INGENIEURIE LOGICIELLE

Steeve ASSOUS ©

pour AJC FORMATION

Notion de Qualité : ISO 9000

- **« Aptitude d'un ensemble de caractéristiques d'un produit, d'un système ou d'un processus à satisfaire les exigences d'un client et des autres parties intéressées. »**
-
- **Au sein d'une entreprise ou d'un projet, la qualité doit être entre les mains d'une équipe dédiée.**
- **Ses tâches principales sont :**
  - La planification des actions qualité à mener.
  - L'exécution de ces actions.
  - L'audit du résultat.
  - L'anticipation de nouvelles actions pour améliorer la qualité de l'existant.
- **Au final, une politique qualité cohérente permet :**
  - De s'assurer que le produit à développer est conforme aux exigences.
  - De vérifier que le produit est conforme à la politique et aux contraintes du client.
  - De mesurer les résultats obtenus.

Notion de Qualité : ISO 9000

- **« Aptitude d'un ensemble de caractéristiques d'un produit, d'un système ou d'un processus à satisfaire les exigences d'un client et des autres parties intéressées. »**
-
-
-
-

Une formule bien connue résume bien le management qualité :

« Ecrire ce que l'on va faire
Faire ce que l'on a écrit
Vérifier que ce que l'on a fait est bien ce que l'on a écrit.
Chercher en permanence à s'améliorer »

-
  - De vérifier que le produit est conforme à la politique et aux contraintes du client.
  - De mesurer les résultats obtenus.

TESTS ET VALIDATION DU LOGICIEL

Préambule

- Les tests existent depuis toujours (mauvaise perception).
- Prise de conscience actuelle de la part des sociétés et notamment des directions informatiques (risque de dysfonctionnement, perte financière, retard…).
- Les tests rassurent et permettent de palier aux erreurs humaines.
- Le sujet des tests est vaste.
- Nous avons tenter dans cette présentation de vous en présenter les principaux aspects (généralités, techniques et outils).

« La politique de tests est aujourd'hui une dimension incontournable de la gestion de projet. »

Quelques exemples de « Bugs »

- 1992 - Les ambulances de Londres sont mal orientées par le logiciel. Des pertes humaines sont à déplorer.
- 1996 - Explosion de la fusée Ariane 5 au bout de 30 secondes de vol suite à une erreur de conversion de données numériques.
- 2004 - Défaillance du système d'alarme d'une centrale qui produisit une coupure d'électricité aux Etats-Unis et au Canada.
- 2006 - Deux grandes banques françaises exécutent un double débit pour plus de 400 000 transactions.
- 2009 : Panne de pilote automatique pour un appareil d'Airbus
- 2010 - Plusieurs millions d'utilisateurs de cartes bancaires allemands pris en otage
- Que dire de …
  - L'avionique civil
  - Les centrales nucléaires
  - Les systèmes de gestion de places de marchés
  - Un système de vente en ligne

Quelques exemples de « Bugs »

- 1992 - Les ambulances de Londres sont mal orientées par le logiciel. Des pertes humaines sont à déplorer.
- 1996 - Explosion de la fusée Ariane 5 au bout de 30 secondes de vol suite à une erreur de conversion de données numériques.
- 2004 - Défaillance du système d'alarme d'une centrale qui produisit une

Les bugs sont omniprésents et ont parfois des conséquences dramatiques.

- 2006 - Deux grandes banques françaises exécutent un double débit pour plus de 400 000 transactions.
- 2009 : Panne de pilote automatique pour un appareil d'Airbus
- 2010 - Plusieurs millions d'utilisateurs de cartes bancaires allemands pris en otage
- Que dire de …
  - L'avionique civil
  - Les centrales nucléaires
  - Les systèmes de gestion de places de marchés
  - Un système de vente en ligne

Définitions des tests

- **Selon Glendford.J Myers dans « The art of software testing » :**
  - « Un test réussi n'est pas un test qui n'a pas trouvé de défauts, mais un test qui a effectivement trouvé un défaut. »
- **Selon l'IEEE (Institute of Electrical and Electronics Engineers) :**
  - « 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. »
- **Selon l'AFNOR :**
  - « Phase du projet dans laquelle le client et le fournisseur testent la correspondance entre ce qui a été commandé et ce qui est effectivement produit. »
- **Selon Bill Hetzel :**
  - « 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. »

Définitions

« On constate une **ANOMALIE**
due à un **DEFAUT** du logiciel lui même
du a une **ERREUR** de programmeur. »

- *Erreur* (IEEE 729) : écart entre une valeur ou condition calculée, observée ou mesurée et la valeur ou condition qui est vraie, spécifiée ou théoriquement correcte.
- *Défaut*, anomalie (fault, bug) (IEEE 729) est la manifestation d'une erreur dans un logiciel. Un défaut peut causer une panne.
- *Panne* (failure) (IEEE 729) est la fin de la capacité d'un système ou d'un de ses composants d'effectuer la fonction requise, ou de l'effectuer à l'intérieur de limites spécifiées.

Les classes de défaut

- **Les erreurs peuvent être de divers ordres.**
  - Il peut s'agir d'erreurs :
    - de calcul
    - de logique
    - d'entrée / sortie
    - de traitement des données (dépassement de tableau)
    - d'interface (communication entre les programmes) de définition des données

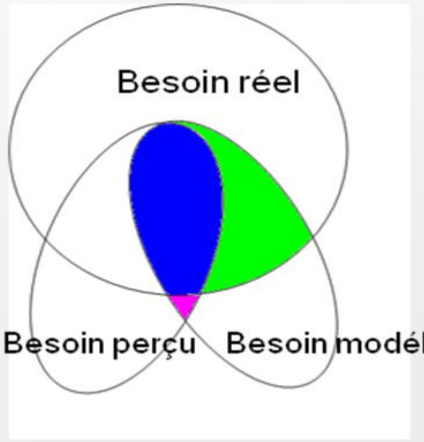
Ces erreurs représentent 90% des cas décelés.

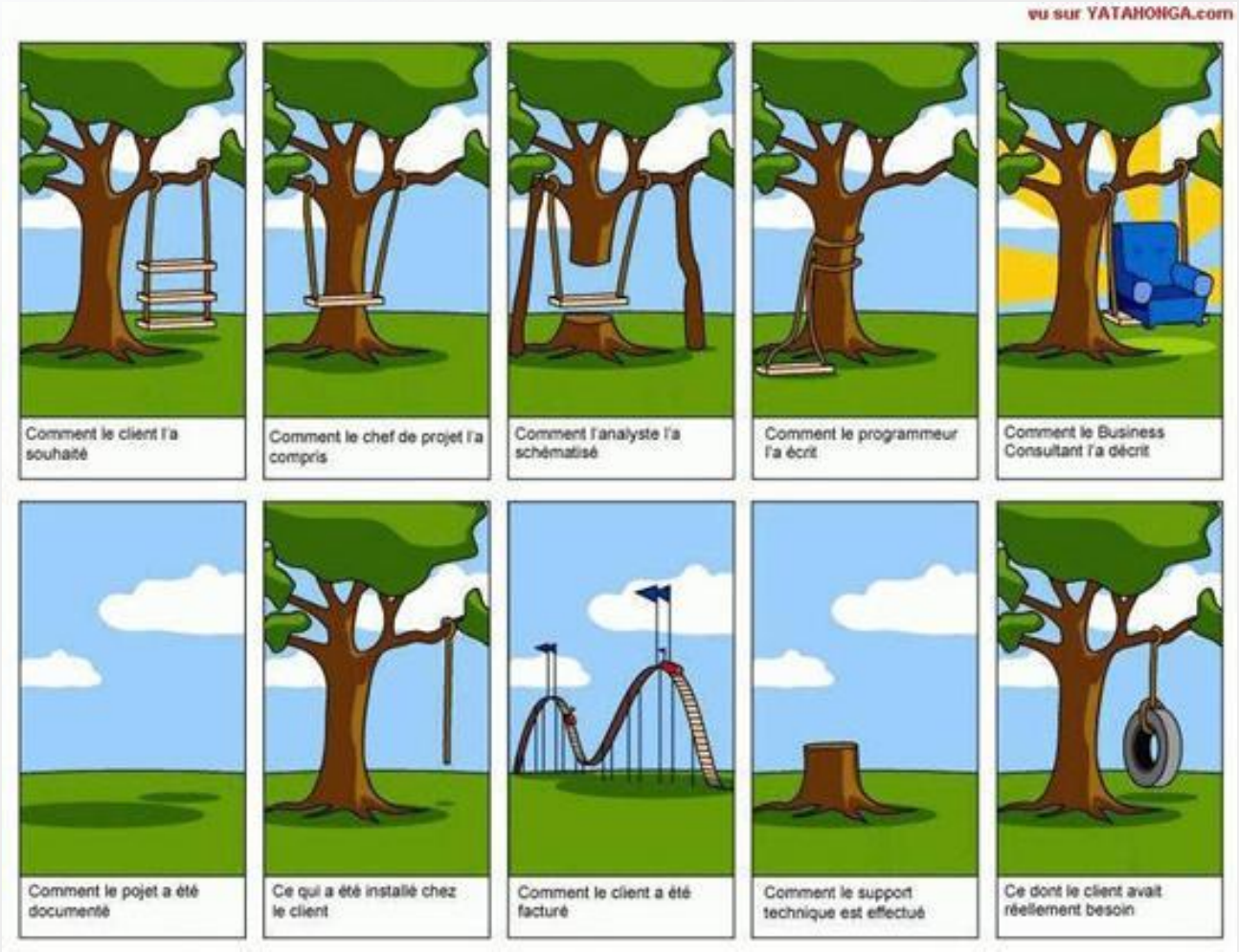
Les difficultés liées aux tests

- Impossible de réaliser un test exhaustif (variables)
- Qualité des tests dépend données utilisées
- Impossible de supprimer l'erreur humaine
- Difficultés d'ordre psychologique, culturel
- Manque d'intérêt pour les tests
- Taille et complexité des programmes
- Différence entre environnement de développement et de production
- Perte d'information naturelle

Difficulté formelle : pas d'algorithme capable de prouver l'exactitude d'un programme

Concernant la perte d'information…





Les différentes façons de classifier les tests

Il n'existe pas de classification officielle

Chaque ouvrage, chaque auteur, chaque site définit à sa manière les différentes techniques de tests.

Selon le mode d 'exécution

- Manuel :
  - Les tests sont exécutés par le testeur qui vérifie les traitements… et compare les résultats obtenus avec ceux attendus.
  - Ces tests sont fastidieux et offrent une plus grande possibilité d'erreurs humaines. Ces tests sont très vite ingérables dans le cas d'applications de grandes tailles.
- Automatique :
  - Le testeur est en partie déchargé des tests dans la mesure où les tests sont réalisés par des outils (JUnit par exemple dans le monde Java).

Selon les modalités du test

- Statiques :
  - Les tests sont réalisés «par l'humain» (testeur), sans machine, par lecture du code dans le but de trouver des erreurs (revue de code…).
- Dynamiques :
  - On exécute le système de manière à tester l'ensemble des caractéristiques. Chaque résultat est comparé aux résultats attendus.

Selon les méthodes de tests

- Structurels (Boîte blanche) :
  - Les tests structurels reposent sur des analyses du code source.
- Fonctionnels (Boîte noire) :
  - Les tests fonctionnels reposent sur une spécification du programme. Le code source du programme n'est pas utilisé. Les tests fonctionnels permettent d'écrire les tests bien avant le « codage ».

- **Il est parfois utile de combiner ces deux méthodes.**

Selon les niveaux de tests

- **Il s'agit de tests réalisés tout au long de la vie du logiciel (cycle de vie).**
  - Unitaires :
    - s'assurer que les composants logiciels pris individuellement sont conformes à leurs spécifications et prêts à être regroupés.
  - D'intégration :
    - s'assurer que les interfaces des composants sont cohérentes entre elles et que le résultat de leur intégration permet de réaliser les fonctionnalités prévues.
  - Système :
    - s'assurer que le système complet, matériel et logiciel, correspond bien à la définition des besoins tels qu'ils avaient été exprimés. On parle de validation ou de recette.
  - De non régression :
    - vérifier que la correction des erreurs n'a pas affecté les parties déjà développées et testées. Cela consiste à systématiquement rejouer les tests déjà exécutés

Selon les caractéristiques des tests

- De Robustesse :
  - Permet d'analyser le système dans le cas où ses ressources sont saturées ou bien d'analyser les réponses du système aux sollicitations proche ou hors des limites des domaines de définition des entrées.
- De performance :
  - Permet d'évaluer la capacité du programme à fonctionner correctement vis-à-vis des critères de flux de données et de temps d'exécution.

Bonnes pratiques

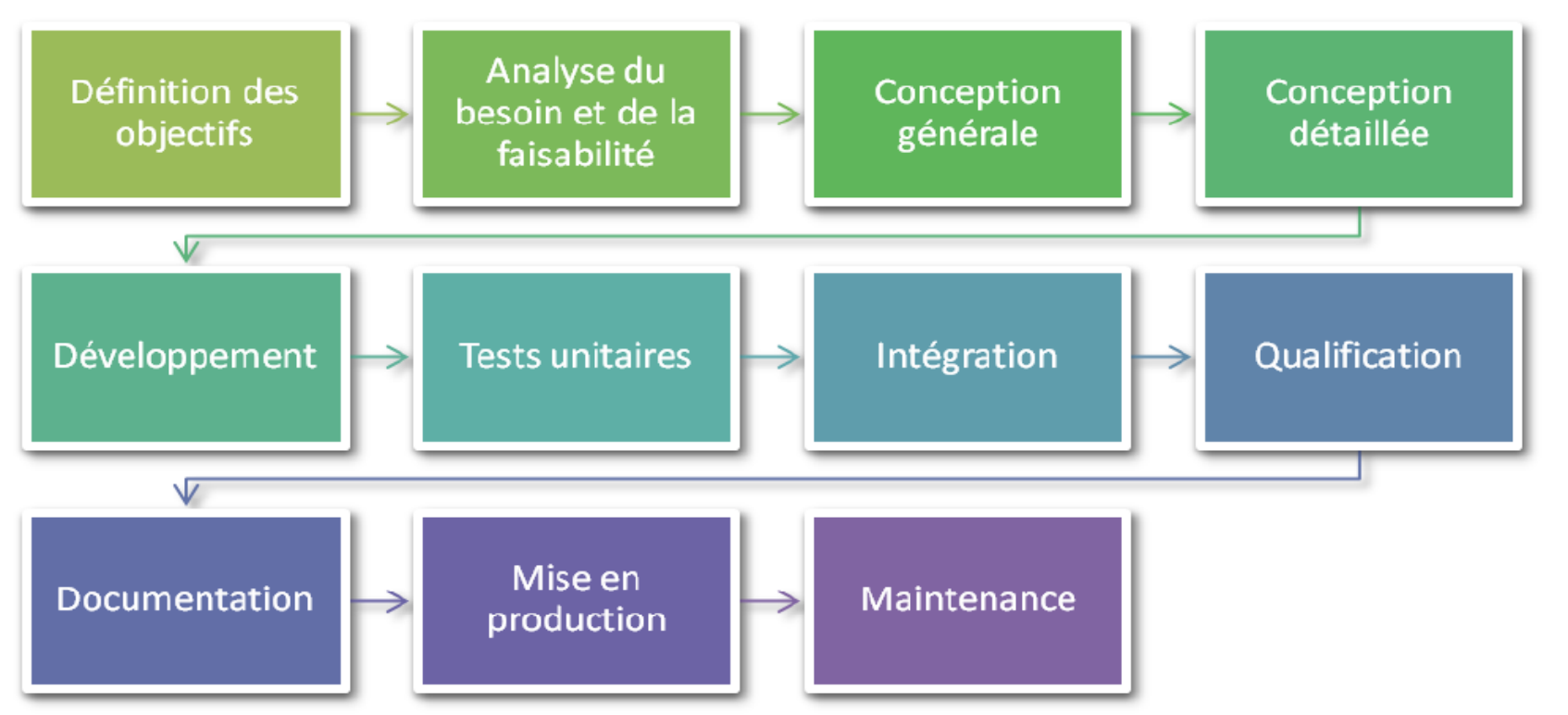
- Si possible faire tester par un autre développeur
- Ne jamais partir du principe qu'un test ne trouvera pas d'erreurs
- Examiner et mémoriser les rapports de tests.
- A la moindre modification ne pas hésiter à refaire les tests (test de non régression)

Tests et Cycles de vie

- Tests et Cycles de vie
- Les tests UNITAIRES
- Les Tests d'INTEGRATION
- Les tests de CHARGE
- Les tests de VALIDATION

Cycle de vie générique

- Un projet informatique va généralement suivre le cycle de vie suivant :



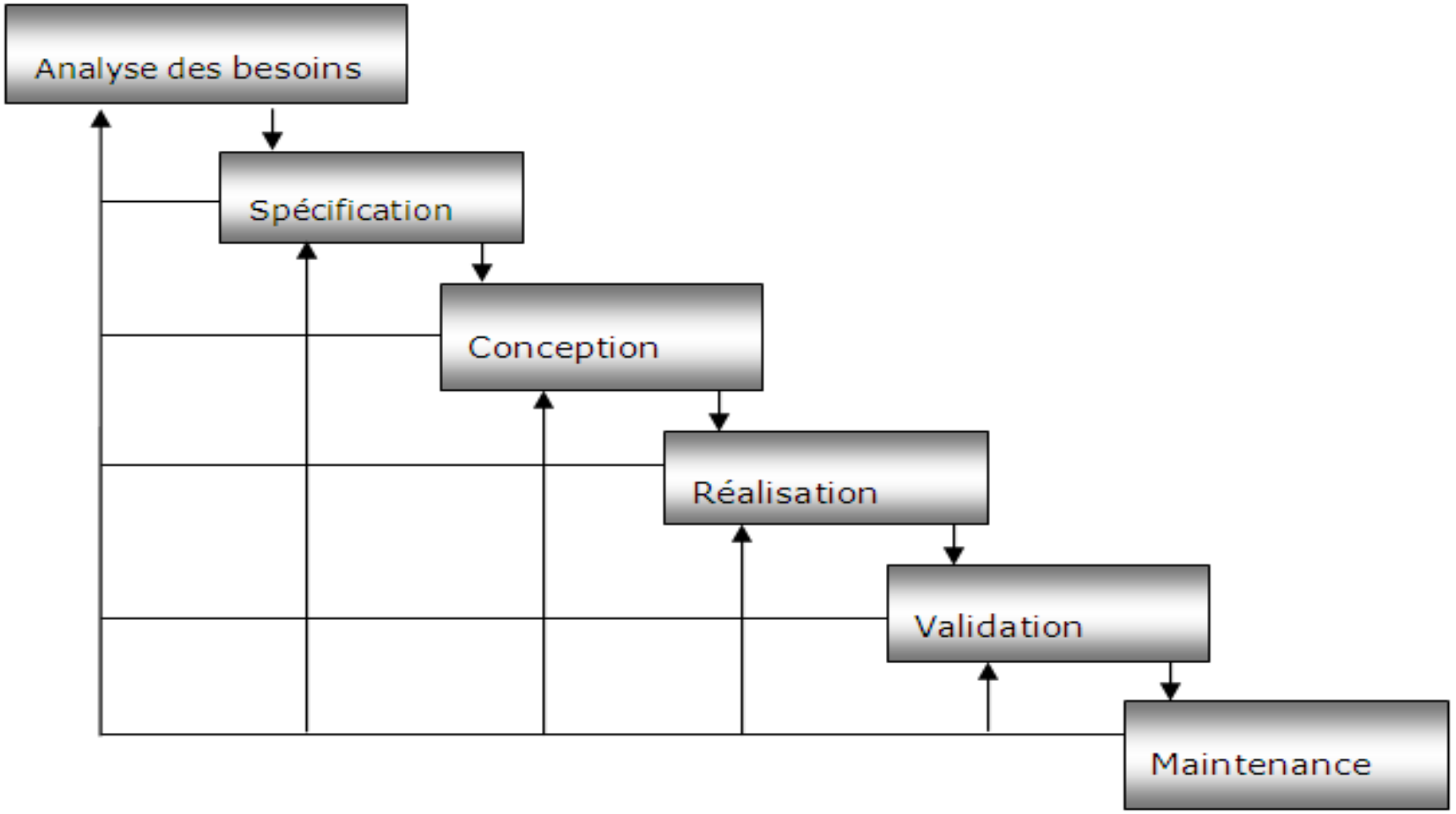
Tests et Cycles de vie

- **3 Cycles de vie principaux**
  - Cascade
  - En V
  - En spirale

Cycle en Cascade

- Définit les phases séquentielles
- A la fin de chaque phase, des documents sont crées pour en vérifier la conformité
- Si c'est bon on passe à la phase suivante
- Si ce n'est pas bon, on retourne en arrière (Feedback)

Cycle en Cascade



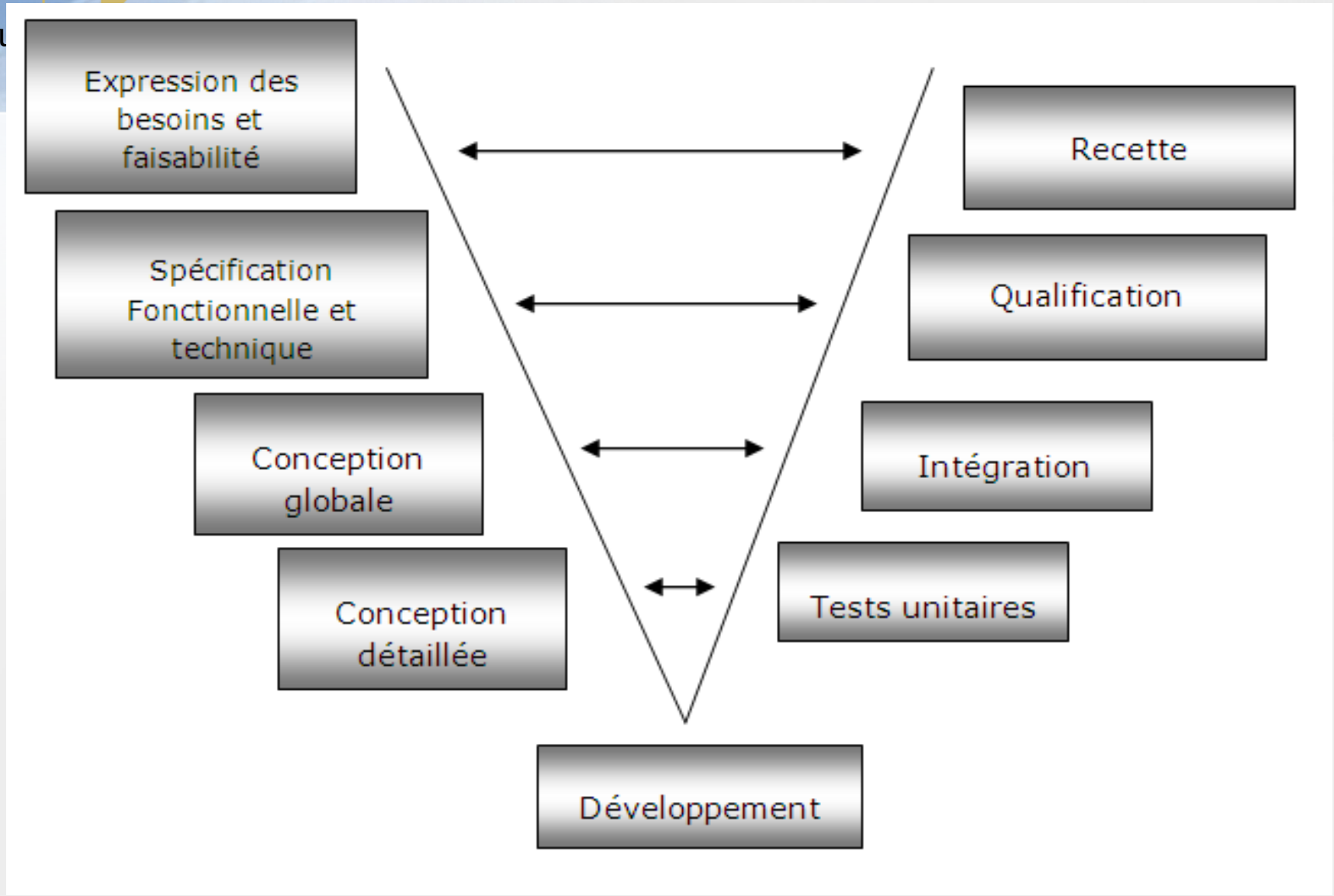
Source : Cours CNAM Génie Logiciel

Cycle en V

- Créé suite au manque de réactivité du modèle en cascade.
- Il part du principe que les procédures de vérification de la conformité doivent être élaboré dès les phases de conception
- Le client connait son besoin dans le détail
- Standard depuis les années 1980

Cycle en V

pou



Cycle en Spirale

- **Il est basé sur une approche et une évaluation des risques. Un risque identifié est assigné d'une priorité. Un prototype possède son cycle de vie propre**

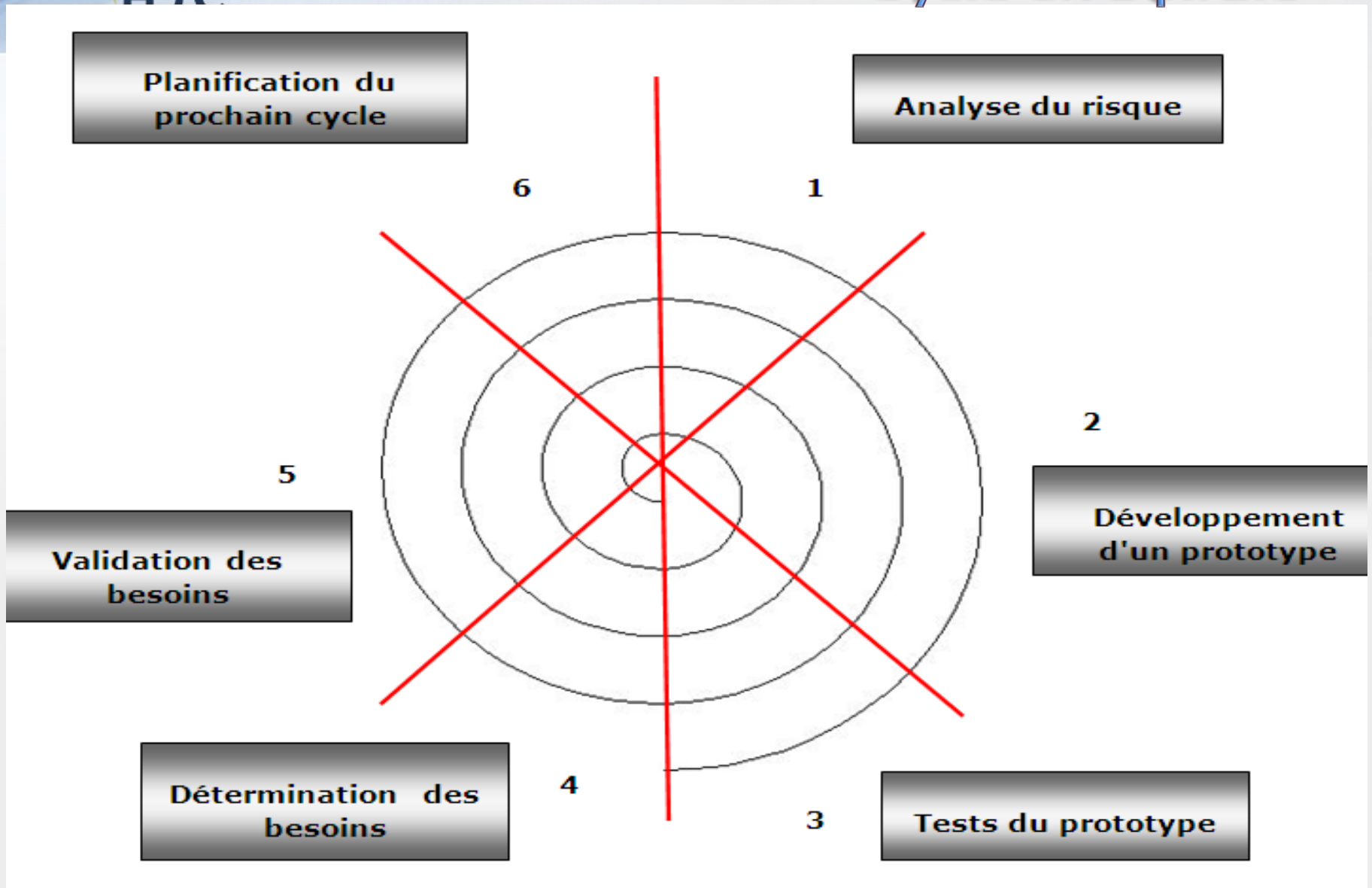
- **Le cycle en spirale couvre l'ensemble du cycle de vie de développement mais ajoute une dimension managériale et donc non technique**

- **Chaque cycle de la spirale comprend 6 phases :**
  - Analyse du risque
  - Développement d'un prototype
  - Test du prototype
  - Détermination des besoins
  - Validation des besoins
  - Planification du prochain cycle

Steeve ASSOUS ©

pour AJC

Cycle en Spirale



Conclusion sur les cycles de vie

- **Chaque cycle de vie contient les mêmes grandes phases du projet :**
  - Analyse du produit
  - Phase de conception
  - Phase d'intégration
  - Phase de validation

Tests Unitaires : Définition

- Vient du fait qu'une partie du code est appelé Unit
- Ce type de test va vérifier un module du code et qu'il fonctionne de manière indépendante du reste
- Respect aussi des spécifications fonctionnelles
- Ils peuvent être manuels ou automatisés par des logiciels

Tests Unitaires : Formalisme

- On va créer une fiche de test unitaire qui contient une liste (ou aide mémoire) pour les grandes actions
- Elle permet de préparer les tests
- Chaque analyste ou chef de projets doit la remplir afin d'assurer un passage en recette dans les meilleures conditions
- Donne une garantie de qualité lors du développement (ce n'est pas une contrainte mais un outil !)

Tests Unitaires : Conditions

- **Pour réaliser des tests unitaires il nous faut :**
  - des jeux de données (fictives, de production, ancien jeux de tests)
  - des ressources (documents de spécifications, scénarios, fiches de tests, tests précédents)
  - Une démarche

Tests Unitaires : Démarches

- **Les différentes démarches :**
  - Analyse statique
  - Analyse dynamique structurelle
  - Analyse dynamique fonctionnelle
  - Boite noire
  - Boite blanche

Tests Unitaires : Analyse Statique

- Cette notion d'analyse permet d'obtenir des informations sur le comportement du programme sans réellement l'exécuter
- On utilise des méthodes comme les nombres cyclomatiques, la mesure de Halstead ...

Tests Unitaires : Analyse Dynamique Structurelle

- Consiste à vérifier la structure du code ainsi que les variables
- On parcourt tous les nœuds, les arcs, et chemins du programme
- On peut vérifier ainsi si un test
  - If … Then … Else ….

  n'est pas utilisé

Tests Unitaires : Analyse Dynamique Fonctionnelle

- Consiste à vérifier le service rendu (la fonction) mais pas comment il est rendu
- On valide les règles de gestion
- La difficulté réside au choix des données de tests pour obtenir les résultats attendus

Tests Unitaires : Boîte Noire / Blanche

- **Noire :**
  - On vérifie que les sorties obtenues sont celles prévues pour les données entrées
  - Le fonctionnement interne n'est pas accessible
  - Souvent utilisé pour les tests de non régression, de robustesse et de charge
- **Blanche :**
  - On valide le code et on vérifie qu'il n'y a pas de plantage
  - On ne teste pas le fonctionnel mais tous les chemins possibles du programme

Tests d'intégration : Définition

- On regroupe chaque partie testée unitairement afin d'établir une nouvelle
- version du produit
- Le test d'intégration a pour but de tester que tout fonctionne ensemble
- Il existe une intégration incrémentale et une intégration globale

Les Tests de CHARGE : Définition

- **On expose l'application à des conditions d'exploitations pour valider le système.**
- **Les tests de charge consistent à exposer une application à des conditions d'exploitation et d'utilisation les plus proches de la réalité afin de valider le comportement du système.**

Objectifs :

Tester la performance
Maintien des fonctionnalités sur une montée en charge
Fiabilité (plateforme, BDD…)

Tests de Charge : Types de Tests

Principaux types de tests de charge :

- Test de performance
- Test de dégradation des transactions
- Test de stress
- Test d'endurance, de robustesse, de fiabilité
- Test de capacité, de montée en charge

Tests de Charge

- **Quand mener les tests de charge ?**
  - Le plus tôt possible dans le processus de développement fin de mettre en évidence rapidement les défauts d'infrastructure.
  .
  - Si l'on sait que le coût des corrections des défauts non découverts progresse exponentiellement à chaque phase suivante du cycle de développement, il n'en est que plus essentiel de débuter les tests de montée en charge le plus tôt possible.

Tests de Validation : Objectifs

- **Vérifier que toutes les exigences du cahier des charges soient respectées. Ils ont lieu immédiatement après les tests d'intégration.**

- **2 approches :**

  - Identifier et tester toutes les fonctions du logiciel
  - Tester les caractéristiques du logiciel (interfaces…)

- **En pratique, les 2 sont utilisées**