

# Qualité du logiciel: éléments normatifs

Matthieu Amiguet

2004 – 2005



# La qualité

## Définition AFNOR NF x50-120

Aptitude d'un produit ou d'un service à satisfaire les besoins des utilisateurs

## Critères de qualité

- McCall, “Factors in software quality” (1977) en propose 11

**Confidentialité** le logiciel est protégé contre tout accès non autorisé

**Couplabilité** aptitude du logiciel à être “couplé” à un autre (échange de données, coopération)

**Efficacité** minimisation de la consommation des ressources (CPU, mémoire, disque, ...)

**Maniabilité** utilisation agréable et facile

**Robustesse** aptitude à conserver un comportement conforme aux besoins même dans des situations imprévues

## Critères de qualité – suite

- Maintenabilité** facilité de localisation et correction des bugs résiduels
- Adaptabilité** facilité d'ajout/modification/suppression de fonctionnalités
- Portabilité** facilité à changer d'environnement d'exécution (Système d'exploitation, matériel, ...)
- Testabilité** facilité de réalisation de tests
- Réutilisabilité** possibilité de réutiliser des portions de code dans le cadre d'un autre projet
- Correction** degré de conformité par rapport aux spécifications.

## Facteurs de qualité

- L'*auditabilité* (mémorisation des accès aux données et aux programmes) et la *protection* (du code, des données, hors exploitation ou en exploitation) ont une influence déterminante sur la confidentialité
- La *standardisation* (des données, des interfaces) a une influence déterminante sur la couplabilité
- L'*efficacité mémoire*, l'*efficacité des périphériques*, l'*efficacité du temps d'exécution*, ont une influence déterminante sur l'efficacité
- La *communicabilité* (facilité de dialogue homme-machine), l'*exploitabilité* (facilité de mise en oeuvre et d'utilisation), la *facilité d'apprentissage* ont une influence déterminante sur la maniabilité

## Facteurs de qualité – suite

- La *précision* (exactitude des résultats obtenus) et la tolérance aux fautes (limitation ou suppression des effets d'une perturbation) ont une influence déterminante sur la robustesse
- La *lisibilité* (compréhension d'un document ou du code par simple lecture), la *modularité* (décomposition d'un logiciel en éléments de taille limitée), la *simplicité* (facilité de compréhension liée à l'absence de bruits superflus) et la *traçabilité* (existence de liens structurés entre les différentes représentations des composants) ont une influence déterminante sur la maintenabilité et sur l'adaptabilité
- *L'observabilité* (facilité de détection des non-conformités) a une influence déterminante sur la maintenabilité

## Facteurs de qualité – suite

- *L'expansibilité* (facilité d'accroissement des fonctionnalités, des zones de données, de la taille des programmes, du nombre d'utilisateurs...) a une influence déterminante sur l'adaptabilité
- La *banalité d'emploi* (indépendance par rapport à une application), l'*indépendance système* et l'*indépendance matériel* ont une influence déterminante sur la portabilité.

## Mettre en place des procédures qualité. . .

- L'organisation du processus: découper le processus pour le maîtriser (QUOI ?)
- Les ressources humaines: les équipes doivent être motivées pour mettre en place des procédures de qualité (QUI ?)
- L'utilisation de techniques, méthodes, outils reconnus (COMMENT ?)
- Considérations managériales, politiques et économiques: évaluer le retour sur investissement par une analyse coûts-bénéfices (POURQUOI ?).

## Organiser le processus

- Découpage en activités
  - Séquentielles (verticales): Spécification, conception, ...
  - Parallèles (horizontales): Gestion de projet, gestion des configurations, ...
- Le projet doit être découpé en éléments maîtrisables
  - Efficacité
  - Correction
  - Motivation
- Pour chaque activité, il faut définir
  - Qui fait
  - Qui approuve
  - Qui vérifie
  - Qui valide
  - ...

# Maîtrise de la qualité

- La maîtrise de la qualité passe par
  - une planification systématique de ce qui doit arriver
  - la mesure des écarts entre ce plan et la réalité
  - la correction de la trajectoire
- Pour chaque projet, il faut donc
  - Prévoir ce que l'on doit faire
  - Écrire ce que l'on a prévu
  - Faire ce qu'on a écrit
  - Vérifier que le système qualité est adéquat et que tout se déroule comme prévu
  - Corriger les écarts
  - *Conserver une trace* de tout ce qu'on fait.

# Assurance qualité

## Définition AFNOR

Mise en oeuvre d'un ensemble approprié de dispositions préétablies et systématiques destinées à donner confiance en l'obtention de la qualité requise

- Dans les faits, les normes qualités insistent sur ces trois points
  - Communication
  - Contrôle
  - Organisation.

# Organisation

- Le processus de développement est organisé en différentes phases
  - Définition des besoins
  - Spécification
  - Planification
  - Conception
  - ...
- L'enchaînement de ces phases est prédéfini et constitue le *workflow* du projet
- Chaque activité débouche sur un produit intermédiaire dont la production détermine la fin de l'activité et qui devra être contrôlé avant de passer à l'activité suivante.

# Contrôle

- L'assurance qualité passe par des contrôles réguliers et inclut
  - la validation (du latin *validare*, déclarer valide) permet de répondre à la question "Sommes nous en train de faire le bon produit ? "
  - la vérification (du latin *veritas*, la vérité): répond à la question "Est-ce que nous faisons le produit correctement ?"
- Les erreurs sont de plus en plus coûteuses à réparer lorsqu'elles sont découvertes tard dans le cycle de vie
  - rôle primordial de contrôles intermédiaires .

## Inspection et revues

- La validation et la vérification sont en partie garanties par la mise en place des inspections et revues pour tous les produits intermédiaires du développement
  - documents de spécification, de conception, ...
  - code
  - jeux de tests
  - prototypes
  - ...

**Inspection** relecture critique d'un document visant à améliorer sa qualité

**Revue** Réunion permettant de valider une phase du cycle de vie.

# Inspection

- Généralement faite par une équipe indépendante du projet
- Pour être profitable, doit donner lieu à la rédaction d'une *fiche de défauts*
  - Échelle de gravité
  - Définition des responsabilités pour la correction
  - ...
- Les inspections sont à la base des décisions prises en revues.

# Revue

- Différents types

**Revue produits** état d'un projet sous ses différents aspects:  
Technique, Financier, Commercial, Calendrier, ...

**Revue techniques** fournissent au marketing et à l'unité de développement une évaluation des aspects techniques du projet et des coûts de réalisation

**Réunions de décision** valident le passage à la phase suivante et font bien souvent suite à l'une des deux précédentes.

# Communication

- Pour produire un logiciel de qualité, la communication est indispensable
  - entre développeurs et environnement (clients, support, ...): cahier des charges, documentation utilisateur, dossier de conception, ...
  - entre développeurs: commentaires de code, résultats de tests, ...
- Pour maîtriser cette communication, il faut la *documenter*.

## Les documents

**Manuel Qualité** décrit les principes généraux, les règles applicables à toutes les activités

- peut contenir jusqu'aux procédures détaillées représentant le savoir-faire et l'expertise de l'entreprise
- par conséquent interne à l'entreprise

**Manuel Assurance Qualité** décrit les dispositions générales pour donner confiance dans la capacité de l'organisation à délivrer la qualité

- doit être présentable à un client
- peut être un élément contractuel officiel.

## Les documents – suite

**Plan Qualité** décrit les standards et procédures applicables pour tout projet

- général et interne à l'entreprise

**Plan d'Assurance Qualité** décline le Plan Qualité applicable à un projet particulier

- par essence ouvert au client
- concerne à la fois les activités de contrôle des produits et les activités de contrôle des processus conduisant à la création des produits ou services.

# Normes de qualité

- AFNOR

- Recommandation de Plan qualité logiciel Z67-130
- Guide de rédaction de Plan qualité logiciel Z67-130
- Gérer et assurer la qualité : document AFNOR
- Du bricolage à l'industrialisation : la qualité des logiciels, J-P Martin

- DGA

- Méthodologie de développement des logiciels intégrés dans les systèmes militaires : GAM-T-17 version 2 (juin 88)

- IEEE

- IEEE 730, 732 et 738
  - Std 828-1983 Standard for software test documentation
  - Std 829-1983 : Standard for software configuration management

## Normes de qualité – suite

- ESA
  - PSS : PSS01, PSS05 Assurance qualité logiciel
- OTAN
  - AQAP 13 et 14 (mai 1984)
- DOD
  - DoD-STD-2167A : Military standard-Defense system software development (2/88)
- AFCIQ
  - Recommandation de Plan assurance qualité logiciel (V0 du 23-03-89)
  - Recommandation de Plan de développement logiciel (V1 du 17-06-88)

## Normes de qualité – suite

- ISO
  - ISO 9001 et associés
- Bell Canada
  - Trillium
- ...

# La norme ISO 9001

**ISO 9000** ensemble de recommandations et standards pour la garantie de la qualité dans les relations clients-fournisseurs (pas spécialement logiciel).  
Comprend notamment:

**ISO 9000-1** recommandations pour l'utilisation de ces standards

**ISO 9001** le standard à utiliser pour la fourniture de logiciels, bien qu'il ne soit pas spécifique à ce domaine

**ISO 9003** guide pour l'utilisation des standards ISO 9001 pour la fourniture de logiciels.

# Philosophie ISO 9001

## Philosophie ISO 9001

- Toute opération influençant la qualité doit être sous contrôle
- Ce contrôle doit être visible
- La certification permet d'obtenir une large reconnaissance de ce contrôle
  - Elle est payante...
  - ... et valable trois ans.

## Les éléments de qualité

- Le chapitre 4 de la norme définit les 20 éléments de qualité à respecter:
  - Responsabilités du management
  - Définition d'un système de qualité
  - Analyse du contrat entre client et fournisseur
  - Contrôle de la conception
  - Documentation et contrôle des données
  - Spécification des achats et fournitures
  - Contrôle des produits fournis par le client
  - Identification des produits et traçabilité
  - Contrôle du processus
  - Nécessité de mettre en oeuvre des tests et inspections
  - Contrôle des inspections et tests, mesures des outils de tests
  - Statut des tests et inspections
  - Contrôle des produits non conformes

## Les éléments de qualité – suite

- Suite des 20 éléments de qualité à respecter:
  - Actions correctives et préventives
  - Emballage, stockage, livraison,
  - Contrôle des enregistrements concernant la qualité
  - Audit qualité internes
  - Organiser la formation
  - Service après vente
  - Techniques statistiques
- On voit donc que ISO 9001 n'est pas spécifique au logiciel.

# ISO 9003

- ISO 9003 d'un guide de recommandations équivalent à un standard
- Lors de toute certification, il faudra justifier tout manquement à ces recommandations.

## Définir le cadre de l'assurance qualité

- Responsabilités du management
  - définition d'une politique qualité
  - mise en place d'une organisation
  - validation périodique du système qualité
- Définition d'un système qualité
  - La politique qualité doit être documentée dans un manuel qualité conforme aux normes en vigueur et aux habitudes de l'entreprise
  - Il est important que les procédures qualité mises en place trouvent l'agrément des développeurs et ne soient pas perçues comme un frein à leur créativité mais plutôt comme un cadre rassurant dans lequel ils pourront évoluer et produire un travail de qualité
- Évaluation du contrat
  - Il est tout à fait indispensable de ne pas s'engager sur un contrat irréaliste.

# Contrôler le développement

- Planification du développement et de la qualité:
  - définition du calendrier
  - des différentes phases du projet
  - des critères qualité
- Organisation du travail
  - En particulier organisation des espaces de travail.

## Contrôler le développement – suite

- Contrôle des Spécifications (Design input)
  - ISO 9000 requiert des spécifications rigoureuses et complètes
  - En logiciel tout le monde sait que ceci est en général un voeu pieux
  - Un autre paragraphe des recommandations prévoit donc les procédures de prise en compte de changements dans les spécifications.
  - Afin de définir ces spécifications, des outils peuvent être utilisés (en particulier tous les outils d'aide à l'analyse tels que Rational Rose, ...).

## Contrôler le développement – suite

- Sortie de la conception (Design output)
  - La conception peut être réalisée à la main
  - elle peut également être générée plus ou moins automatiquement.
  - Dans tous les cas, la phase de conception doit produire une architecture logicielle et de la documentation
- Contrôle de la Conception (Design review)
  - Dans tous les cas la conception nécessite un contrôle rigoureux par des inspections qui peuvent être réalisées au moyen de listes de défauts typiques.

## Contrôler le développement – suite

- Validation vérification (Design verification, Design validation)
  - procédures de validation/vérification du code
    - analyse statique
    - tests fonctionnels
    - tests structurels
  - Le Béta test pris en charge par le client doit explicitement figurer dans le contrat si une telle décision de qualification est adoptée
  - Il est tout à fait fondamental d'être capable de rendre compte des résultats des tests et des jeux de tests qui ont été effectués
  - Les critères d'arrêt doivent être définis dans un plan de tests, on doit pouvoir prouver qu'on les a atteints.

## Contrôler le développement – suite

- Modification de la conception et des spécifications
  - Il faut vérifier la cohérence des changements demandés avec le développement déjà réalisé
  - ISO9000 requiert une procédure rigoureuse d'acceptation des modifications.

## Gérer les activités de support

- Gestion de configurations: spécifique aux projets logiciels
  - Les produits logiciels sont constitués de différents éléments qui évoluent au cours du temps et qui peuvent différer d'une installation à l'autre
  - Contrôler l'ensemble des données constituant le système:
    - documents (sources, jeux de tests, plans d'intégration, . . .)
    - assurer la cohérence des divers composants
    - construire/reconstruire un système
- Gestion de la documentation et des données
  - Quiconque en a besoin doit pouvoir y accéder
  - Tout document doit être approuvé
  - Versions des documents cohérentes entre elles et avec le code qu'elles documentent.

# Traçabilité

- Suivi tout au long du cycle de développement des liens entre cahier des charges, spécifications, conception et codage
- La norme impose de savoir répondre aux questions suivantes
  - De quel document initial cette spécification s'inspire-t-elle ?
  - À quelle spécification, à quel document de conception ce bout de code est il relié ?
  - Quelles sont les corrections ou améliorations qui ont été réalisées dans tel module ?
  - A partir de quel code source cet exécutable a-t-il été généré ?
  - Avec quel outil cet exécutable a-t-il été généré ?
  - Qu'est il advenu de chaque rapport d'incident ? A-t-il été pris en compte, corrigé ? la nouvelle version a-t-elle été distribuée ?

## Traçabilité: attention danger!

- C'est souvent à cause de ce chapitre que les certifications échouent
  - La mauvaise version d'un fichier source a été cataloguée
  - Une erreur est répertoriée comme réparée et ne l'est pas
  - Un manager ou un chef de projet a été incapable de montrer quelles versions des sources étaient utilisées au moment des tests
  - Incapacité de montrer les différentes demandes de modifications ou rapports d'incidents.

## Autres points

- Contrôle de production (du media de distribution)
- Inspection et tests (du media de distribution)
- Statut des inspections et tests
  - On doit pouvoir à tout moment savoir si un document, un code source...a été inspecté, validé ou est en attente de validation
  - En aucun cas un document non validé ne doit servir de base à de nouveaux développements
  - Les documents inspectés et validés doivent être conservés dans un espace différent de ceux qui ne le sont pas
- Contrôle des produits non-conformes
- Actions correctives et préventives.

## Autres points

- Stockage, livraison, emballage
- Contrôle des enregistrements qualité
- Audit qualité interne
- Formation du personnel
- Service après-vente
- Méthodes statistiques
  - Nombre de pannes
  - Nombre d'erreurs découvertes (par le client, par le fournisseur)
  - Temps moyen entre deux pannes
  - Durée moyenne d'indisponibilité.

# Capability Maturity Model

- Développé par: Software Engineering Institute, Carnegie Mellon University
- À partir des années 70 pour le compte du DOD
- Largement adopté dans le civil
- Avantages:
  - Développé spécifiquement pour le logiciel
  - Permet de se positionner dans une grille et d'*évoluer* dans cette grille
- Cependant, moins répandu et réputé en Europe que la norme ISO 9000.

## Philosophie CMM

- La grille CMM permet de classer une organisation qui développe du logiciel selon sa compétence
- Distingue 5 niveaux de maturité:
  - 1 Initial
  - 2 Reproductible
  - 3 Défini
  - 4 Géré
  - 5 En optimisation constante.

## 1 – Niveau initial

- Le logiciel est développé sans méthode prédéfinie
  - Le développement repose sur la compétence de quelques personnes
  - Il est impossible de récupérer l'expérience acquise dans un projet lors du développement d'un autre projet
  - On ne peut prédire en terme de gestion de projet des éléments aussi importants que la taille ou le coût d'un projet
  - Les réactions se font essentiellement en fonction des crises et non pas de façon systématique et planifiée
- ~> Beaucoup d'organisations sont encore assez proches de ce niveau.

## 2 – Niveau reproductible

- Il existe un système commun de management de projet et des techniques de contrôle
- La gestion de projet et la planification sont faites sur des bases reproductibles
- Des activités de mesures commencent à être mises en place, en particulier au niveau des coûts et des délais
- On réagit de façon planifiée et non plus en fonction des crises
- Les problèmes sont traités au fur et à mesure qu'ils arrivent et ne sont pas accumulés jusqu'à provoquer une crise majeure
- Les mesures utilisées pendant un projet permettent de prévoir ce qui sera susceptible de se passer pour les projets futurs.

### 3 – Niveau défini

- Il existe un système commun pour toutes les activités de développement de logiciel à la fois du point de vue managérial et technique
- Des mesures sont faites régulièrement pour améliorer le processus
- Des revues sont mises en place afin de garantir la qualité du logiciel.

## 4 – Niveau géré

- Le procédé de développement logiciel est stable et permet de garantir un niveau constant de qualité
- Des objectifs précis de qualité et de productivité sont affectés à chaque projet
- Des mesures régulières permettent de garder sous contrôle ces deux indicateurs, et
- Des actions correctrices sont prises dès qu'une divergence par rapport aux objectifs est constatée
- Des procédés de mesure statistiques sont mis en place afin de déterminer s'il s'agit d'un manquement passager aux objectifs ou bien s'il s'agit d'une divergence grave par rapport aux standards de productivité et de qualité.

## 5 – Niveau en optimisation constante

- Le processus de développement porte en lui les moyens de réaliser sa propre optimisation
  - Des contrôles statistiques sur le processus sont utilisés pour guider l'organisation
  - Le processus intègre un feed back provenant de ces mesures
- ↪ De nombreuses entreprises ont atteint les niveaux 2 et 3
- ↪ Pratiquement aucune n'est encore arrivée aux niveaux 4 et 5.

## Les apports du CMM

- Des résultats publiés montrent que se positionner dans la grille CMM permet d'accroître la rentabilité
- Exemple: le département Logiciel de Hughes Air Craft, Fullerton, Californie
  - a dépensé environ 500'000\$ entre 87 et 90 pour améliorer son processus de production de logiciels
  - Pendant cette période passés du niveau 2 au niveau 3, avec de bons espoirs d'atteindre les niveaux 4 et 5
  - Estimation de l'économie engendrée: 2 millions de \$ par an
    - diminution du nombre de crises
    - diminution du nombre d'heures supplémentaires
    - amélioration de la compétence des employés
    - diminution du turn- over.

## Comparaison Iso 9000 – CMM

- ISO 9000 est une norme, CMM n'en est pas une
- CMM est dédié à l'industrie du logiciel, ISO 9000 définit un cadre pour les rapports clients fournisseurs
- CMM est plus détaillé et spécifique
- ISO 9000 établit un niveau acceptable de management de projet auquel le fournisseur doit souscrire pour que les relations client-fournisseur puissent s'établir avec certaines garanties de qualité pour le client
- CMM permet au fournisseur de s'auto-évaluer et de progresser sur une grille allant de 1 à 5.

## Sources principales

- [http://www.enseeiht.fr/lima/vision/Membres/MB/COURS/PROJETS/3AI/02-03/ProjetN7\\_docQualite.pdf](http://www.enseeiht.fr/lima/vision/Membres/MB/COURS/PROJETS/3AI/02-03/ProjetN7_docQualite.pdf)
- <http://www.essi.fr/~hugues/GL/chapitre1.pdf>
- <http://perso-info.enst-bretagne.fr/~beugnard/cours/GL-3.pdf>

# Acronymes

**ISO** International Standard Organization

**IEEE** Institute of Electronics and Electrical Engineers

**AFNOR** Association Française de NORmalisation

**AFCIQ** Association Française pour le Contrôle Industriel de la  
Qualité

**ESA** European Space Agency

**DOD** US Department of Defense

**OTAN** Organisation du Traité de l'Atlantique Nord