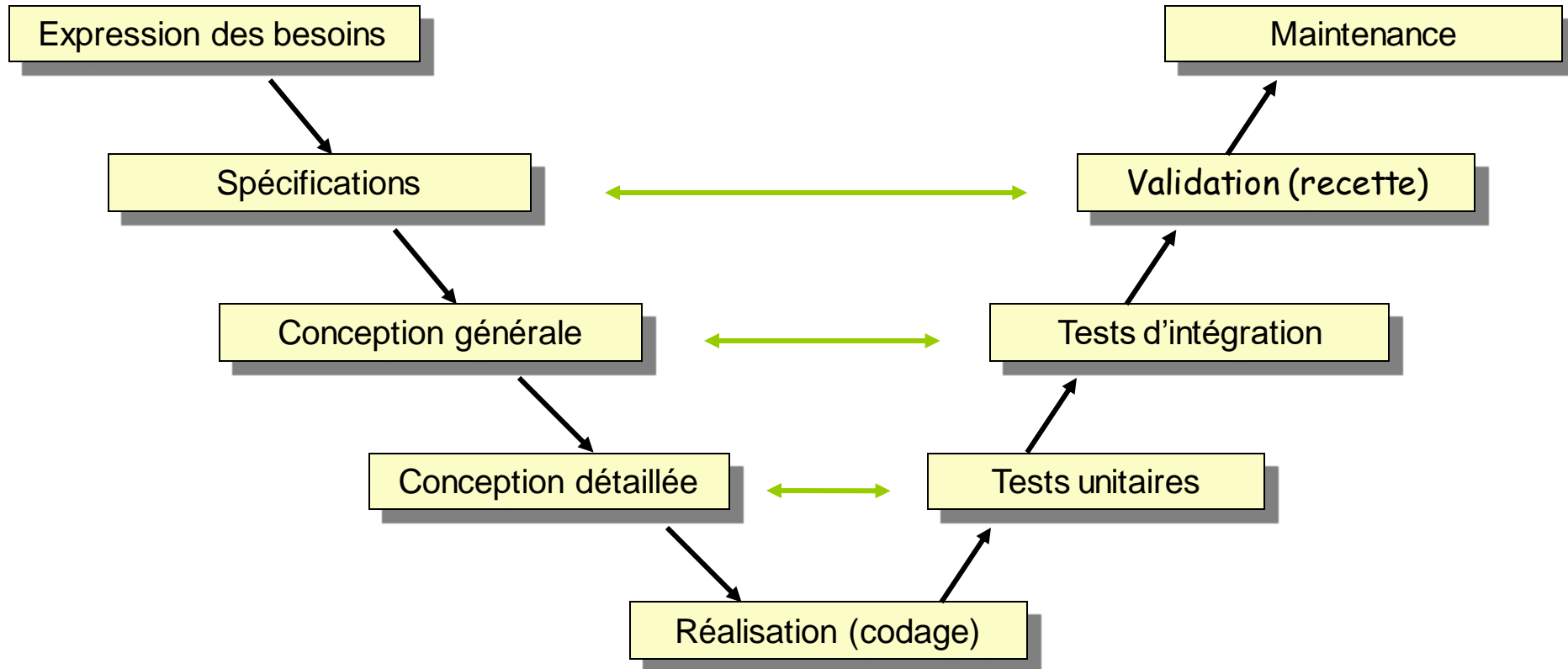
***Discussion !***

# 1-1 Introduction : suite et caricature !

Le cycle en V : une réalité ?



# Cycle en V



# 1-1 La mise en œuvre : Une vision !



**Comment le client l'a souhaité**



**Comment le chef projet l'a compris !**



**Comment l'analyste l'a schématisé**



**Comment le programmeur l'écrit**



**Comment le consultant l'a écrit**



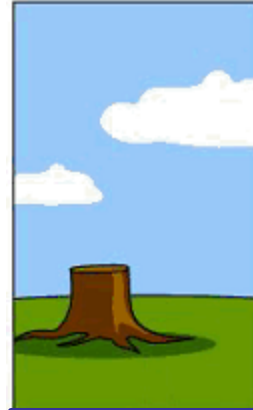
**Comment le projet a été documenté**



**Ce qui a été installé chez le client**



**Comment le client a été facturé**

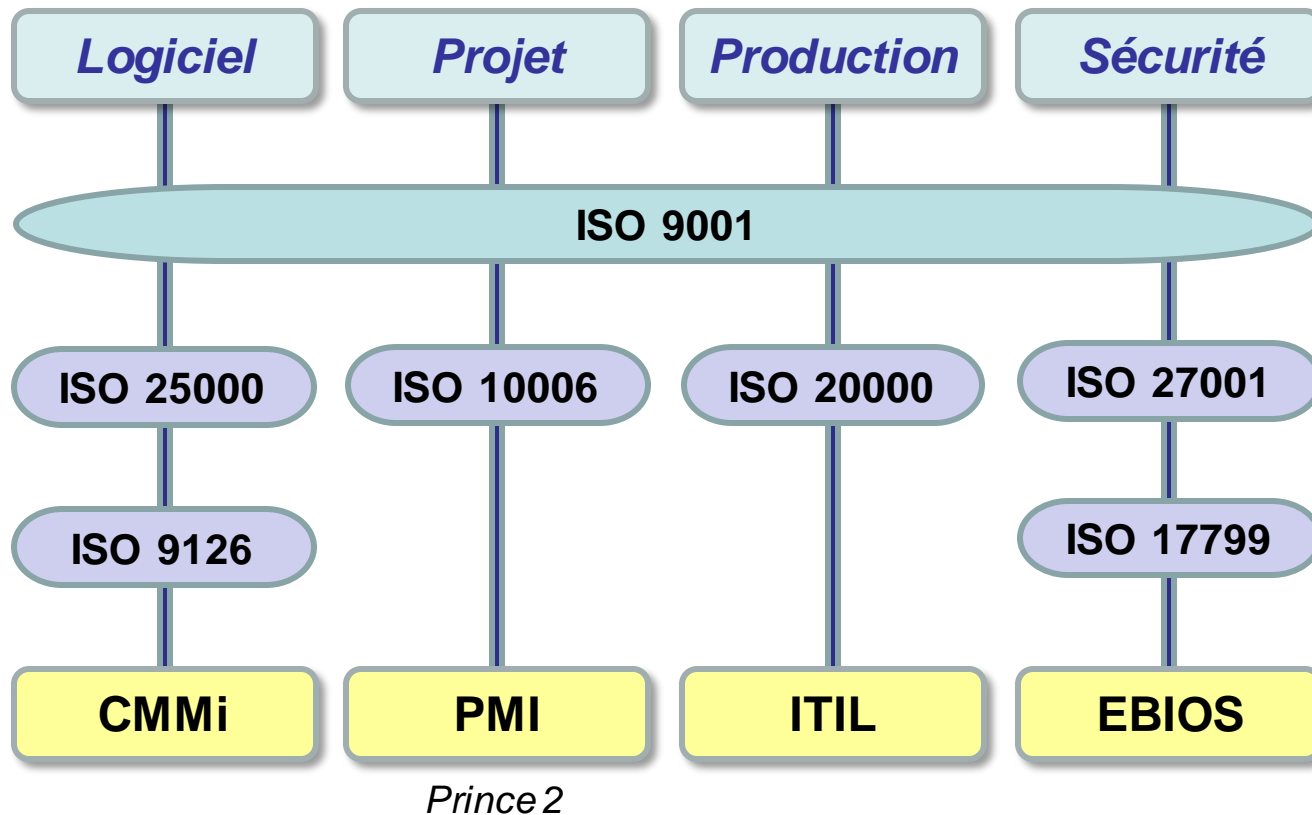


**Comment le support technique est réalisé**



**Ce que le client a réellement besoin**

# 1-1 Les principales normes ...



- L'ISO 9001, le référentiel le plus connu spécifie les exigences requises par un système de management de la qualité.
- ISO 9001 fait partie des normes ISO 9000 relatives à la qualité parmi lesquelles la norme ISO 9000 précise les principes essentiels et le vocabulaire.
- L'ISO 9004 énonce les lignes directrices pour l'amélioration des performances.
- Référentiel généraliste et flexible, cette norme est applicable à toutes les activités et à tous les métiers.



# 1-1 Les principales normes ...

---

- **CMMi** (Capability Maturity Model intégration) est le référentiel dédié à la conception et au développement de logiciels. La norme ISO 25000 a pour objectif de poser le cadre et la référence pour définir les exigences qualité d'un logiciel (ISO 9126).

La méthodologie de gestion de projet peut faire appel à **PMI** prônée par le Project Management Institute alors que l'ISO 10006 intervient sur le management de la qualité de projet

**PMI**, Project Management Institute association professionnelle internationale qui propose une méthodologie pour la gestion de projet et de programme dans le cadre du Project Management Body of Knowledge (connaissance) ou PMBOK disponible en Français.

- Pour un centre de production informatique, **ITIL** est un ensemble d'expériences et de bonnes pratiques structurées en pôles d'activités dont l'ISO 20000 (management de la qualité de service) peut être considérée comme le prolongement.

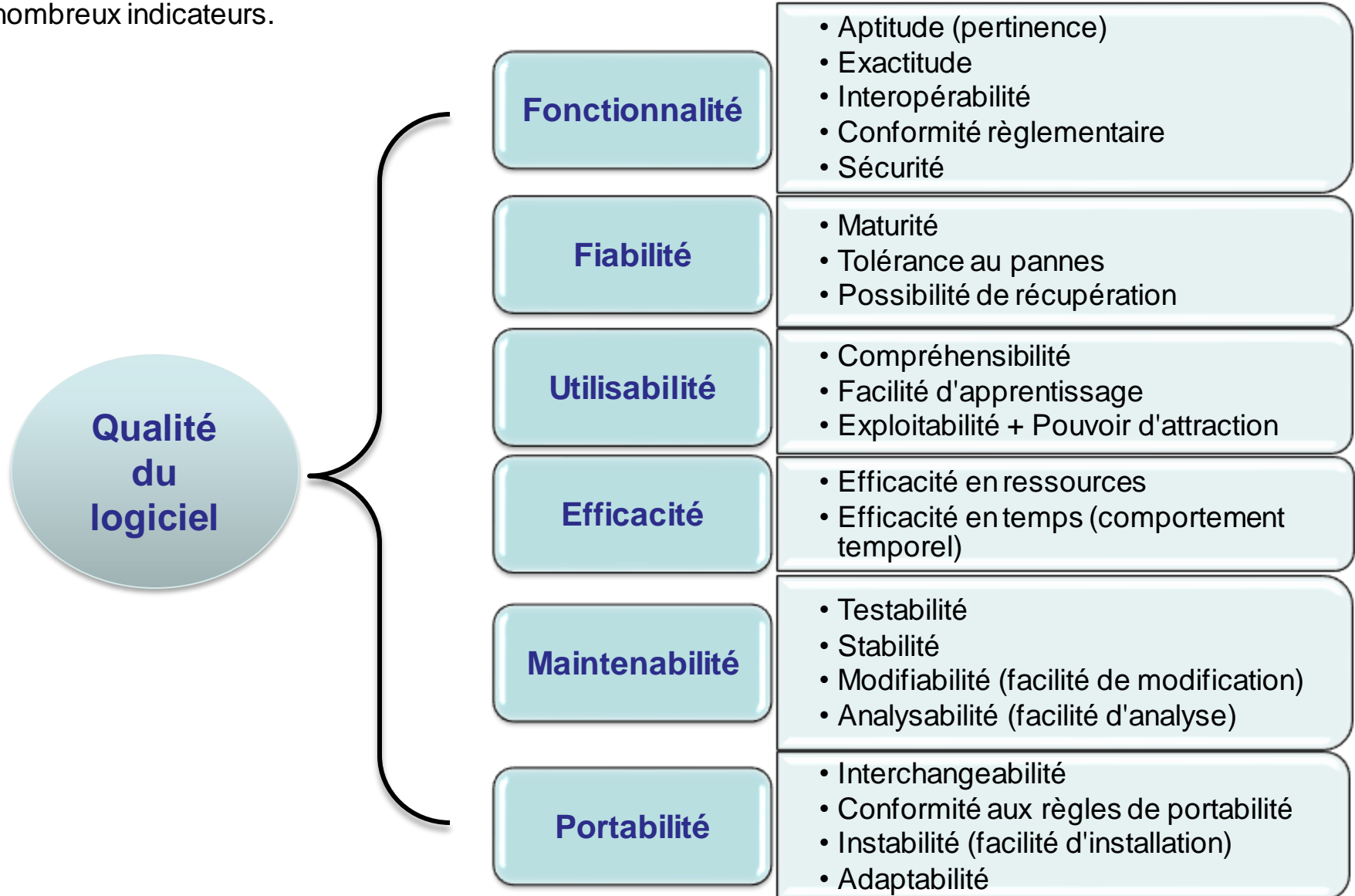
- **EBIOS** (Expression des Besoins et Identification des Objectifs de Sécurité) est une méthode établie par la DCSSI (Direction Centrale de la Sécurité des Systèmes d'Information) pour identifier les besoins de sécurité d'un système d'information.

L'ISO 17799 guide de bonnes pratiques est une annexe de l'ISO 27001 qui définit un système de gestion de la sécurité des systèmes d'information.

# 1-1 Qualité logicielle

La qualité d'un logiciel peut être définie (**norme ISO/CEI 9126**) par six caractéristiques principales, Chaque caractéristique se décompose en plusieurs sous-caractéristiques.

De ce fait, la **qualité logicielle** peut se définir comme une appréciation globale d'un logiciel, basée sur de nombreux indicateurs.



# 1-1 Qualité logicielle

---

## Capacité Fonctionnalité

***Est-ce que le logiciel répond aux besoins fonctionnels exprimés ?***

Représente un ensemble d'attributs portant sur l'existence d'un ensemble de fonctions et leurs propriétés données.

Les fonctions sont celles qui satisfont aux besoins exprimés ou implicites.

## Fiabilité

***Est-ce que le logiciel maintient son niveau de service dans des conditions précises et pendant une période déterminée ?***

Ensemble d'attributs portant sur l'aptitude du logiciel à maintenir son niveau de service dans des conditions précises et pendant une période déterminée.

## Utilisabilité Facilité d'usage

***Est-ce que le logiciel requiert peu d'effort à l'utilisation ?***

Ensemble d'attributs portant sur l'effort nécessaire pour l'utilisation et sur l'évaluation individuelle de cette utilisation par un ensemble défini ou implicite d'utilisateurs.

## Efficacité

***Est-ce que le logiciel requiert un dimensionnement rentable et proportionné de la plate-forme d'hébergement en regard des autres exigences ?***

Ensemble d'attributs portant sur le rapport existant entre le niveau de service d'un logiciel et la quantité de ressources utilisées, dans des conditions déterminées.

## Maintenabilité

***Est-ce que le logiciel requiert peu d'effort à son évolution par rapport aux nouveaux besoins ?***

Ensemble d'attributs portant sur l'effort nécessaire pour faire des modifications données.

## Portabilité

***Est-ce que le logiciel peut être transféré d'une plate-forme ou d'un environnement à un autre ?***

Ensemble d'attributs portant sur l'aptitude de logiciel à être transféré d'un environnement à l'autre.

# 1-1 Qualité logicielle

---

J.A. McCall, (au cours d'une étude pour l'US AIR FORCE) a identifié plus de cinquante facteurs candidats, permettant d'exprimer la qualité, puis ensuite il a sélectionné les onze facteurs principaux qui sont :

## **L'adaptabilité**

Mesure l'aptitude du logiciel à faciliter l'adjonction de nouvelles fonctionnalités ou la modification de fonctionnalités existantes (cet aspect couvre la correction des défauts provenant d'une mauvaise expression des besoins).

## **La confidentialité**

Mesure l'aptitude d'un logiciel à être protégé contre tout accès non autorisé.

## **La correction**

Mesure le degré de conformité par rapport aux spécifications.

## **La couplabilité**

Mesure l'aptitude d'un logiciel à être intégré dans un ensemble plus vaste.

## **L'efficacité**

Mesure l'aptitude d'un logiciel à minimiser la consommation des ressources qu'il utilise.

## **L'ergonomie**

Mesure l'aptitude d'un logiciel à être d'une utilisation agréable et facile pour l'utilisateur à qui il est destiné.

## **La maintenabilité**

Mesure l'aptitude d'un logiciel à faciliter la localisation et la correction d'erreurs résiduelles (il s'agit bien là de correction de défauts de non-conformité par rapport aux spécifications et non de défauts provenant d'une mauvaise expression des besoins).

# 1-1 Qualité logicielle

---

## **La portabilité**

Mesure l'aptitude du logiciel à minimiser les conséquences d'un changement d'environnement technique.

## **La réutilisabilité**

Mesure l'aptitude du logiciel à une réutilisation de tout ou partie de ses composants dans le cadre d'un autre projet.

## **La robustesse**

Mesure l'aptitude du logiciel à conserver un comportement conforme aux besoins dans le cas d'événements imprévus.

## **La testabilité**

Mesure l'aptitude d'un logiciel à faciliter la vérification de son comportement par rapport à des critères de test et de recette.

# 1-1 Qualité logicielle

---

## Exigences de la norme ISO 9000 :

La norme internationale ISO 9000 est une norme générique qui regroupe les normes ISO 9001, 9002 et 9003.

La différence entre les normes ISO 9001, ISO 9002 et ISO 9003 tient simplement au domaine d'application.

✚ **ISO 9001** établit les exigences relatives à une organisation dont les activités vont de la conception et du développement à la production, à l'installation et aux prestations associées.

✚ **ISO 9002** est la norme appropriée pour une organisation qui ne s'occupe pas de la conception et du développement; elle ne contient pas les exigences relatives à la maîtrise de la conception d'ISO 9001, les autres exigences étant identiques.

✚ **ISO 9003** est la norme qui correspond aux besoins d'une organisation dont les processus d'activité ne portent pas sur la maîtrise de la conception, la maîtrise des processus, les achats ou les prestations associées, et qui a fondamentalement recours aux contrôles et aux essais pour assurer que les produits et les services finals satisfont aux exigences spécifiées.

*C'est donc à l'organisation de choisir ISO 9001, ISO 9002 ou ISO 9003 pour faire certifier son système qualité, en fonction des processus d'activité couverts par son système qualité. Il n'y a pas de différence de nature hiérarchique entre les trois normes.*

# 1-1 Qualité logicielle – Récapitulatif ISO 9000 / chapitres

ISO 9001	ISO 9002	ISO 9003	Chapitres de l'ISO 9000
X	X	X	Responsabilité de la direction
X	X	X	Système Qualité
X	X		Revue de contrat
X			Maîtrise de la conception
X	X	X	Maîtrise des documents
X	X		Achats
X	X	X	Maîtrise du produit fourni par le client
X	X		Identification et traçabilité
X	X		Maîtrise du processus
X	X		Contrôle et essais
X	X	X	Maîtrise des équipements de contrôle, de mesure et d'essai
X	X	X	Etat des contrôles et essais
X	X		Maîtrise du produit non-conforme
X	X		Actions correctives et préventives
X	X	X	Manutention, stockage, ...
X	X		Enregistrements relatifs à la qualité
X	X		Audits qualité internes
X	X		Formation
X	X		Prestations associées
X	X		Techniques statistiques



# 1-1 Qualité logicielle – Récapitulatif normes !

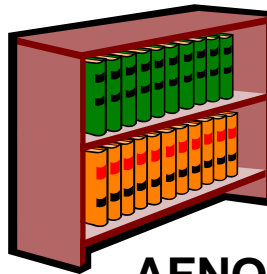
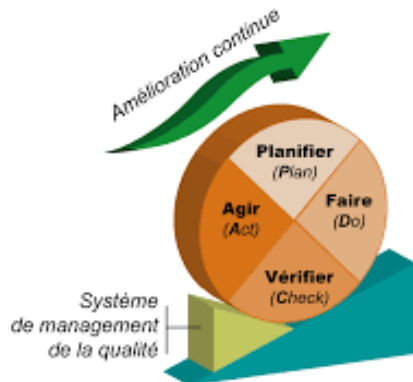


## Référentiel des normes et d'autres ....



### ISO :

- ISO 9000 : référentiel international du système qualité 5 normes
- Normes relatives aux SI :
  - ISO 8402 (Assurance Qualité)
  - ISO 10011 (Audit Qualité)
  - ISO 12207 (Cycle de vie)
  - ISO/CEI 9126 (Evaluation)
  - ISO/CEI 12119 (Progiciels)



### AFNOR :

X50-120 (Vocabulaire)  
Z67-100, 101, 111, 130, 131, 135, 150, 151,  
900, 901, 902



# 1-1 Qualité logicielle – Quelques concepts autour de la qualité

## La qualité externe :

- Correspond à la satisfaction du client.
- Fournir un produit ou des services conformes à l'attente du client.

Cela passe par la nécessité d'écoute du client en prenant en compte les besoins exprimés et "implicite !".

## La qualité interne :

- Correspond à l'amélioration du fonctionnement interne de l'entreprise.
- Mettre en œuvre des moyens permettant de décrire au mieux l'organisation, de repérer et de limiter les dysfonctionnements.

La qualité interne passe par une étape d'identification et de formalisation des processus internes réalisés grâce à une démarche participative.

## La non-qualité et la sur-qualité :

- L'opposé de la qualité, nommée **non-qualité**, possède également un coût.
- Il s'avère plus coûteux de corriger les défauts, les erreurs que de "faire bien" dès le départ.

Le coût de la non qualité est d'autant plus important qu'elle est détectée tardivement.

La différence de prix sera moins grande si le défaut est détecté en cours de production que s'il est détecté par le client final (insatisfaction du client, traitement de l'incident, suivi du client, frais divers, etc.). Trouver le juste équilibre pour éliminer au maximum la **non qualité**.

L'objet de la qualité est de fournir une offre adaptée au Client, avec des processus maîtrisés tout en s'assurant que l'amélioration ne se traduit pas par un surcoût général.

Dans ce cas, on parle de **"sur-qualité"**. Il est possible d'améliorer un grand nombre de dysfonctionnements à moindre coût, mais, à l'inverse, plus on souhaite approcher la perfection plus les coûts grimpent !

# 1-1 Qualité logicielle – Quelques concepts autour de la qualité

Il existe des pertes quelquefois considérables dans l'entreprise qui sont engendrées par une mauvaise qualité !

De ce fait, il existe des coûts "cachés ou pas" de cette non qualité.

Il y a donc un gisement à exploiter pour accroître au sein de l'entreprise de la compétitivité, il faut donc établir des priorités pour réaliser des actions correctives et mesurer les progrès.

Ces coûts résultants de la non qualité se décomposent en :

## **Coût de détection :**

Ce sont essentiellement les dépenses engagées pour vérifier la conformité des produits aux exigences de qualité.



## **Coût de prévention :**

Ce sont tous les investissements humains et matériels engagés pour vérifier, prévenir et réduire les anomalies.



## **Coût des anomalies internes :**

Représente les frais encourus lorsque le produit ne satisfait pas aux exigences avant d'avoir quitté l'entreprise.



## **Coût des anomalies externes :**

Représente les frais encourus lorsque le produit ne répond pas aux exigences de qualité. Comme les pénalités !



***Économies et satisfaction du client !***

***Humour ou qualité ou économie !!***



# 1-1 Qualité logicielle

---

*On peut commenter !*

## **Exemple 1 :**

- On compare deux logiciels :
  - Le premier a encore 10 erreurs résiduelles.
  - Le second n'a que 3 erreurs résiduelles.
- Est-ce que le second est de meilleure qualité que le premier ?

## **Exemple 2 :**

- On compare deux logiciels :
  - Le premier marche très bien, mais on ne dispose plus ni des sources, ni de son dossier de développement.
  - Le second marche à peu près, a encore des erreurs, mais on dispose de ses sources et de son dossier complet de développement.
- Lequel est de meilleure qualité ?

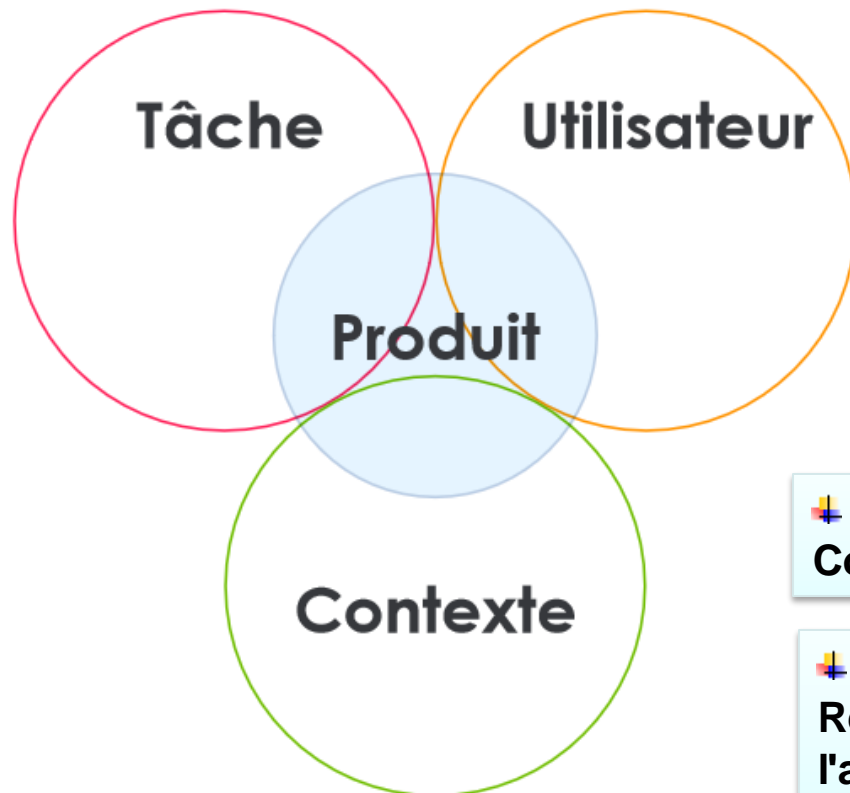
# 1-1 Qualité logicielle - Ergonomie

## L'ergonomie du logiciel :

C'est souvent un domaine que l'on "laisse de côté"; "on verra plus tard", "c'est inutile", "suis pas compétant", "ça ira" ...

L'ergonomie représente l'étude des situations de travail et des relations entre l'être humain et un système.

De plus, est difficile de mesurer exactement le gain apporté lors de la conception ergonomique. Le but principal est d'améliorer le confort, accroître sa productivité, voire le plaisir, de l'utilisateur ... Cela amène l'efficacité dans son travail et dans l'utilisation du logiciel.



✚ **Ergonomie :**  
Ayant pour but d'adapter (utilité + utilisabilité) un produit (ou un service) à :

- Une Tâche,
- Un Utilisateur,
- Un contexte précis.

✚ **Utilisateur :**  
Personne qui va utiliser l'application.

✚ **Tâche :**  
Comment un Utilisateur interagit avec l'application.

✚ **Contexte :**  
Représente toutes les conditions d'utilisation de l'application..

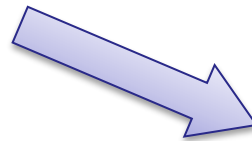
# 1-1 Qualité logicielle – Ergonomie (les phases)

## La conception des IHM :

- Importance des maquettes, des prototypes.  
Ces deux points servent pour évaluer et de support de communication entre tous les acteurs.
- Importance des outils à mettre en œuvre. permettre

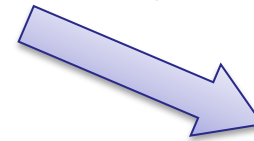
**De la maquette**

**A jeter !**



**Au prototype**

**Réutilisable !**



*Il n'y a pas obligatoirement  
passage ou évolution continue du  
prototype vers le Produit final.*

**Vers le Produit  
final**

**Opérationnel**

### Maquette :

- Ensemble d'objets graphiques donnant une image de l'écran/utilisateur mais sans les fonctionnalités.
- C'est un support de communication entre les concepteurs.

### Prototype :

- Pour une évaluation de fonctionnement.
- Il faut nécessairement de la rapidité lors du développement.
- Cela permet une spécification précise et définitive.

### Produit final :

- Toujours en production avec son environnement.



# 1-1 Qualité logicielle – Les normes (principales)

Il existe une multitude de normes sur l'ergonomie que ce soit sur la qualité logicielle ou dans le milieu industrielle.

Il est conseillé de lire les documents spécialisés dans ce domaine.

<b><i>N° ISO</i></b>	<b><i>Chapitres des "ISO"</i></b>
<b>ISO 13407</b>	Processus de conception centrée sur l'opérateur humain pour les systèmes interactifs
<b>ISO/TR 16982</b>	Méthodes d'utilisabilité pour la conception centrée sur l'opérateur humain
<b>ISO 9241-10</b>	Principes de dialogue
<b>ISO 9241-11</b>	Lignes directrices concernant l'utilisabilité
<b>ISO 9241-12</b>	Présentation de l'information
<b>ISO 9241-13</b>	Guidage de l'utilisateur
<b>ISO 9241-14</b>	Dialogues de type menu
<b>ISO 9241-15</b>	Dialogues de type langage de commande
<b>ISO 9241-16</b>	Dialogues de type manipulation directe
<b>ISO 9241-17</b>	Dialogues de type remplissage de formulaires
<b>ISO 14915</b>	Conception d'interfaces utilisateur multimédia
<b><i>Autres normes</i></b>	<b><i>Libellé</i></b>
<b>IUSR &amp; CIF</b>	Faciliter les échanges entre consultants mais aussi entre consultants et universitaire (en cours de normalisation)
<b>ISO/TS 16071</b>	L'accessibilité des logiciels
<b>AFNOR Z-67133-1</b>	Définit l'ergonomie du logiciel par sept critères

# 1-1 Gérer la Qualité – Qualité technique

---

## Un postulat, un principe, une évidence ... :

- **La qualité est la résultante d'actions préventives et non pas d'actions correctives.**
- Dans un projet informatique, on définit 3 axes principaux de mesure de la qualité :
  - Qualité technique,
  - Qualité relationnelle,
  - Qualité organisationnelle.

## Qualité technique :

- Il est obligatoire de composer et de constituer l'équipe(s) en fonction des compétences techniques et/ou fonctionnelles requises par le projet (à discuter !).
- Eviter les rôles de composition, le copinage, les relations opportunistes, les formations sur le tas...

## De plus :

La qualité technique s'appuie essentiellement et obligatoirement sur :

- Les normes et méthodes de développement (conception, architecture, règles de développement, charte de présentation, règle de l'art, ...).
- Les moyens de contrôle (niveau de validation, revue de code, conception de jeux et plans de tests, documentation ad hoc, ...).

# 1-1 Gérer la Qualité – Qualité relationnelle

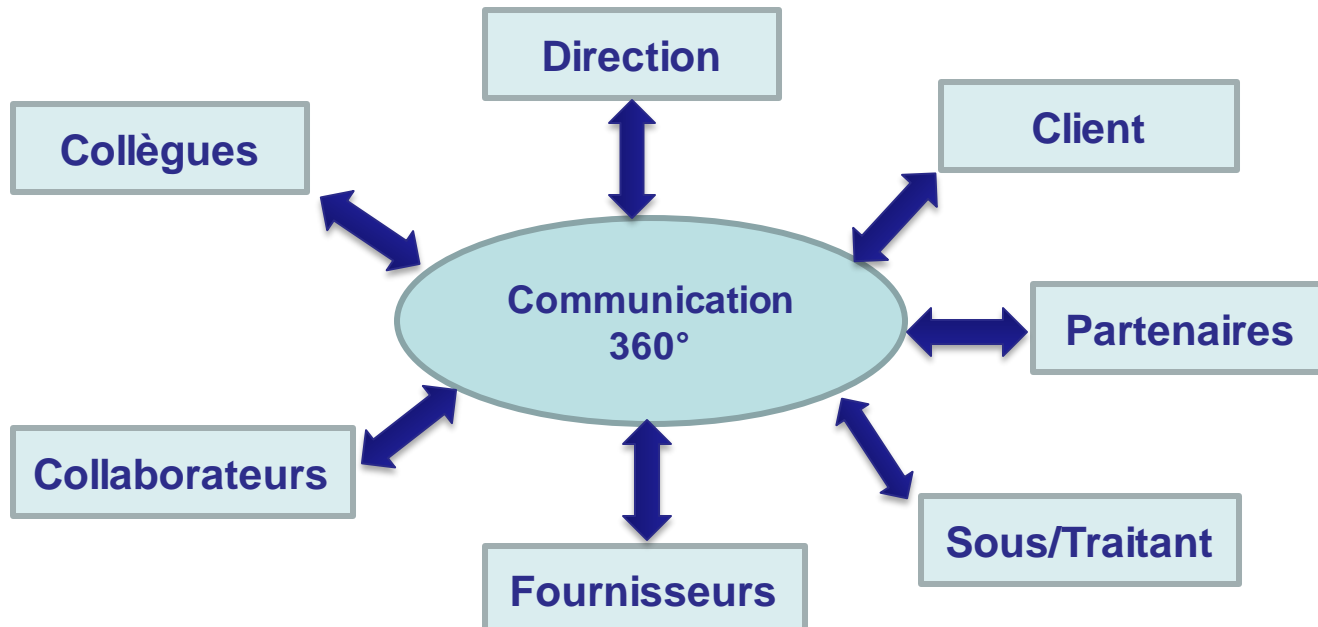
## Qualité relationnelle :

- Le Chef Projet est l'élément fondamental du projet. C'est lui qui doit principalement communiquer autour et dans le projet.
- Sa communication doit être "au top" pour qu'il puisse faire vivre le projet
- Il doit aussi faire la promotion du projet, de son équipe, de ses atouts, ...
- Chacun doit *"savoir ce que l'autre fait !"* (à discuter !).

## De plus :

Le Chef Projet doit assurer la communication :

- Au sein de son équipe,
- Auprès du client, des fournisseurs et des différents acteurs prenant part au projet.
- À l'extérieur du projet :
- Doit assurer l'arbitrage !



# 1-1 Gérer la Qualité – Qualité organisationnelle

## Qualité organisationnelle :

- Cette qualité doit s'appuyer par la mise en place de divers documents comme par exemple le **PAQ** : **P**lan d'**A**ssurance **Q**ualité.
- Ce PAQ permet de définir et de cadrer le projet dans une organisation et un environnement défini par : les règles, les standards, les normes, les méthodes, les outils, les diverses responsabilités, etc.
- Le Plan Assurance Qualité (PAQ) définit toutes les dispositions spécifiques prises par l'équipe projet et le client afin de garantir la conformité des produits livrés avec les exigences spécifiées dans le cadre de la réalisation du projet
- Ce document doit permettre le bon déroulement du projet et d'assurer sa réussite.

## De plus :

La réussite du projet peut s'apprécier :

- Par la qualité des résultats obtenus,
- Par la qualité des résultats attendus,
- Par le respect des coûts et délais.

Le rôle du Plan Assurance Qualité (**PAQ**) consiste à décrire :

- Ce que l'on veut faire (les objectifs),
- Le périmètre et les limites de la prestation (important !),
- Comment le faire (les outils, les méthodes, ...),
- Quand le faire (délais),
- Qui en a la responsabilité (mentionner personnes, les interlocuteurs),
- Qui va le faire (indiquer Personne/Equipes),
- Les moyens humains et matériels à mettre en œuvre,
- Comment mesurer les résultats (indicateurs, tableaux de bord).

**Exemple  
PAQ**

# 1-1 Gérer la Qualité – Qualité organisationnelle

## Il existe principalement 3 niveaux de "Plan Qualité" :

### *Le Manuel d'Assurance Qualité (**MAQ** : niveau le plus haut)*

C'est un document énonçant la politique qualité et décrivant l'ensemble des procédures et autres composants organisationnels du système qualité d'un organisme. C'est la description des principes de qualité généraux au sein d'une société.

### *Le Plan d'Assurance Qualité (**PAQ** : niveau intermédiaire)*

"Le **PAQ** est un document qui précise les éléments permettant de s'assurer de la mise en œuvre et de l'efficacité des activités prévues pour obtenir la qualité requise." (AFNOR/Z67-100-3).

On dit aussi que le **PAQ** régit les principes de qualité entre un client et un fournisseur.

### *Le Plan Qualité Projet (**PQP** : niveau le plus bas)*

Il permet de régir les principes de qualité relative aux spécificités d'un seul projet. Il a pour but de présenter les dispositions prises par la MOA et la MOE pour :

- Organiser et planifier le projet,
- Et en assurer la qualité.
- Il constitue un outil de travail et un référentiel commun à tous les acteurs.

## Il faut une conclusion :

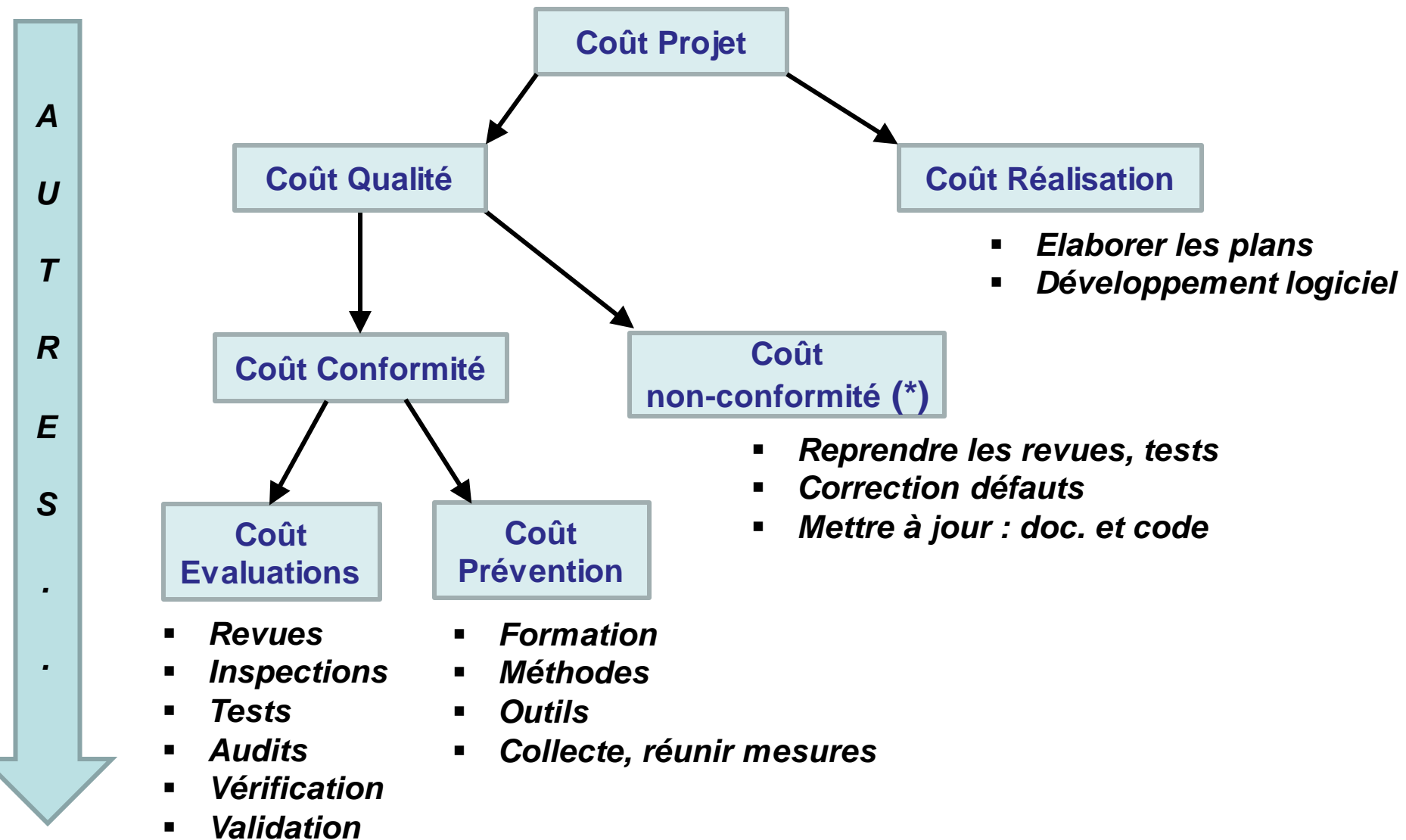
Le PAQ est un document, un outil utilisé dans tous les domaines et en particulier dans le domaine du service informatique. Certes, il facilite le travail, les rapports entre tous les acteurs.

Néanmoins, son existence n'assure pas forcément la réussite du projet. **Ce n'est pas un parapluie.**

Un projet est un "amalgame de petites choses" ; bonne entente, compétence, bon outil, bonne méthode, honnêteté, sérieux, travail en qualité etc.

Ce cumul engendrera certainement la réussite du projet.

# 1-1 Gérer la Qualité – Coût d'un projet

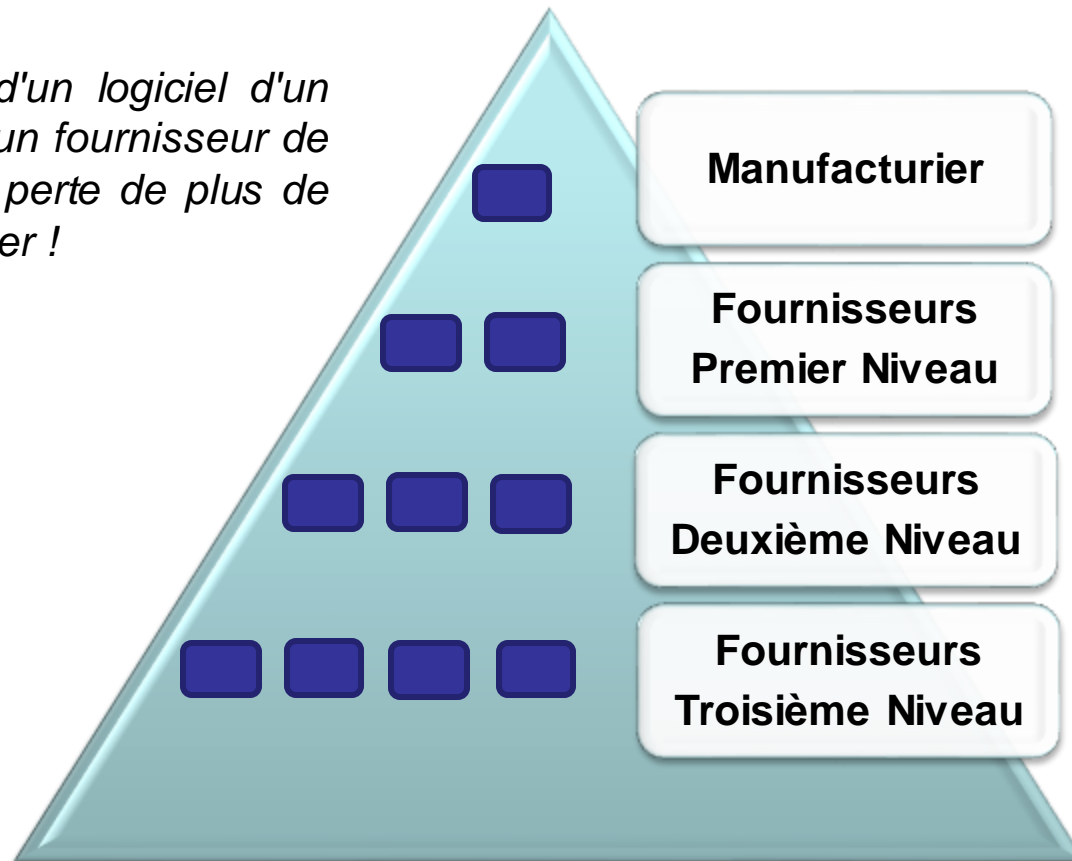


(\*) *C'est une fonction (une étape) qui demande nécessairement sa reprise totale.*

# 1-1 Gérer la Qualité – Coût d'un projet

*Dans un projet, nous ne sommes pas exclusivement une seule entité !*

Exemple : Un défaut d'un logiciel d'un composant produit par un fournisseur de Niveau 3 a causé une perte de plus de 200 M \$ au manufacturier !



Nota : La mise en œuvre d'une démarche qualité sur un projet n'est pas, en soi, une garantie de la tenue des objectifs du projet ...



# 1-1 Gérer la Qualité – Les tests

## **Définitions de l'IEEE :**

*"Le test est l'exécution ou l'évaluation d'un système ou d'un composant, par des moyens automatiques ou manuels, pour vérifier qu'il répond à ses spécifications ou identifier les différences entre les résultats attendus et les résultats obtenus."*

*"Le test est une activité destinée à déterminer si l'évaluation d'une caractéristique ou d'une aptitude d'un programme ou d'un système donne les résultats requis."*

**Selon Bill Hetzel**

## **Qu'est ce qu'un test réussi ?**

Un test réussi n'est pas un test qui n'a pas trouvé de défauts, mais un test qui a effectivement trouvé un défaut (ou une anomalie).

**G.J. Myers**

## **Le test des logiciels :** Edition Hermes

*"Le test d'un logiciel est une activité qui fait partie du processus de développement. Il est mené selon les règles de l'assurance de la qualité et débute une fois que l'activité de programmation est terminée. Il s'intéresse aussi bien au code source qu'au comportement du logiciel. Son objectif consiste à minimiser les chances d'apparition d'une anomalie avec des moyens automatiques ou manuels qui visent à détecter aussi bien les diverses anomalies possibles que les éventuels défauts qui les provoqueraient."*

## **Le meilleur ! :**

Le test d'un programme peut être utilisé pour montrer la présence de Bug mais jamais pour montrer leur absence.

**Edsger Wybe Dijkstra**

# 1-1 Gérer la Qualité – Les tests

---

Le test est souvent associé avec 2 termes qui sont Vérification et Validation.

## Différence entre : Vérification et Validation (V&V)

### Vérification :

Confirmation par examen et apport de preuves tangibles que les exigences spécifiées ont été satisfaites.

*"Avons-nous construit le produit bien ?"*  
*Le logiciel doit être conforme à sa spécification.*  
*Faisons-nous le travail correctement ?*

Consiste à s'assurer que les choses sont réalisées conformément à ce qui avait été défini.

### Validation :

Confirmation par des preuves tangibles que les exigences pour une utilisation spécifique ou une application prévue ont été satisfaites (ISO 9000).

*"Avons-nous construit le bon produit ?"*  
*Le logiciel doit faire ce que l'utilisateur a besoin.*  
*Faisons-nous le travail attendu ?*

Consiste à s'assurer que les résultats sont bien atteints.

Il existe une chronologie à respecter :

- **La vérification** est réalisée avant la **validation**.
- Attention : Tout changement peut induire la nécessité d'effectuer de nouvelles vérifications et validations.

# 1-1 Gérer la Qualité – Les tests

## Définition de classes des erreurs :

- Calcul
- Logique
- Entrée / sortie
- Traitement des données
- Interface
- Définition des données



## Paramètres rendant les tests difficiles :

- Tests Exhaustif
- Processus d'introduction des défauts
- Difficultés d'ordre psychologique ou culturel
- Difficultés formelles
- Autres ...

### Calcul

On fait :  $X \leftarrow X + 2$  au lieu de  $X \leftarrow X - 2$

### Logique

On fait : SI  $A > B$  Alors au lieu de SI  $A < B$  Alors ....

### Entrée/Sortie

Mauvaise description des données, mauvais passage d'arguments, mauvaise conversion, erreur dans le formatage, ...

### Traitement de données

Mauvais accès des fichiers, variables non définie, dépassement de tableau ...

### Interface

On appelle sur les variables : P1 au lieu de P2, mauvais passage de paramètres, ...

### Définition des données

Définition de Car au lieu de Int ou Long etc.

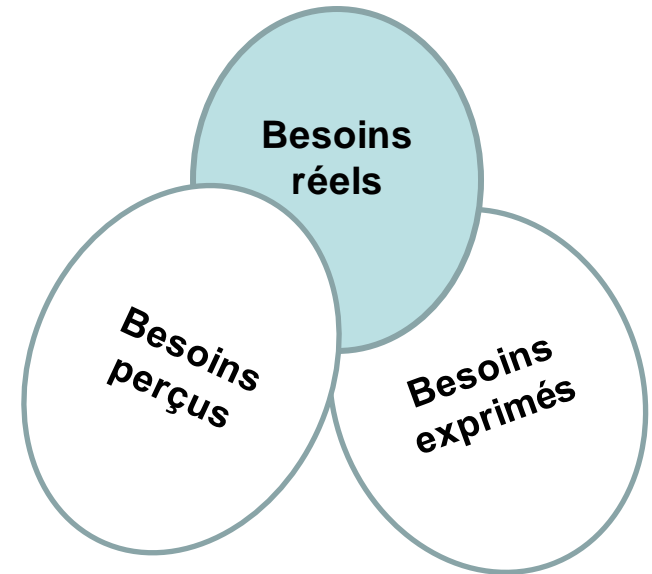
# 1-1 Gérer la Qualité – Les tests

---

- Le test "exhaustif" est très difficile à réaliser.
- Le test est une méthode de vérification partielle de logiciels.
- La qualité du test dépend de la pertinence du choix des données de test.

## Processus d'introduction des défauts par :

- Manipulation de données abstraites
- Succession de transformations :
  - Idée "abstraite" et confuse du client
  - Spécifications
  - Conception
  - Code source
  - Assemblage, ....
- Perte d'informations, introduction d'erreurs, etc.



## Difficultés d'ordre psychologique ou culturel :

- Activité tendant à montrer les erreurs des programmeurs !
- Manque d'intérêt pour ce domaine du génie logiciel !

## Autres difficultés : (ce ne sont pas des excuses ...!)

- Taille et complexité des programmes,
- Perturbation / modification du comportement normal du code par l'outillage de test,
- Distinction environnement de développement et environnement de production,
- Le temps réel, Système et machine,
- L'environnement extérieur.

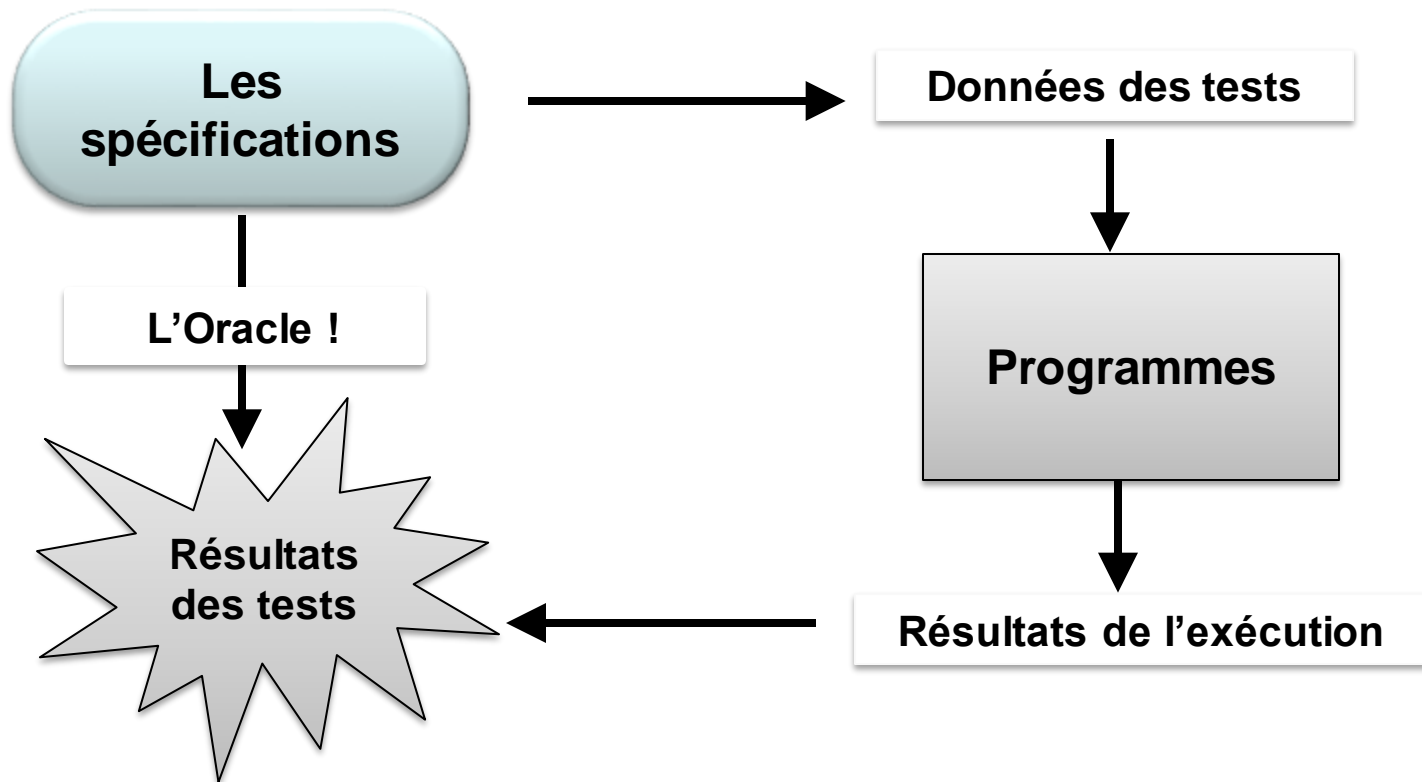
# 1-1 Gérer la Qualité – Les tests

## Technique : Par la Boîte noire

Ne pas oublier :

- Le test est une méthode de vérification partielle.
- La qualité du test dépend de la pertinence du choix des données de test.

### *Tests de conformité par rapport aux spécifications définies*



Ce sont des tests permettant de vérifier l'adéquation du logiciel aux contraintes définies dans les spécifications. Les tests de fonctionnalités évaluent la réaction du logiciel par rapport à des données d'entrées représentatives. *On teste ici ce que le programme est censé faire et non pas ce qu'il fait réellement.*

# 1-1 Gérer la Qualité – Les tests

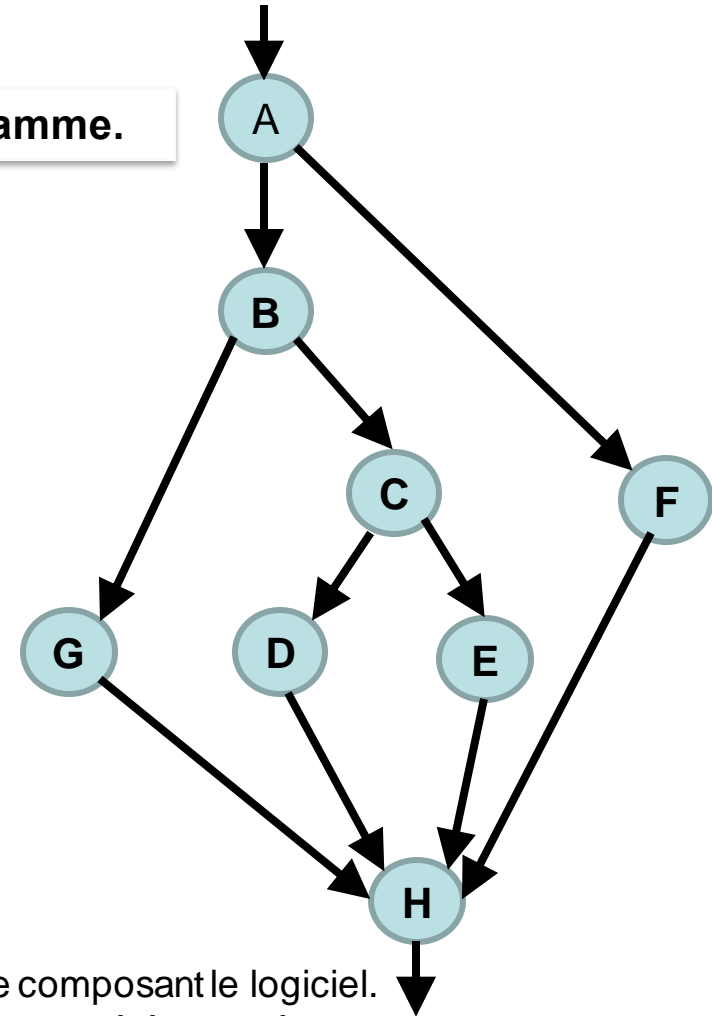
## Le test dit « Structurel » : Par la boîte Blanche

- Toutes les données des tests sont forcément réalisées et produites à partir de l'analyse du code source.

Flot des contrôles d'un programme.

**Quels sont les critères des tests ? :**

- Tous les chemins,
- Toutes les branches,
- Toutes les instructions,
- La non régression !
- Comparaisons, Bilan.



Ce sont des tests permettant de valider la structure de chaque module composant le logiciel.

Ils permettent d'avoir l'assurance que toutes les parties d'un programme ont été essayées.

*On teste ici ce que le programme fait et non pas ce qu'il est censé faire.*

Tests fonctionnels et tests structurels sont complémentaires, dans le sens où ensemble, ils permettent de tester ce que le programme doit faire et ce que le programme fait.

# 1-1 Gérer la Qualité – Les tests : Les outils

---

**Le portail du test :** <http://portaildutest.qualifiez.fr/>

## **Outil Automatisation :**

- Selenium
- AutoIT
- Canoo WebTest
- Watir
- CubicTest

## **Outil Gestion des tests :**

- Outils borland
- Salome
- TestLink
- Fitnessse
- Outil HP
- Twist

## **Outil test de Charge :**

- Grinder
- OpenSta
- JMeter