Principes de bases de la qualité logiciel

Clémence Agrapart
clemence.agrapart@cnrs-orleans.fr

Lundi 11 octobre

Introduction

- **Contexte**

  - Le logiciel est aujourd'hui une part prépondérante dans le coût total d'un système informatique
  - La demande de nouvelles applications (numériques, performances, …) ne cessent de croître (historique)
  - Les utilisateurs sont de plus en plus exigeants en terme de fiabilité et sécurité (sélection)

  → Les pratiques qualité logiciel se généralisent (outils d'automatisation aux différentes étapes du cycles de vie du logiciel)

Introduction

- **Contexte**
  - Incitation de plus en plus pressante à la réutilisation des données, la reproductibilité, la pérennisation des données de la recherche *(Open date et « FAIR » data)*

→Qui s'applique aussi aux logiciels de recherche

Approche qualité système

- **Qu'est que la qualité ?**

La qualité est « l'aptitude d'un ensemble de caractéristiques intrinsèques d'un objet à satisfaire des exigences »
(selon NF ISO 9000_2015)

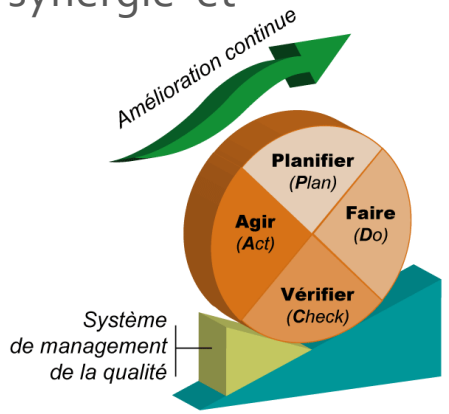
- Au quotidien, vous faites tous de la qualité (c'est du bon sens !) :
  - → Formalisée ou non
  - → Dans une démarche collective ou non

- La qualité s'applique à tous les domaines, tous les corps de métier. Elle permet de mettre en cohérence des bonnes pratiques, mieux s'organiser pour être plus efficace et éviter les dysfonctionnements.

Outil au service de la conduite des projets

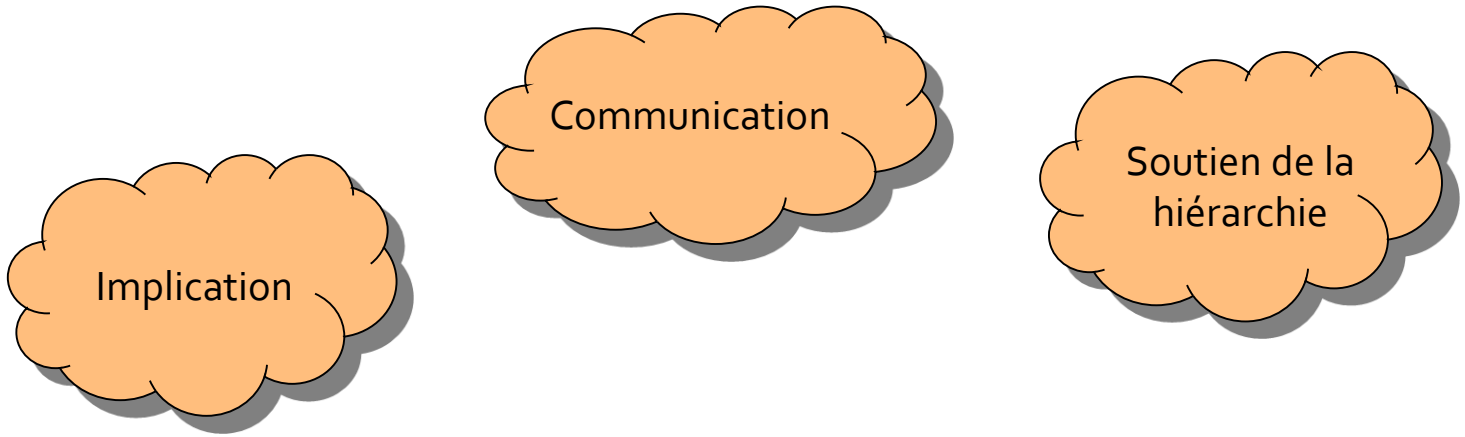
Approche qualité système

- **Comment fonctionne la démarche qualité ?**
  - On formalise pour capitaliser, transmettre et faciliter la communication (nouveaux entrants, turnover de personnel)
  - Au sein des projets, des équipes ou de l'entité, on définit des outils et des procédures qui permettent d'harmoniser les méthodes
  - On sensibilise les agents pour une meilleure synergie et visibilité des pratiques



Approche qualité système

- **A quoi ça sert ?**
  - Améliorer les règles de fonctionnement
  - Renforcer la confiance entre les équipes et les partenaires impliqués
  - Maîtriser les savoir-faire
  - Optimiser le travail et gagner en efficacité
  - Réduire les risques de dysfonctionnement



Approche qualité système

- **Quels sont les enjeux ?**

  - Les enjeux internes : fédérer, formaliser le fonctionnement, pérenniser les savoir-faire et le transfert de compétences, …

  - Les enjeux externes : valoriser l'organisation de l'entité, du projet, de votre travail ; gagner en compétitivité, favoriser la satisfaction des partenaires/parties intéressées (HCERES, client, …), mettre à disposition son savoir-faire et son expertise, …

Approche qualité produit

- **Qu'est que l'assurance produit ?**

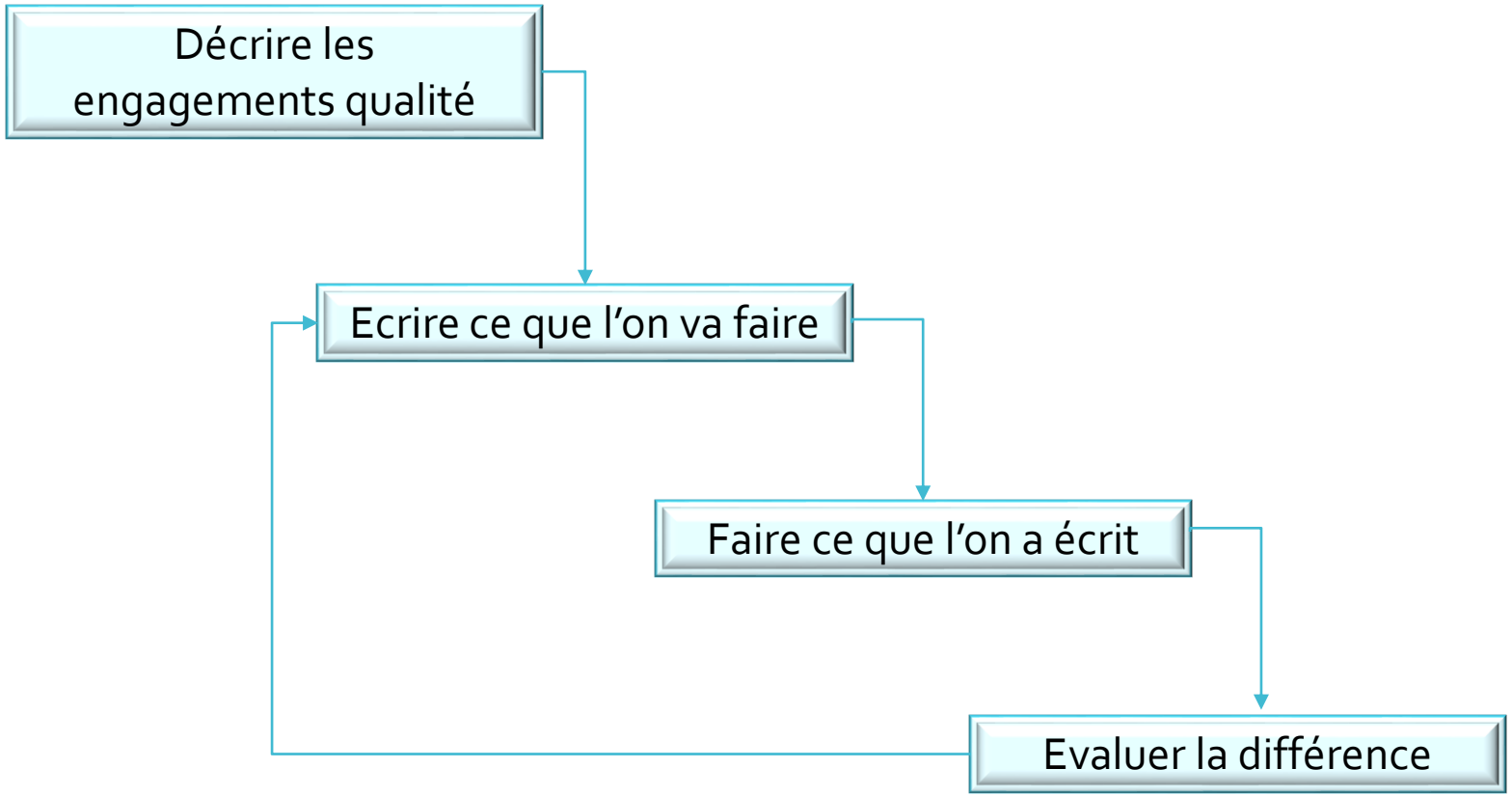
→ Orienter la qualité sur le produit et le projet

→ Ensemble des activités mises en application pour répondre aux spécifications qualité des partenaires (CNES, ESA, NASA, …) et assurer la qualité de l'instrument

Approche qualité produit

- **Pour quoi faire ?**
  - S'assurer de la conformité du produit / exigences
  - Donner une image globale sécurisée d'un projet au niveau des risques
  - Améliorer la gestion de projet
  - Apporter un support au management
- **Par quels moyens ?**
  - Investir dans la mise en place de méthodes
  - Mettre à disposition des outils et documents applicables
  - Identifier, évaluer, prévenir et maitriser les risques
  - Permettre un « Retour d'Expérience » et une capitalisation des savoirs – faire
- **Pour qui ?**
  - Chaque membre d'un projet a un rôle à jouer dans la démarche qualité
  - Interfère notamment avec le management, l'ingénierie, la technique

Approche qualité produit



- **En résumé**

Ecrire ce que l'on fait

Application au logiciel

Définitions

Qu'est-ce qu'un logiciel ?

- « Un logiciel est un ensemble de séquences d'instructions interprétables par une machine et d'un jeu de données nécessaires à ces opérations »
  Le logiciel est un produit à part entière qui doit justifier des mêmes activités que tout autre produit matériel.

Par logiciel, on entend : autonome, applicatif, outil de simulation, bibliothèque, framework, intergiciel, SE, plug-in, environnement de développement, logiciel embarqué, drivers, …

Qu'est-ce qu'un processus qualité ?

- Un processus est un ensemble d'activités corrélées ou en interaction qui utilise des éléments d'entrée pour produire un résultat escompté *(selon NF ISO 9000_2015)*

Qu'est-ce que la qualité logiciel ?

- « Ensemble des attributs et caractéristiques d'un produit ou d'un service qui portent sur sa capacité à satisfaire des besoins donnés » (*ANSI*)

→ C'est répondre aux objectifs d'utilisation, de maintenance et de portage en respectant les délais et les budgets définis

Application au logiciel

Définitions

Qu'est-ce que la conformité ?

- La satisfaction d'une exigence

Qu'est-ce que la non-conformité ?

- La non satisfaction d'une exigence

Qu'est-ce qu'un document ?

- Un ensemble d'informations saisies sur un support numérique ou papier

Application au logiciel

Définitions

L'assurance qualité logiciel s'inscrit dans une démarche d'assurance produit globale.

Les domaines suivants sont applicables au produit logiciel :

- Gestion de la documentation
- Gestion de la configuration
- Gestion des risques
- Suivi des tests
- Intégration et recette
- Retour d'expérience

Elle doit permettre notamment de répondre à des exigences applicables au logiciel, du développement à la maintenance de celui-ci. Il est important de vérifier et valider au travers de tests la bonne santé du code et de constamment veiller à la traçabilité qui lui est liée.

Choix des ingrédients

Recette

Respect des différentes étapes

Projet :
Faire un gâteau au yaourt

Cuisson

Dégustation

L' assurance produit

Faisabilité, Définition préliminaire

Liste des ingrédients

Protocole à suivre

Recette : Spécifications

Contrôle de la propreté du saladier, et du plan de travail

Définition détaillée

Achat de 6 œufs pour une utilisation de 3 œufs (redondance, anticipation du risque de perte d'œufs)

Sélection des ingrédients

Vérification des quantités : utilisation d'une balance

Respect de l'ordre d'ajout des ingrédients

Réalisation

Respect des spécifications

Anticipation du risque de grumeaux : utilisation d'un batteur électrique

Risque de brûlure (sécurité) : utilisation de gants de cuisine anti-chaleur

Qualification

Contrôle des paramètres :
- Temps
- Température

Suivi de la cuisson

Contrôle de conformité par rapport aux exigences des invités :
- gâteau aéré
- gâteau moelleux

Exploitation, Utilisation

Retour d'expérience : Si satisfaisant, réutilisation de la recette

Application au logiciel

Retour d'EXpérience

Le REX : *INSU-PRO-023 (« Mener le REX »)*

Le REX porte un regard sur la démarche développée, les méthodes employées, les productions réalisées, le rôle et le niveau d'implication des acteurs concernés, ainsi que sur les moyens utilisés.

Il s'agit là d'une auto-critique mais qui se veut être constructive en apportant une valeur ajoutée pour les projets suivants.

Il s'applique à des projets en cours ou terminés. Il doit permettre l'évaluation de ces derniers et la capitalisation des savoir-faire liés à ceux-ci.

Il s'applique au projet dans son ensemble et/ou à des phases (faisabilité, conception, …) et il concerne l'ensemble de l'équipe projet.

Application au logiciel

Définitions

L'assurance qualité logiciel donc permet de garantir que :

- Les règles et normes de codage sont respectées
- La documentation liée au code est rédigée au fur et à mesure. Elle contribue à la garantie de pérennité du code et son interopérabilité
- La gestion de versions du code est gérée au fur et à mesure et à travers les outils adaptés

Application au logiciel

Définitions

L'assurance qualité logiciel est donc l'ensemble des activités, normes, contrôles et procédures couvrant la totalité de la durée de vie d'un logiciel, permettant de s'assurer que le logiciel livré est conforme aux exigences du client et est développé suivant les règles du génie logiciel.

Elle répond aux contraintes spécifiques suivantes :

- La durée de vie du matériel sur lequel est installé le logiciel
- Les langages et méthodes qui évoluent
- La complexité du logiciel

Application au logiciel

Méthodologie

→ Souhait d'optimiser le fonctionnement ?

→S'appuyer sur des référentiels

→Rédiger un guide de bonnes pratiques inter projet/équipe

→Disposer d'outils mis à disposition de la communauté

→Utiliser une terminologie commune

→Bénéficier d'un label/ d'une certification (reconnaissance)

Application au logiciel

Méthodologie

Application au logiciel, comment faire ?

1) Définir l'organisation autour du logiciel
2) Maîtrise de la documentation liée au logiciel
3) Amélioration continue et fiabilité du code : gestion des risques

Application au logiciel

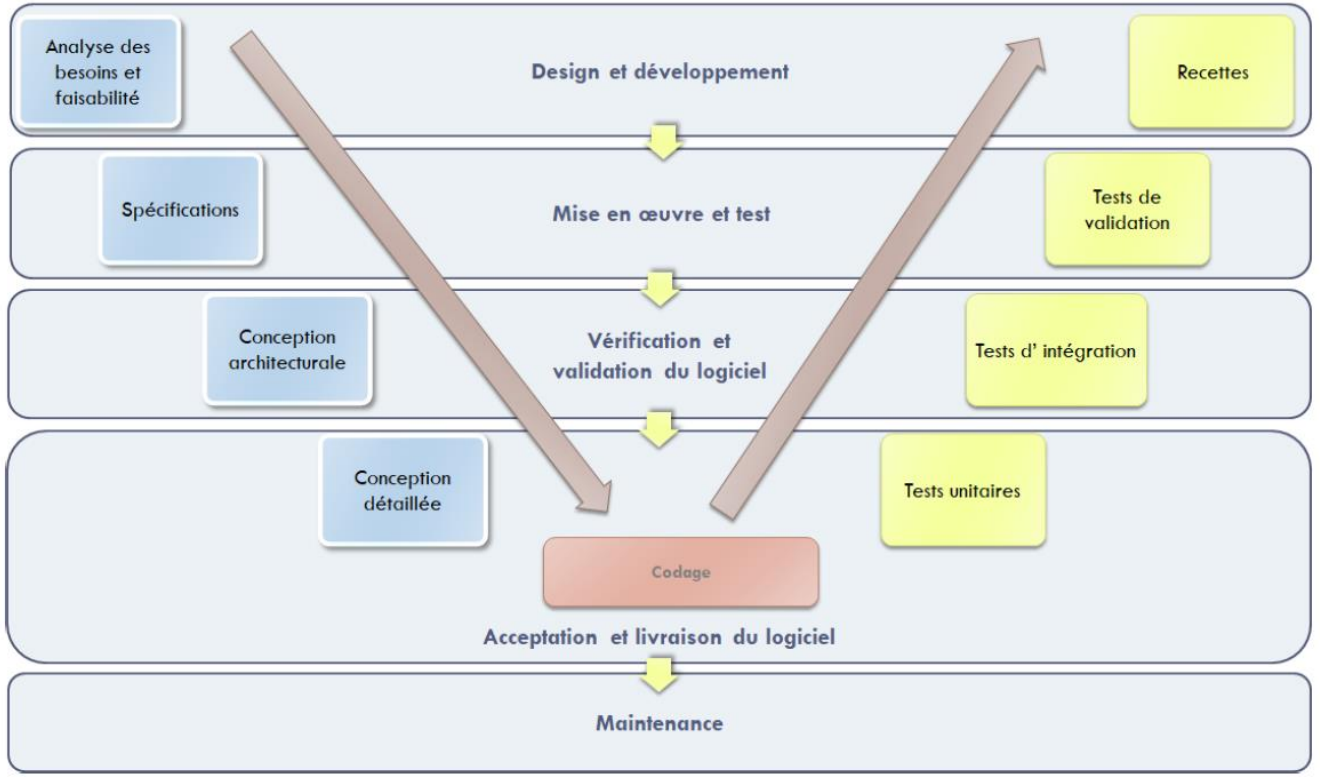
Méthodologie

1) Organisation au sein de l'équipe ou du projet
   - Le cycle de développement du logiciel doit être défini en tenant compte des caractéristiques techniques du logiciel et des contraintes de développement associées comme le choix des techniques employées (système de gestion de bases de données, réutilisation de produits existants, générateurs d'interface homme/machine, etc.) et les risques inhérents au projet.
   - Une analyse de besoins doit être menée avant tout développement logiciel.
   - Dans le cas d'un logiciel intégré dans un équipement, le logiciel participe en tant que constituant à la phase d'intégration matériel-logiciel et son cycle de développement doit être en cohérence avec celui du matériel.

Application au logiciel

Méthodologie

1) Organisation au sein de l'équipe ou du projet
   - Cycle en V

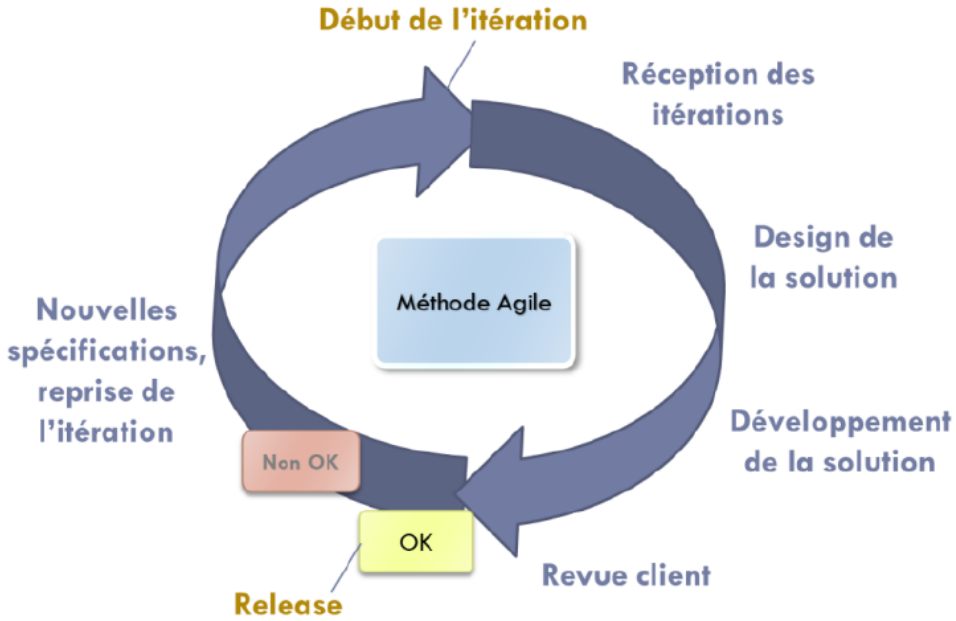


Source: groupe de travail AP QeR, inspirée des "ECSS"

Application au logiciel

Méthodologie

1) Organisation au sein de l'équipe ou du projet
   - L'approche Agile



Source: groupe de travail AP QeR

Application au logiciel

Méthodologie

1) Organisation au sein de l'équipe ou du projet

Basées sur un processus itératif et incrémental, les méthodes Agiles présentent l'avantage de gagner en réactivité et flexibilité car elles prennent en compte les constantes évolutions des besoins.
Le produit logiciel est alors adapté et ajusté sur des temps très courts.

→ Le client est placé au centre de l'organisation

Les méthodes et éventuels outils support adaptés aux différentes phases du cycle de développement du logiciel sont choisis en fonction des compétences de l'équipe projet (plan de formation, expériences antérieures, …), de ses moyens financiers et des contraintes projet/client.

Application au logiciel

Méthodologie

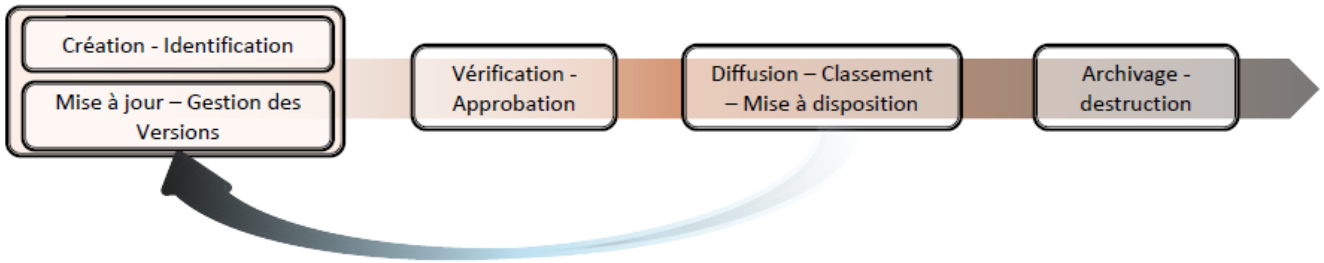
2) Maîtrise documentaire
   - Objectifs :
     - Enregistrer et stocker l'ensemble des données relatives à un projet donné (caractéristiques, activités, résultats, …)
     - Garantir aux acteur la fiabilité, la qualité et la rapidité des renseignements diffusés
     - Favoriser l'accès de tous les acteurs à l'information (documentation à jour)

Application au logiciel

Méthodologie

2) Maîtrise documentaire

- Documenter les éléments clés du système
  - Formaliser les règles de fonctionnement
  - Enregistrer les informations
  - Assurer la traçabilité
- Disposer d'une documentation utile et homogène
  - Documents fiables et conformes aux pratiques utilisateur
  - Présentation standard et codification structurée
- Développer les circuits d'information
  - Diffusion de l'information
  - Sensibilisation du personnel



Application au logiciel

Méthodologie

2) Création du document
   - Utilisation d'un modèle → Uniformité des documents d'un projet
   - Le modèle doit contenir toutes les informations nécessaires à la bonne gestion des documents

A titre d'exemple :

- le titre du document
la référence et la version du document
- la date de la publication et l'historique des révisions
- le nom de l'auteur et si nécessaire le nom du vérificateur et/ou de l'approbateur
- une zone de signature
- La liste des destinataires ou liste de diffusion
- le nombre de pages
- Le sommaire
- La liste des acronymes, si besoin
- La liste des documents applicables et de référence

Identification de manière unique (référence unique (bloc + numéro chrono) , un numéro de version/édition→ une identification définie

Application au logiciel

Méthodologie

2) Exemple d'arborescence documentaire

1. Science
    1. Mission
    2. Exigences scientifiques / Performances
2. Management
    1. Organisation
    2. Planning
    3. Budgets
    4. Contrats/Conventions
    5. Revues
    6. Réunions
3. Instrument / Logiciel
    1. Exigences/Manuels Techniques
    2. Interfaces
    3. Ingénierie/Modélisation
    4. Fabrication (à organiser par étape ou modèle par exemple)
        1. Etape 1 ou Modèle 1
        2. Etape 2 ou Modèle 2
        3. ...
    5. Assemblage/Intégration/Tests
    6. Maintenance
4. Système
    1. Exigences système
    2. Interfaces
    3. Traitements données
    4. Opérations
    5. Exploitation
    6. Archivage données
5. Assurance Produit
    1. Exigences/Manuels Assurance Produit (Plan Assurance Produit,...)
    2. Risques
    3. Composants Electroniques
    4. Matériaux/Procédés/Contamination
    5. Inspections/Contrôles d'entrée
    6. Configuration : Non conformités/Anomalies, Modifications, Evolutions
6. Communication
    1. Logos et Modèles
    2. Articles/Publications/Posters
    3. Rapport de stage
    4. Média et Photos

Application au logiciel

Méthodologie

2) Maîtrise documentaire

- Le caractère modifiable du produit logiciel, qui peut se faire sur antériorité longue, nécessite un suivi très particulier et fait notamment intervenir les notions de gestion de version, tests de régression, commentaires, …
- Il est nécessaire de formaliser les activités comme :
  - L'organisation et la description de ces activités,
  - Les rôles et responsabilités,
  - Les méthodes et outils utilisés,
  - La documentation à produire

- Ces éléments peuvent être décrits dans un chapitre AQL du plan d'assurance produit. Le cas échéant, en fonction de sa criticité, il fera l'objet d'un plan particulier d'assurance qualité logiciel (PAQL).

Application au logiciel

Méthodologie

2) Maîtrise documentaire

- D'autres documents devront être rédigés dans le cadre du développement du logiciel : spécifications techniques du besoin logiciel, plan de développement, document/fichier de configuration logiciel, plan de tests et vérification, manuel utilisateur, …
- La documentation produite doit contenir les informations suivantes : Contexte de développement (hypothèses, limitations), Spécifications, Informations destinées aux utilisateurs (manuel d'utilisation, de référence), Informations nécessaires à la maintenance du produit développé (liste des essais de validation, document de conception, historique des choix, dossier de configuration, …)

- Le tableau ci-après donne une liste non-exhaustive de documents d'ingénierie logiciel sur lesquels vous pouvez vous appuyer :

Application au logiciel

Méthodologie

2) Maîtrise documentaire

INGENIERIE LOGICIEL	REGLES D'UTILISATION DES LANGAGES	ASSURANCE QUALITE LOGICIEL
• Logiciels : Principes et exigences [RNC-ECSS-E-ST-40] • Règles et recommandations pour conduire une recette de produit logiciel [RNC-CNES-E-HB-40-506] • Contenu type de la Spécification Technique de Besoin Logiciel (STBL) [RNC-CNES-E-HB-40-501] • Règles pour l'élaboration de documents de description d'interfaces informatiques [RNC-CNES-E-HB-40-507] • Manuel utilisateur et manuel d'exploitation d'un système informatique sol de traitement [RNC-CNES-E-HB-40-503] • Démarche de développement objet pour les logiciels [RNC-CNES-E-HB-40-508] + Principes de mise en œuvre des concepts objets [508-A] + Documentation [508-B] + Terminologie orientée objet [508-C] + Exemple de développement orienté objet [508-D] • Règles et recommandations d'ergonomie des IHM informatiques [RNC-CNES-E-HB-40-504] • Règles et recommandations pour l'utilisation du formalisme UML [RNC-CNES-E-HB-40-509] • Règles et recommandations pour la réalisation d'un serveur web [RNC-CNES-E-HB-40-505] + Règles pour la réalisation de chartes pour un serveur web [505-A]	• Règles communes pour l'utilisation des langages de programmation [RNC-CNES-Q-ST-80-501] • Règles pour l'utilisation du langage C [RNC-CNES-Q-HB-80-506] + Règles complémentaires pour l'utilisation du langage C dans les logiciels embarqués [506-A] • Règles pour l'utilisation du langage C++ [RNC-CNES-Q-HB-80-513] • Règles pour l'utilisation du langage Java [RNC-CNES-Q-80-527] • Règles pour l'utilisation du langage IDL [RNC-CNES-Q-HB-80-534] • Règles pour l'utilisation du langage Python [RNC-CNES-Q-HB-80-535]	• Assurance produit Logiciel : exigences pour le développement et la maintenance des logiciels [RNC-ECSS-Q-ST-80] • Exigences AQ simplifiées pour le développement logiciel [RNC-CNES-Q-ST-80-100] • Guide : Sélection et interprétation des mesures de complexité Logiciel [RNC-CNES-Q-ST-80-503] • Contenu d'un bilan qualité logiciel [RNC-CNES-Q-ST-80-510] • Contenu d'un bilan de projet logiciel [RNC-CNES-Q-ST-80-511] • Contenu d'un plan de contrôle de la qualité d'un logiciel [RNC-CNES-Q-ST-80-515]

Application au logiciel

Méthodologie

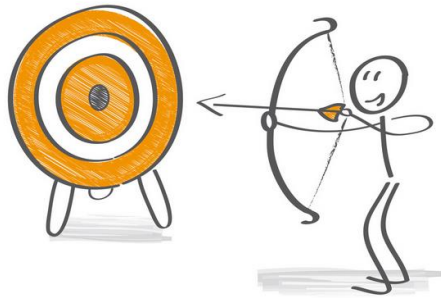
3) Risques à maîtriser

- Mauvaise identification des entrées et sorties
- Matériel non fiable, inadéquat
- Ressources insuffisantes, manque de planification
- Personnel non qualifié ou non formé dans le temps
- Absence de contrôle à cause de l'absence d'indicateurs
- Absence de documentation
- Absence de mise à jour
- Méconnaissance par le personnel
- Non respect des référentiels

Prise en compte des risques et opportunités

- Trouver le bon compromis entre :
  - le besoin réel (attention à la sur-qualité) ,
  - ce qui est spécifié,
  - ce qui est fait
    (attention à la bonne prise en compte du cahier des charges)

- Prendre en compte la « non qualité »
  - Coût impacté (maintenance corrective, maintenance évolutive)
  - Approche pragmatique :
    - Indisponibilité du logiciel/matériel
    - Corrections → mises à jour
    - Retard de livraison
    - Difficulté d'utilisation
    - Peu de pérennité
    - Mise en péril de la sécurité ou valeur juridique

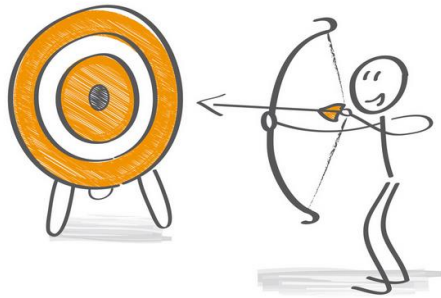


Prise en compte des risques et opportunités

- Selon vous :

Quelles sont les risques projet qui peuvent intervenir ?

Quels sont les facteurs d'occurrence du risque ?

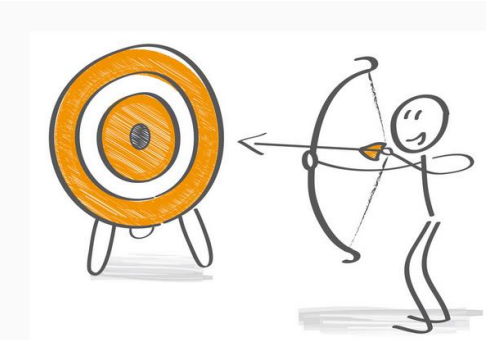


Prise en compte des risques et opportunités

- Etudier la faisabilité
  - Faisabilité environnementale
  - Faisabilité humaine
  - Faisabilité technique

→**Risques = Evènement redouté**

<u>Facteurs d'occurrence :</u> Unicité du projet, complexité, expérience,…



Prise en compte des risques et opportunités

- Matrice de gestion des risques
  - Lister les risques et les identifier
  - Déterminer les actions à venir pour pallier aux risques
  - Analyse des causes et des enjeux

  - Evaluation du risque :
    - Risque majeur ?
    - Probabilité de réalisation du risque
    - Impact si le risque se réalise

    → **P x I**

→Anticiper les risques est donc une nécessité pour mener à bien son projet

Prise en compte des risques et opportunités

- Classification des risques dépend de la politique de gestion des risques adoptée

	Gravité Négligeable	Gravité Majeure	Gravité Elevée
Probabilité Elevée	A surveiller	Inacceptable	Inacceptable
Probabilité Moyenne	Acceptable	Inacceptable	Inacceptable
Probabilité Faible	Acceptable	Acceptable	Acceptable

Tableau 1 : Table de définition des niveaux de criticité

- Mettre en place un suivi des risques (gestion des actions)

Prise en compte des risques et opportunités

- Opportunités :
  - Traçabilité des principaux évènements survenus sur le projet (mise à jour régulière)
  - Suivi de l'avancement des tâches : activités et efficacité de l'équipe
  - Gestion des non-conformités (remplir une fiche de non-conformité : émetteur, date de détection, référence, descriptif de l'incident, dates de réception, intervenants, résolution)
  - Gestion des évolutions/modifications (remplir une fiche d'évolution/modification)
  - Gestion des essais et recettes (suivi des tests, conditions de la recette)

→ Retrouver à tout moment la trace d'une décision prise

→ Permettre le retour à une version stable si problème rencontré

Gestion de la configuration

- La gestion de configuration est le processus permettant d'établir et de conserver un enregistrement cohérent des caractéristiques fonctionnelles et physiques d'un produit par rapport à ces exigences de conception et de fonctionnement.

- La gestion de configuration doit s'appliquer tout au long du cycle de vie du produit (logiciel) et va permettre de :
  - Connaître à tout moment la description technique d'un produit à l'aide d'une documentation approuvée
  - Enregistrer et contrôler l'évolution et la description technique d'un produit
  - Assurer la traçabilité de l'évolution
  - Assurer la cohérence des interfaces internes
  - Vérifier et démontrer que la documentation reste l'image des produits qu'elle décrit
  - Identifier la configuration de référence et la configuration finale d'un produit (enregistrement et contrôle des anomalies, modification/dérogation)

→ **Gérer de manière continue le produit logiciel, son support et son environnement**

Rôles et responsabilités

- Chef de projet
- Développeurs
- Responsable assurance produit (RAP) : s'assure que le logiciel est conforme aux exigences client. Il met en place la méthodologie et les outils nécessaires.
  - Pour cela, il doit veiller à ce que :
    - Les exigences du logiciel soient correctement spécifiées,
    - Des documents de définition formels soient émis,
    - Les normes, pratiques et conventions soient appliquées (par exemple : structure logique, codage, …),
    - Les activités de conception et de développement soient soumises à des revues officielles,
    - Toutes les vérifications et les tests soient effectués suivant des procédures de test rédigées en fonction de la criticité du logiciel,
    - Les procédures de contrôle de la gestion de la configuration soient appliquées.
- Dans certains projets, on parle également de responsable assurance qualité logiciel (RAQL)

Rôles et responsabilités

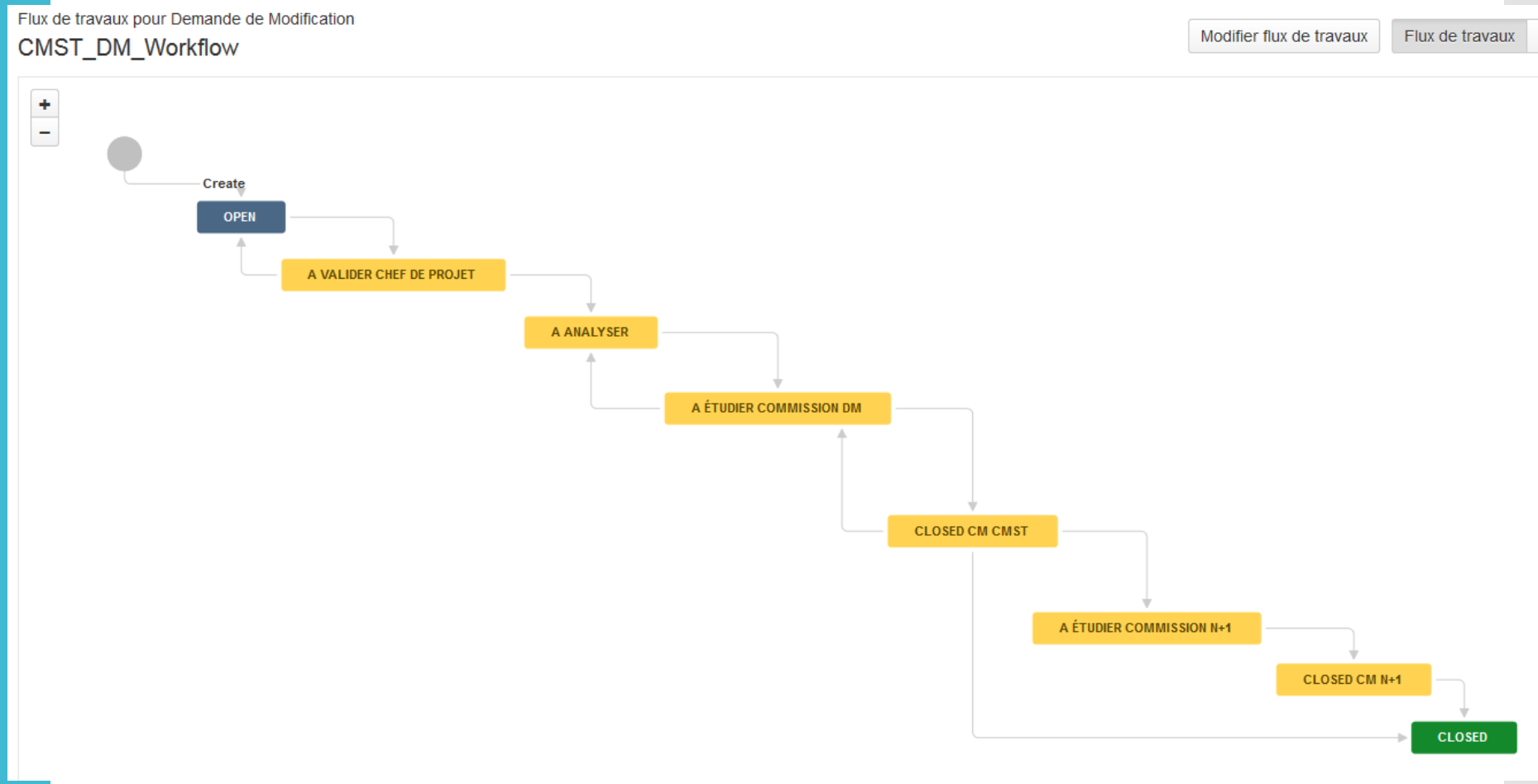
- Responsabilités à définir au niveau de la gestion de documentation, des modifications, de la configuration matérielle, …
  - Organigrammes
  - Responsabilités, autorité, relations mutuelles du personnel chargé de la gestion, exécution et vérification des travaux affectant la qualité logiciel doivent être définies et documentées

→Définir un cycle de vie et un état des composants

→Exemple de « workflow » ci-après *(Outil JIRA)*

Rôles et responsabilités

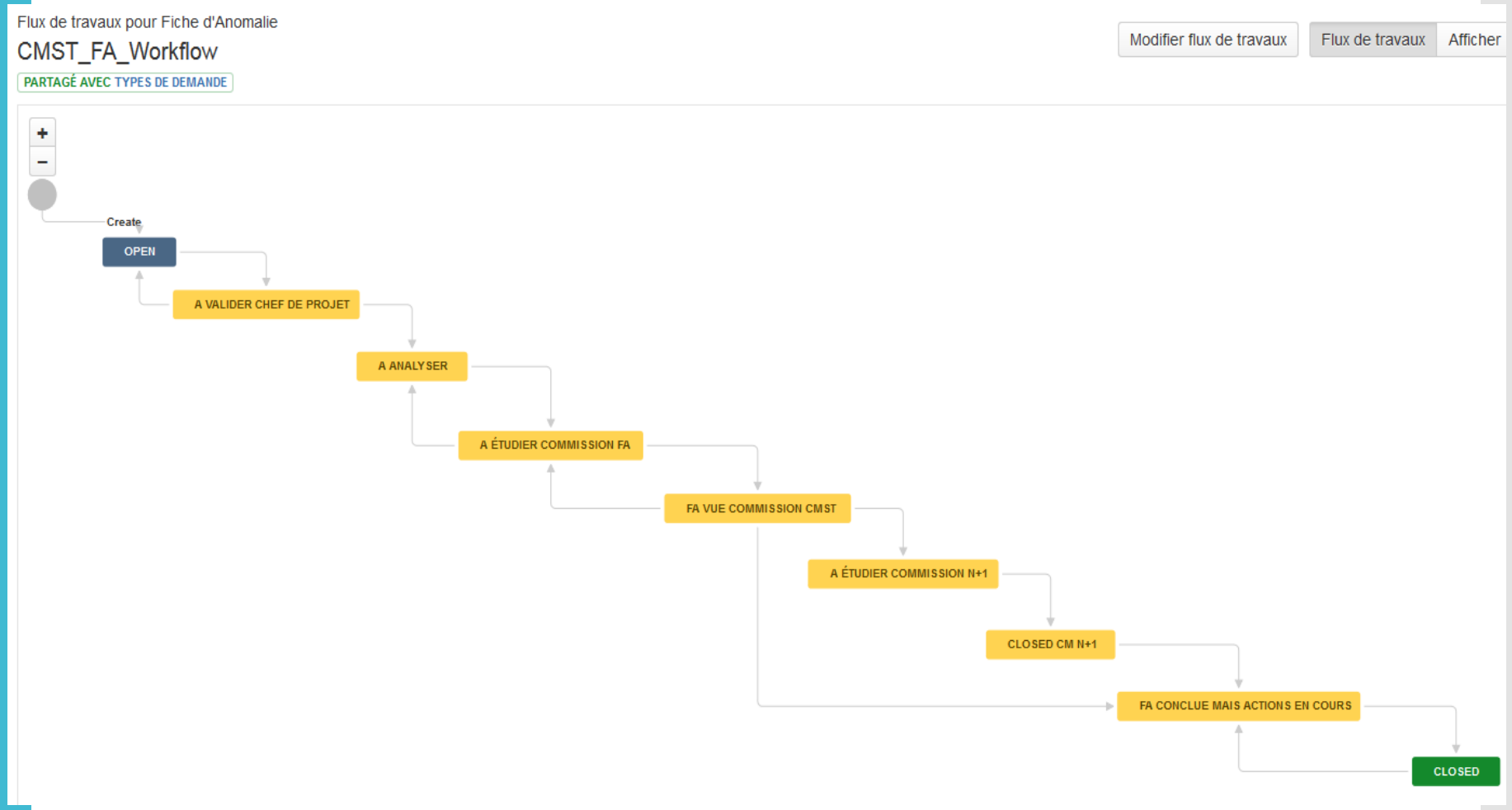
- Gestion des Demandes de Modification (LPC2E_CMST_TARANIS)



Présentation de l'outil JIRA au LPC2E_01/06/2016_C.Agrapart

Rôles et responsabilités

- Gestion des Anomalies (LPC2E_CMST_TARANIS)



Présentation de l'outil JIRA au LPC2E_01/06/2016_C.Agrapart

Conduite de changement

- **Communiquer pour faire adhérer :**
  - Sensibiliser (formation, retour d'expériences, …)
  - Décrire et expliquer les étapes
  - Etre à l'écoute
- **S'appuyer sur des personnel « terrain »**
- **Développer une approche participative :**
  - Rédiger et améliorer les procédures entre les différentes activités
  - Exploiter les suggestions d'amélioration du personnel et traiter les dysfonctionnements
  - Mettre en place des réunions de suivi d'actions
- **Motiver :**
  - Définir des leviers pertinents
  - Enrichir les outils méthodologiques
  - Recueillir et valoriser les bonnes pratiques
- **Evaluer la satisfaction des parties intéressées/partenaires**
- **S'appuyer sur les ressources existantes (réseaux qualité, projet, développeurs et supports pédagogiques liés)**

Ressources / Bibliographie

- Réseau Qualité en Recherche (QeR) : *https://qualite-en-recherche.cnrs.fr/*
  Présentation « Qualité au service de la recherche » (GT_QeR_2015)
  - Journée thématique AQL *https://jt-aql.sciencesconf.org/*
  - Guide Assurance Produit (GT_APAQ_2019) : https://qualite-en-recherche.cnrs.fr/guide/
- « 20190515_plaquette_developpement » (CNRS_Devlog)
- Afnor
  - Norme ISO 9000 / 90001
- https://gns.cnes.fr/fr/qualite-logiciel-ap-7

Merci pour votre attention !

Clémence Agrapart
clemence.agrapart@cnrs-orleans.fr

Lundi 11 octobre