



THESIS / THÈSE

MASTER EN SCIENCES INFORMATIQUES

Déploiement de l'approche MoCQA en environnement professionnel

Hanoteau, Samuel

Award date:
2012

Awarding institution:
Universite de Namur

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Facultés Universitaires Notre-Dame de la Paix, Namur
Faculté d'informatique
Année académique 2011-2012



Déploiement de l'approche MoCQA en environnement professionnel

Samuel Hanoteau

Mémoire présenté en vue de l'obtention du grade de master
en sciences informatiques

Résumé

Sujet

L'approche MoCQA (*Model-Centric Quality Assessment*) définit un cadre générique pour l'évaluation de la qualité de produits logiciels, en se basant essentiellement sur le principe que, pour être efficient, un modèle de qualité doit être réalisé sur mesure en fonction de l'environnement logiciel du projet ou de l'organisme.

Cette approche est actuellement à l'état de développement au sein de la Faculté d'Informatique de Namur.

Le but du présent mémoire est d'appliquer la méthodologie MoCQA au sein du Service Informatique de la *Direction Générale opérationnelle de l'Agriculture, des Ressources Naturelles et de l'Environnement* du Service Public de Wallonie. Ce Service Informatique est en charge d'environ une centaine de produits logiciels : des applications métier mais aussi des logiciels achetés et des composants distribués. A quelques exceptions localisées près, aucune métrique n'est appliquée actuellement sur ces produits.

Les objectifs principaux du présent mémoire sont de déployer l'approche MoCQA ce qui permettra de s'assurer de l'applicabilité de la méthodologie, ainsi que de rédiger un guide de déploiement de l'approche à destination des professionnels de l'informatique.

La principale valeur ajoutée, pour les FUNDP, du présent mémoire est une validation concrète de la méthodologie MoCQA dans un environnement professionnel, à partir d'un environnement quasi vierge de toute métrique, ce qui permettra un retour sur expérience intéressant pour les développeurs de la méthodologie.

Mots clés

MoCQA, Modèle, Qualité, Indicateur, Métrique, Mesure, Produit, Evaluation

Abstract

Topic

The MoCQA (*Model-Centric Quality Assessment*) approach defines a generic framework for the assessment of the quality of software products, based essentially on the principle that, to be efficient, a quality model must be adapted to the software environment where the project or the organization to which it belongs.

This approach is currently being developed at Namur's Faculty of Computer Sciences.

The goal of this Master's thesis is to apply the MoCQA methodology inside the IT Department of the "*Direction Générale opérationnelle de l'Agriculture, des Ressources Naturelles et de l'Environnement*" of the "*Service Public de Wallonie*". This IT Department is in charge of about a hundred software products: mainly business applications but also acquired software packages and distributed components. Except for a few localized cases, no metrics are currently applied on these products.

The main objectives of this Master's thesis consist in deploying the MoCQA approach in order to ensure the applicability of the methodology, as well as writing a deployment guide targeting IT professionals.

The main added value of this Master's thesis for FUNDP resides in a validation of the MoCQA methodology in a professional environment, without metrics, which will bring valuable feedback to the developers of the methodology.

Keywords

MoCQA, Model, Quality, Indicator, Metric, Measurement, Product, Assessment

Avant-propos

Le travail présenté dans ce mémoire n'aurait pu se faire sans le support et le soutien d'autres intervenants. Je tiens donc à remercier ceux qui y ont participé, de près ou de loin.

Merci à *Naji Habra* et *Benoît Vanderose* pour leur encadrement et leurs précieux conseils tout au long de mon travail, y compris lors de la finalisation de ce mémoire. Au-delà du cadre de celui-ci, j'espère continuer la collaboration productive que nous avons eue jusqu'ici.

Merci à *Fabian Dermine*, *Patrick Lebrun* et *Laurent Vanhalle* de m'avoir permis de réaliser ce travail de mémoire dans le cadre de mon environnement professionnel.

Merci à la société *Aubay.Promotic*, plus particulièrement *Michel Soetens*, de m'avoir permis, au travers de mes diverses expériences vécues ces onze dernières années, de trouver ma voie dans les domaines de la gestion de projet et de la qualité, ce qui a indirectement aiguillé le sujet du présent mémoire.

Merci à *Pierre Girboux* d'avoir accepté de participer à la première itération du déploiement, sans savoir avec certitude vers quoi on avançait.

Merci à *Pascal Coupé*, *Yves Konen*, *Jean-François Bastin*, *Michaël Waerenburgh*, *Philippe Vervier* et *Olivier Davreux* pour leurs remarques avisées lors de la révision du présent mémoire.

Merci à *Patrick Heymans* pour son aide lors de ma recherche de documentation.

Merci à *Jean-François Richard*, *Séverine Colin*, *Julien Barbier* et *Olivier Reiland* pour le « petit coup de pouce » dans mes tâches de traduction.

Merci à *Marie Gevers* pour ses conseils en matière de rédaction et de présentation de ce mémoire.

Merci à *Benjamine Lurquin* pour ses conseils pratiques et sa gentillesse au jour le jour.

Merci à *mes parents* qui m'ont encouragé tout au long de mes études et d'avoir fait de moi une grande partie de ce que je suis devenu.

Enfin, sans doute l'hommage qui me tient le plus à cœur, je tiens à remercier affectueusement mon épouse *Valérie* et ma fille *Colline* qui, grâce à leur soutien sans faille et les sacrifices concédés ces dernières années, m'ont permis de vivre l'aventure, à la fois passionnante et éprouvante, que sont les études en horaire décalé. Je vais pouvoir m'occuper pleinement de ma petite famille désormais...

Contenu

Table des matières

Résumé.....	2
Avant-propos.....	4
Contenu	5
Prérequis à la lecture.....	8
Abréviations	9
Introduction.....	10
Chapitre 1 : Les modèles de qualité	12
1.1 Qu'est-ce qu'un modèle de qualité ?	12
1.2 Les modèles de qualité existants	13
1.3 L'approche GQM	19
1.4 L'approche MoCQA	20
Chapitre 2 : L'approche MoCQA.....	21
2.1 Processus général.....	21
2.2 Première phase : Acquisition.....	22
2.3 Seconde phase : Modélisation	23
2.4 Troisième phase : Opérationnalisation	23
2.5 Quatrième phase : Evaluation	24
2.6 Cinquième phase : Exploitation.....	24
Chapitre 3 : L'environnement	26
3.1 Présentation de l'organisme	26
3.2 La démarche d'amélioration qualité	33
3.3 Présentation de l'étudiant.....	36
Chapitre 4 : Le déploiement	37
4.1 Prise de connaissance de l'approche MoCQA.....	37
4.2 Première itération	38
4.3 Itérations suivantes	41
4.4 Communications.....	41
4.5 Suite.....	41
Chapitre 5 : Contributions	42
5.1 Retour d'expérience sur MoCQA.....	42

5.2 Guide de déploiement de l'approche MoCQA	46
5.3 Utilisation de la langue française	46
5.4 Cycle de développement au format MoCQA	47
5.5 Evaluation du coût.....	50
5.6 Réflexion sur la psychologie du déploiement	50
5.7 Réflexion sur la force des indicateurs	54
5.8 Participation à la validation.....	55
Chapitre 6 : Futurs sujets de recherche	57
6.1 Déploiement dans d'autres environnements.....	57
6.2 Utilisation du MoCQA Toolkit.....	57
6.3 Intégration de la norme ISO/IEC 25000 via MoCQA	57
6.4 Indicateurs de qualité sur les « projets »	58
6.5 Automatisation et présentation des indicateurs	58
6.6 Mise en place d'une activité de contrôle qualité informatique.....	58
6.7 Indicateurs identifiant un besoin de ressources humaines	59
6.8 Diffusion du guide de déploiement.....	59
Conclusion	60
Références.....	61
Bibliographie	61
Webographie.....	62
Références internes DGARNE.....	62
Annexe 1 : Guide de déploiement.....	63
Annexe 2 : Livrables du déploiement.....	99
A2.1 Objectifs de qualité	99
A2.2 Modèles MoCQA	100
A2.3 Plans de mesure et règles d'interprétation	106
A2.4 Résultats d'évaluation.....	119
A2.5 Plans d'actions	120
Annexe 3 : Coûts du projet.....	121
Annexe 4 : Comptes rendus de réunions	122

Table des illustrations

Figure 1 : Modèle de qualité de McCall	13
Figure 2 : Modèle de qualité de Boehm	14
Figure 3 : Modèle de qualité de Dromey.....	15
Figure 4 : Modèle de qualité FURPS.....	16
Figure 5 : Modèle de qualité ISO 9126 - Internal and External Quality	17
Figure 6 : Modèle de qualité ISO 9126 - Quality in use.....	17
Figure 7 : Modèle de qualité SQuaRE - Software product quality	18
Figure 8 : Modèle de qualité SQuaRE - Quality in use.....	18
Figure 9 : Processus général de gestion d'un indicateur de qualité.....	21
Figure 10 : Rappel allégé de la notation BPMN.....	22
Figure 11 : Représentation de la phase d'acquisition	22
Figure 12 : Représentation de la phase de modélisation.....	23
Figure 13 : Représentation de la phase d'opérationnalisation	23
Figure 14 : Représentation de la phase d'évaluation.....	24
Figure 15 : Représentation de la phase d'exploitation	24
Figure 16: Organigramme du SPW.....	26
Figure 17: Organigramme de la DGARNE.....	27
Figure 18: Organigramme de la D4	28
Figure 19: Organigramme de la D44	29
Figure 20: Organigramme de la D443	30
Figure 21: Organigramme du PAD.....	31
Figure 22 : Structure de l'objectif de qualité "Fiabilité"	39
Figure 23 : Modèle MoCQA "Survenance des incidents"	39
Figure 24 : Modélisation MoCQA d'une collection d'incidents.....	44
Figure 25 : Modélisation MoCQA de la mesure du nombre d'incidents (solution 1).....	44
Figure 26 : Modélisation MoCQA de la mesure du nombre d'incidents (solution 2).....	45
Figure 27 : Modèle du domaine des produits SPI	48
Figure 28 : Modèle MoCQA des entités mesurables liées au Produit.....	49
Figure 29 : Modèle MoCQA des entités mesurables liées au Produit développé.....	49
Figure 30 : Modèle MoCQA des entités mesurables liées à l'Application métier.....	50
Figure 31 : Extrait des tableaux d'interprétation "Fiabilité"	51

Prérequis à la lecture

La lecture du présent mémoire est accessible à toute personne ayant quelques connaissances dans le domaine du développement de logiciels, à savoir au moins connaître les grandes étapes du cycle de développement.

La connaissance du formalisme des diagrammes de classes et des diagrammes d'objets UML permettra d'appréhender plus facilement les modèles présentés mais n'est pas essentielle à la compréhension globale du sujet.

De même, certains processus sont présentés au format BPMN mais la connaissance de la notation n'est pas obligatoire car la compréhension de la notation est généralement reconnue comme assez intuitive.

Enfin, l'objectif du mémoire n'étant pas de décrire l'approche MoCQA, et bien qu'un résumé de celle-ci soit donné au chapitre 2, la compréhension des concepts suppose une connaissance, au moins sommaire, de l'approche (cette connaissance pouvant être acquise à la lecture de l'annexe 1).

Abréviations

Abréviation	Signification
ALM	Application Lifecycle Management
AQI	Assurance Qualité Informatique
BPMN	Business Process Model and Notation
DGARNE	Direction Générale de l'Agriculture, des Ressources Naturelles et de l'Environnement
FUNDP	Facultés Universitaires Notre-Dame de la Paix
GQM	Goal Question Metric
IT	Information Technology
J.H	Jour.Homme
MoCQA	Model-Centric Quality Assessment
PAD	Pôle d'Appui au Développement
PL/SQL	Procedural Language / Structured Query Language
PMBOK	Project Management Body Of Knowledge
SGIQ	Système de Gestion des Indicateurs Qualité
SOA	Service Oriented Architecture
SPI	Service de Production Informatique
SPW	Service Public de Wallonie
SSII	Service de Support Informatique et Informatique
UML	Unified Modeling Language
XOCQAM	XML Operational Customized Quality Assessment Model

Introduction

L'approche MoCQA propose un cadre générique pour la mise en place de modèles de qualité produisant des indicateurs de qualité sur les produits logiciels. L'approche est actuellement à l'état de développement au sein des FUNDP et se compose de trois éléments principaux :

- le méta-modèle MoCQA qui, lorsqu'il est instancié, permet la création des modèles ;
- la méthodologie MoCQA qui décrit le processus de définition, d'évaluation et d'exploitation des indicateurs ;
- les outils MoCQA, à savoir un format de fichier XOCQAM, le plug-in Eclipse MoCQA Toolkit ainsi qu'un logiciel d'édition multiplateformes permettant d'instancier le méta-modèle et de définir le plan de mesure des indicateurs.

A noter que les outils MoCQA n'ont pas été utilisés dans le présent déploiement car pas assez aboutis et donc non exploitables dans les délais de réalisation du sujet de mémoire.

Jusque maintenant, l'approche MoCQA a été testée avec des projets d'étudiants mais pas encore hors du cadre académique. Le sujet du présent mémoire vise à déployer l'approche dans un environnement professionnel, ceci afin d'apporter une aide à la validation de celle-ci.

Le sujet du présent mémoire vise trois objectifs :

- une utilisation concrète de l'approche dans un environnement professionnel, ce qui permettra de s'assurer de la faisabilité de celle-ci, de donner un retour d'expérience aux FUNDP et d'éventuellement suggérer des adaptations ;
- la rédaction d'un guide de déploiement MoCQA destiné à toute entreprise désireuse de se lancer dans un tel déploiement : une sorte de « vulgarisation » des concepts universitaires parfois complexes et inhabituels à appréhender lorsqu'ils sont proposés sous forme d'articles scientifiques ;
- l'identification des possibilités d'extension et/ou de recherches concernant l'approche MoCQA.

La suite du présent mémoire est structurée comme suit :

- le *chapitre 1* reprend un état de l'art des modèles de qualité existants les plus connus et introduit l'approche MoCQA.
- Le *chapitre 2* décrit l'approche MoCQA dans les grandes lignes : le méta-modèle et les cinq étapes principales de la méthodologie.
- Le *chapitre 3* donne une description de l'environnement dans lequel le déploiement a été réalisé et donne un aperçu de l'historique ayant amené le projet de déploiement.
- Le *chapitre 4* explique le déploiement en lui-même : la prise de connaissance de l'approche MoCQA, les difficultés rencontrées, les résultats des premières itérations et constitue un premier retour d'expérience.
- Le *chapitre 5* décrit les contributions du présent mémoire à l'approche MoCQA : les travaux réalisés (notamment le guide de déploiement), les études menées, etc.

- Le *chapitre 6* décrit quelques sujets de réflexion à mener à l'avenir, au sein de l'environnement de déploiement ou au sein des FUNDP, qui n'ont pas été traités dans le présent mémoire.
- Enfin, une conclusion rapide reprend, pour chaque objectif, le degré d'atteinte de celui-ci ainsi qu'une vision critique et constructive sur les travaux effectués dans le cadre du mémoire.

Quelques annexes viennent compléter le présent document :

- L'annexe 1, sans doute la plus importante, est le *guide de déploiement MoCQA* lui-même.
- L'annexe 2 reprend les différents éléments produits lors des itérations du déploiement : les indicateurs, les modèles de qualité, les plans de mesure, les résultats d'évaluation et les plans d'actions liés à l'exploitation de ceux-ci.
- L'annexe 3 donne un aperçu des coûts du projet.
- L'annexe 4 reprend l'ensemble des comptes rendus de réunion ayant eu lieu entre l'étudiant et ses (co-)promoteurs.

Chapitre 1 : Les modèles de qualité

Bien que le sujet du présent mémoire concerne l'approche MoCQA (décrite dans le chapitre suivant), il est intéressant de rappeler en quoi consiste un modèle de qualité logicielle et de faire un état des lieux sur les modèles de qualité existants les plus connus. Enfin, une brève description des principaux avantages de l'approche MoCQA et des modèles de qualité « classiques » est présentée.

1.1 Qu'est-ce qu'un modèle de qualité ?

La norme ISO/IEC 9126 [ISO/IEC 2001] définit un modèle de qualité comme étant « the set of characteristics and the relationships between them which provide the basis for specifying quality requirements and evaluating quality », ce qu'on pourrait traduire comme étant « l'ensemble des caractéristiques, et des relations entre elles, fournissant les bases pour la spécification des exigences de qualité et l'évaluation de la qualité ».

Un modèle de qualité logicielle fournit donc des spécifications permettant au développeur¹ d'appréhender le niveau de qualité exigé des livrables qu'il produit et à l'évaluateur de la qualité de disposer d'un support dans son travail d'évaluation.

¹ On comprendra ici par *développeur*, la personne, l'ensemble de personnes ou la société réalisant tout ou une partie de la production logicielle.

1.2 Les modèles de qualité existants

1.2.1 Modèle de qualité de McCall

En 1977, Jim McCall propose un modèle dans le but de réduire le fossé entre les utilisateurs et les développeurs informatiques. Le modèle de McCall identifie 3 grandes perspectives définissant la qualité d'un produit logiciel, chacune étant composée de facteurs de qualité.

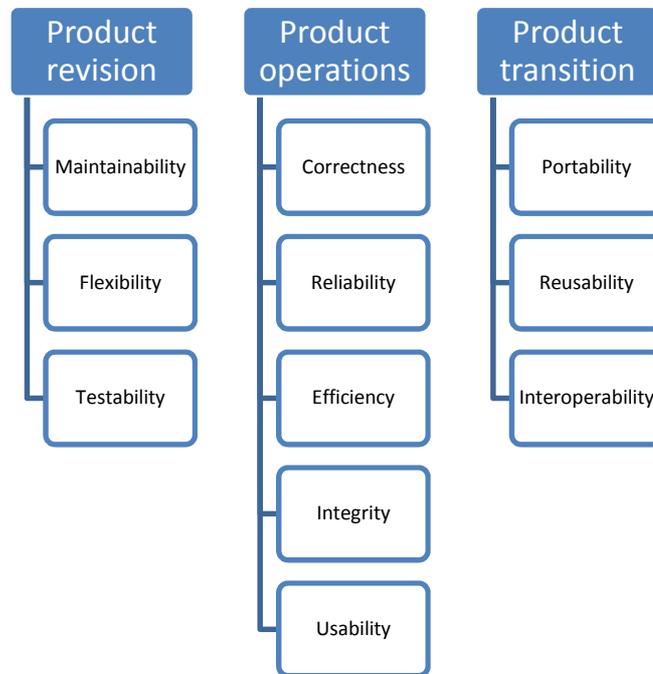


Figure 1 : Modèle de qualité de McCall

Le principal apport du modèle de McCall est la relation entre les caractéristiques de qualité et les métriques, même si l'objectivité des métriques peut être critiquée. Ce modèle ne tient pas compte de l'aspect fonctionnel du produit logiciel. [Ortega, Pérez& Rojas 2003]

Une autre faiblesse de ce modèle est qu'il n'envisage le problème que du point de vue du code source d'un produit.

1.2.2 Modèle de qualité de Boehm

En 1978, Barry Boehm introduit son modèle de qualité afin d'évaluer quantitativement et automatiquement la qualité d'un produit logiciel. Le modèle de Boehm définit des caractéristiques de haut-niveau, certaines étant définies par des caractéristiques de niveau intermédiaire.

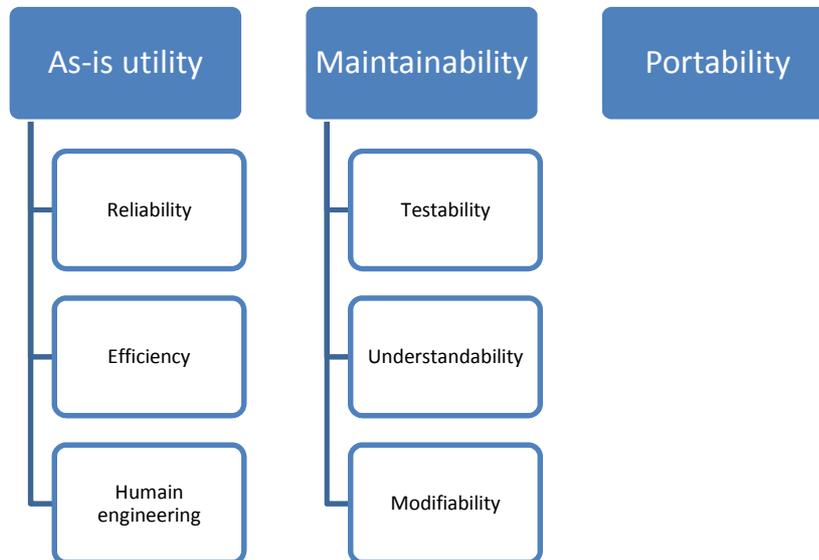


Figure 2 : Modèle de qualité de Boehm

Le modèle de Boehm est similaire au modèle de McCall dans sa représentation hiérarchique des caractéristiques, chacune contribuant à la qualité globale. Boehm ajoute les caractéristiques matérielles non rencontrées dans le modèle de McCall. [Ortega, Pérez& Rojas 2003]

1.2.3 Modèle de qualité de Dromey

En 1996, R. Geoff Dromey présente un nouveau modèle de qualité basé sur le fait que les critères de qualité peuvent différer d'un produit logiciel à un autre. Le modèle de Dromey définit 4 propriétés pour un produit logiciel, chaque propriété étant atteinte par des attributs de qualité.

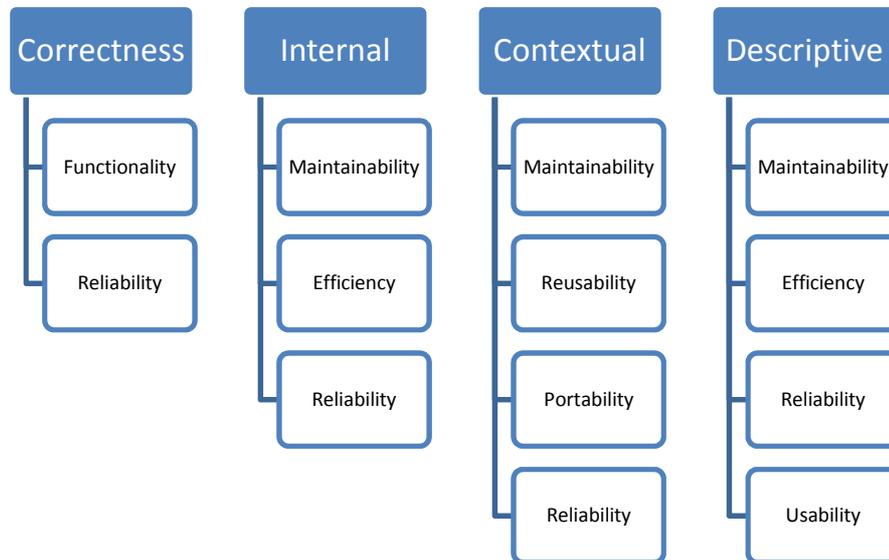


Figure 3 : Modèle de qualité de Dromey

Le modèle de Dromey cherche à augmenter la compréhension de la relation entre les attributs et les sous-attributs de qualité. Il tente également d'identifier les propriétés du produit logiciel qui affectent les attributs de qualité. [Ortega, Pérez& Rojas 2003]

Il s'agit donc d'un premier pas vers une modélisation orientée produit.

1.2.4 Modèle de qualité FURPS

En 1987, Robert Grady et Deborah Caswell présentent le modèle FURPS qui s'appuie sur 5 caractéristiques principales incluant des sous-caractéristiques. Une version étendue, le modèle FURPS+, fut proposée en 2000 par la société IBM.

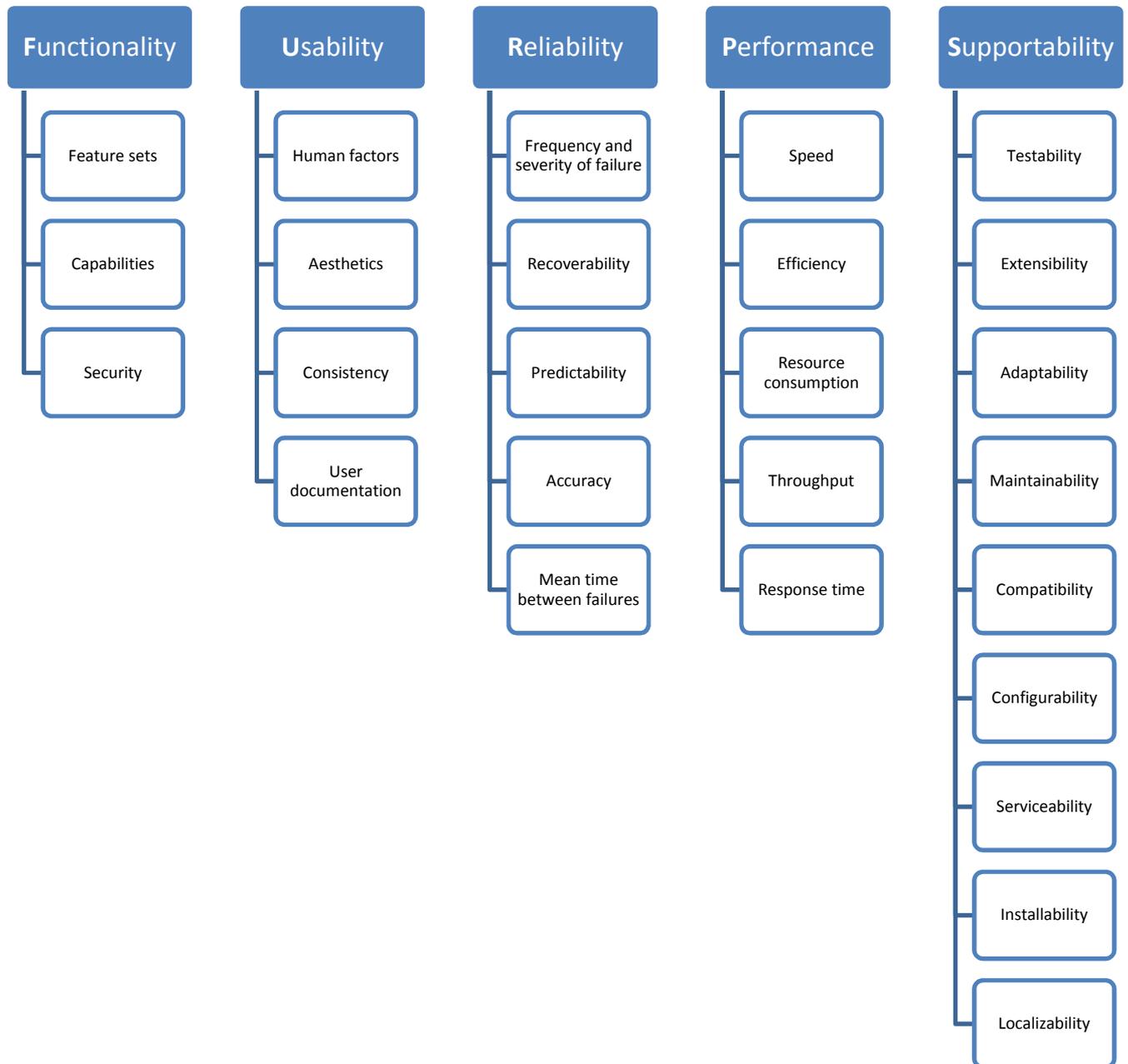


Figure 4 : Modèle de qualité FURPS

Le modèle FURPS décompose les caractéristiques de qualité en deux catégories : l'aspect fonctionnel et l'aspect non-fonctionnel. Un inconvénient du modèle FURPS est qu'il ne tient pas compte de la portabilité du produit logiciel. [Ortega, Pérez & Rojas 2003]

1.2.5 Modèle de qualité ISO 9126

En 1991, l'organisme de normalisation ISO publie ISO 9126, amélioré entre 2001 et 2004 en une série ISO 9126 divisée en quatre documents et proposant des métriques pour le modèle de qualité. ISO 9126 est décomposé en deux parties : un modèle de la qualité interne/externe et un modèle de la qualité à l'utilisation.

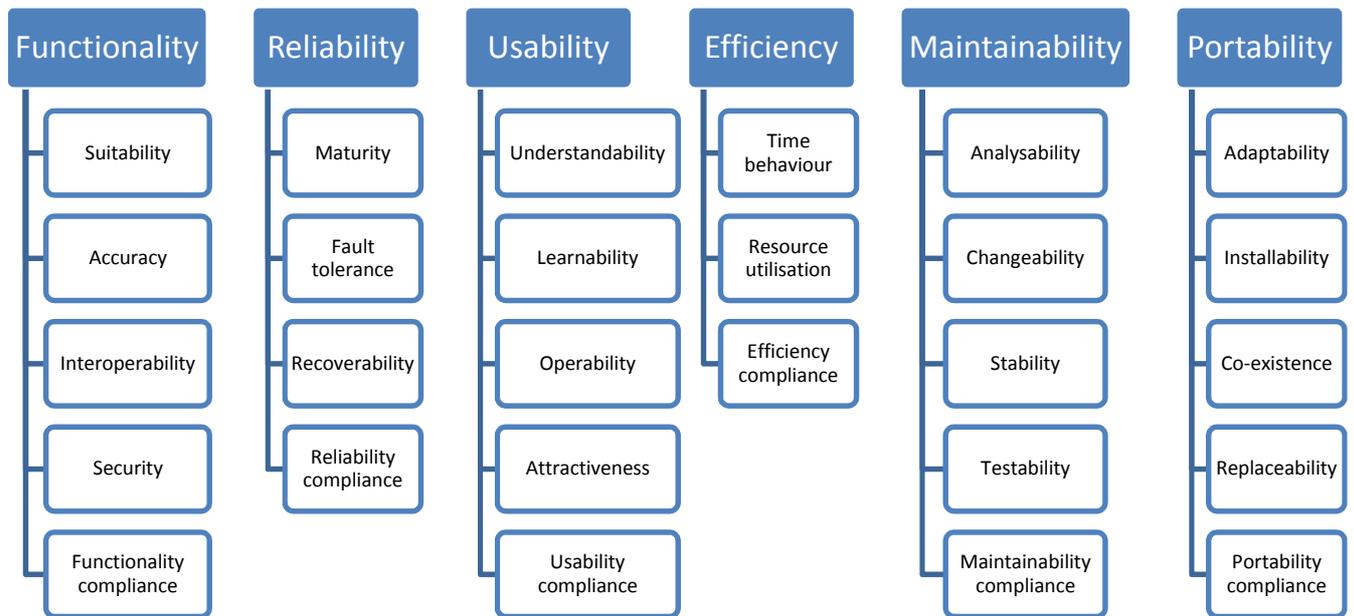


Figure 5 : Modèle de qualité ISO 9126 - Internal and External Quality

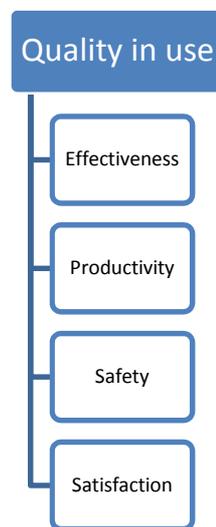


Figure 6 : Modèle de qualité ISO 9126 - Quality in use

Un des principaux avantages de ce modèle est qu'il identifie les caractéristiques de qualité interne et externe du produit logiciel. Toutefois, il a l'inconvénient de ne pas faire apparaître clairement comment ces aspects peuvent être mesurés. [Ortega, Pérez& Rojas 2003]

1.2.6 Modèle de qualité SQuaRE (ISO 250xx)

En 2005, l'organisme de normalisation ISO propose un regroupement logique des normes ISO 9126 (spécification de la qualité des produits logiciels) et ISO 14598 (évaluation de la qualité des produits logiciels) en une seule série appelée SQuaRE (Software product Quality Requirements and Evaluation). Comme ISO 9126, la sous-série ISO 2501x propose deux parties : un modèle de la qualité du produit logiciel et un modèle de la qualité à l'utilisation.

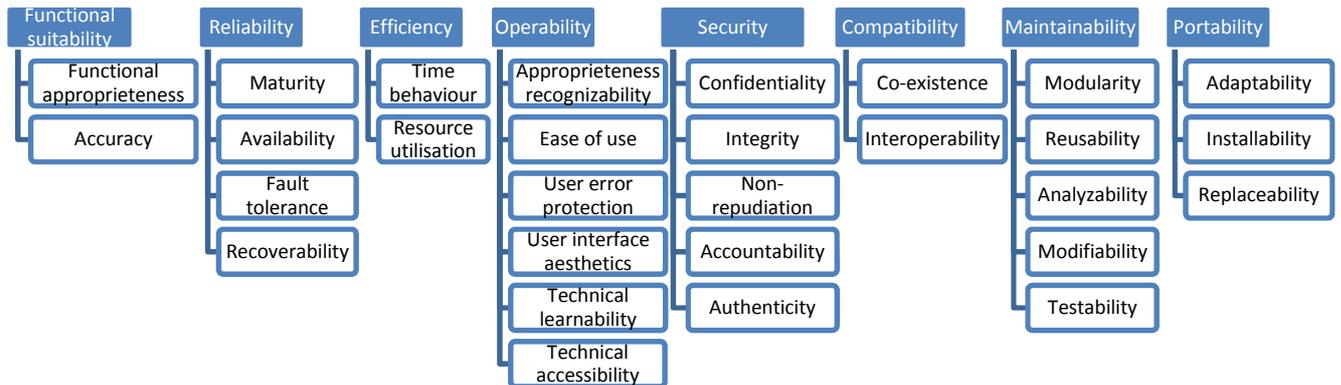


Figure 7 : Modèle de qualité SQuaRE - Software product quality

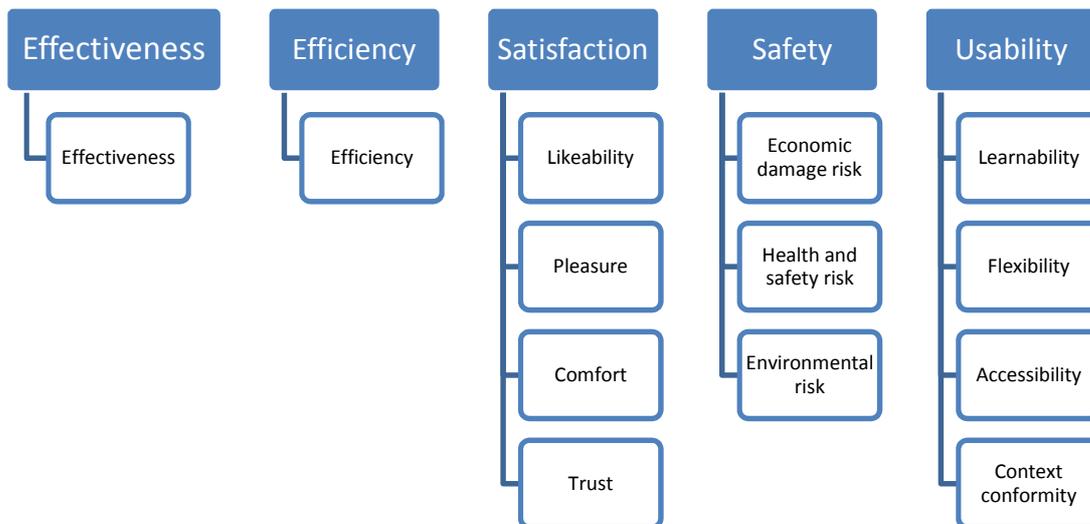


Figure 8 : Modèle de qualité SQuaRE - Quality in use

Comparé au modèle ISO 9126, le modèle de qualité SQuaRE amène deux nouvelles caractéristiques, à savoir la compatibilité et la sécurité, et structure la qualité à l'utilisation de manière plus complète.

1.2.7 Autres modèles de qualité

Il existe d'autres modèles de qualité plus spécifiques : l'objectif du présent mémoire n'étant pas d'en faire une liste exhaustive, les plus connus ont été résumés ci-dessus.

Tous ces modèles ne sont en fait que des divisions hiérarchiques du concept de qualité. Ce ne sont donc pas de véritables modèles, dans le sens où ils ne s'adaptent pas (ou peu) à l'environnement pour refléter celui-ci.

1.3 L'approche GQM

L'approche GQM (Goal Question Metric) fut proposée par Victor R. Basili en 1992 [Basili, Caldiera & Rombach 1994]. GQM part du principe qu'un modèle de mesures doit être défini avec une approche *top-down* en trois niveaux :

- GOAL (niveau conceptuel) : l'objectif de qualité à atteindre portant sur un produit, un processus ou une ressource ;
- QUESTION (niveau opérationnel) : un ensemble de questions visant à évaluer l'atteinte d'un objectif de qualité ;
- METRIC (niveau quantitatif) : un ensemble de données mesurables (objectives ou subjectives) permettant de répondre à chaque question.

Contrairement aux modèles présentés ci-avant, l'approche GQM ne propose pas un ensemble de caractéristiques prédéterminées mais vise à les définir en fonction du contexte de l'évaluation. Il est donc à noter que GQM ne propose pas un modèle de qualité en soi mais plutôt une méthodologie visant à fournir un modèle de l'évaluation de la qualité.

1.4 L'approche MoCQA

L'approche MoCQA (Model-Centric Quality Assessment) est le fruit d'un projet de recherche au sein des Facultés Universitaires Notre-Dame de la Paix de Namur [Vanderose, Habra & Kamseu 2011]. Cette approche reprend les principes de base de l'approche GQM en proposant un méta-modèle permettant d'instancier les modèles de qualité et un cycle de vie de l'évaluation de la qualité.

Comme le présente le chapitre suivant, ce cycle de vie reprend, dans ses étapes 3 à 5, les principes proposés par les modèles d'évaluation prédéfinis comme ISO 9126, à savoir définir comment mesurer les attributs, réaliser l'évaluation et exploiter les résultats en vue de définir des actions visant à améliorer la qualité.

Le réel apport de MoCQA est ses deux premières étapes, à savoir la définition des objectifs qui permettent de déterminer quels modèles de qualité créer et la modélisation qui permet, grâce au méta-modèle MoCQA, de représenter ces modèles de qualité. Ce méta-modèle ne s'attache pas à une approche en particulier mais cherche à dégager le méta-modèle implicite des modèles de qualité présentés précédemment. De plus, comme GQM, le méta-modèle MoCQA ne fournit pas une décomposition en facteurs de qualité mais un moyen de modéliser tous les aspects clés de l'évaluation de la qualité du logiciel. L'étape 5 d'exploitation peut donc identifier également le besoin de remettre en cause un modèle MoCQA ou d'identifier un besoin de nouveaux indicateurs de qualité.

Le principal avantage de l'approche MoCQA par rapport à des modèles de qualité prédéfinis est que le modèle de qualité s'adapte à l'environnement de l'organisme où l'approche est déployée. L'autre avantage important est qu'on ne se « cloisonne » pas à un modèle particulier mais à une approche plus large permettant notamment, si on le souhaite, d'y inclure un modèle (ou une partie de modèle) existant.

A l'inverse, l'avantage des modèles de qualité prédéfinis est qu'on dispose, dès le début du travail, de caractéristiques clairement définies, voire d'éléments directement mesurables, ce qui peut être un gain de temps considérable en début de déploiement, mais peut très vite devenir coûteux à moyen terme lorsqu'on décide de façonner le modèle à un domaine particulier.

La comparaison s'arrêtera ici, l'objectif du présent mémoire n'étant pas d'évaluer l'approche MoCQA par rapport aux modèles de qualité existants, mais d'évaluer notamment la faisabilité de son déploiement en environnement professionnel.

Chapitre 2 : L'approche MoCQA

Comme nous l'avons vu dans l'introduction et le chapitre 1.4 ci-avant, l'approche MoCQA propose un cadre générique pour la mise en place de modèles de qualité produisant des indicateurs de qualité sur les produits logiciels, son réel apport étant ses deux premières étapes, à savoir la définition des objectifs qui permettent de déterminer quels modèles de qualité créer et la modélisation qui permet, grâce au méta-modèle MoCQA, de représenter ces modèles de qualité.

L'approche MoCQA consiste en un méta-modèle, une méthodologie et un kit d'outils en cours de développement.

Comme expliqué dans l'introduction ci-avant, le kit d'outils MoCQA n'a pas pu être exploité dans le cadre du présent mémoire.

L'objectif de ce chapitre n'est pas d'aborder la méthodologie MoCQA en profondeur : le lecteur intéressé pourra se référer aux publications de Naji Habra, Benoît Vanderose et Flora Kamseu [Vanderose 2010], [Vanderose, Kamseu & Habra 2010], [Vanderose, Habra & Kamseu 2011], ainsi qu'au *guide de déploiement de l'approche MoCQA en environnement professionnel* [Hanoteau 2012] proposé dans l'annexe 1 du présent mémoire.

Néanmoins, il est intéressant de reprendre les grandes lignes du processus à suivre pour créer et gérer des indicateurs de qualité via l'approche MoCQA.

2.1 Processus général

Comme le montre la figure ci-dessous, le processus général de déploiement d'indicateurs de qualité peut être structuré en 5 phases.

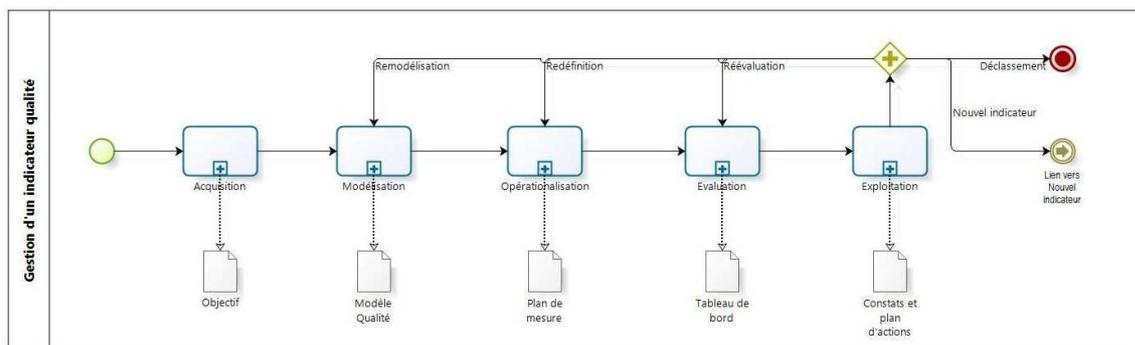


Figure 9 : Processus général de gestion d'un indicateur de qualité

Ci-dessous, un bref rappel des concepts de la notation BPMN utilisés dans ce processus :

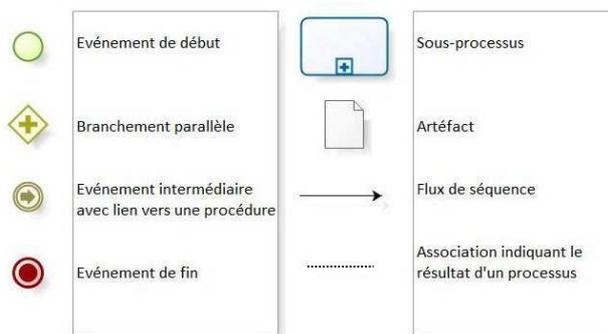


Figure 10 : Rappel allégé de la notation BPMN

2.2 Première phase : Acquisition

La première phase consiste à identifier les objectifs de qualité à atteindre pour les produits logiciels² à évaluer.

On commence par identifier les intervenants ayant un intérêt dans la qualité des produits logiciels. Ensuite, on récolte les besoins en termes de qualité de ces intervenants, généralement via une interview de ceux-ci.

A la fin de cette phase d'acquisition, on a ainsi dégagé les principaux objectifs, éventuellement déjà structurés, comme l'indique la figure ci-dessous. Lors de la récolte des besoins auprès des intervenants, on en profite pour identifier tout ou partie des entités mesurables et des métriques éventuellement déjà existantes.



Figure 11 : Représentation de la phase d'acquisition

² A noter que le processus peut également porter sur un seul produit logiciel.

2.3 Seconde phase : Modélisation

Cette seconde phase vise à instancier le méta-modèle MoCQA pour créer un modèle MoCQA³, à savoir un modèle personnalisé d'évaluation de la qualité relatif aux objectifs identifiés.

C'est ici qu'on déterminera :

- les indicateurs de qualité permettant d'évaluer l'atteinte des objectifs,
- les différentes entités mesurables sur lesquelles vont porter les mesures,
- les différents attributs qui devront être capturés ou calculés et,
- les règles d'interprétation des indicateurs de qualité.

A la fin de cette phase de modélisation, on dispose d'un modèle MoCQA formalisant ce qu'on doit mesurer (principalement le « quoi »), comme l'indique la figure ci-dessous.

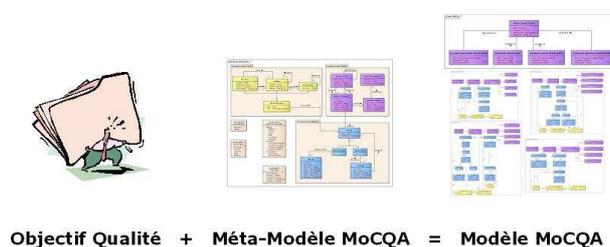


Figure 12 : Représentation de la phase de modélisation

2.4 Troisième phase : Opérationnalisation

Après avoir déterminé ce qu'on allait mesurer lors de la phase 2, la troisième phase vise à déterminer comment le mesurer.

Il s'agit donc ici de déterminer, pour chaque attribut, la façon d'en faire l'acquisition (interrogation d'une ressource humaine ou matérielle, souvent une base de données) et/ou de le calculer (manuellement ou avec automatisation) au sein de l'environnement où se déroule l'opérationnalisation.

A la fin de cette phase d'opérationnalisation, on dispose d'un plan de mesure indiquant comment mesurer (le « comment »), comme l'indique la figure ci-dessous.

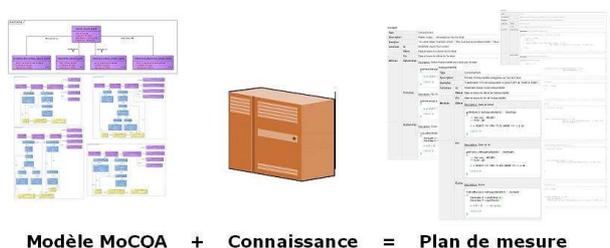


Figure 13 : Représentation de la phase d'opérationnalisation

³ Concrètement, un modèle MoCQA peut être représenté via plusieurs schémas différents, chaque schéma portant sur un « sous-modèle » particulier.

2.5 Quatrième phase : Evaluation

Une fois que l'on a déterminé ce qu'on allait mesurer (phase 2) et comment le mesurer (phase 3), on peut réaliser l'évaluation elle-même.

Il s'agit ici d'appliquer le plan de mesure sur base du modèle MoCQA afin d'acquérir et de calculer les métriques adéquates, ainsi que d'évaluer les indicateurs de qualité.

A la fin de cette phase, on dispose alors d'un tableau reprenant, pour chaque produit logiciel évalué, les valeurs liées à chaque indicateur de qualité, comme l'indique la figure ci-dessous.

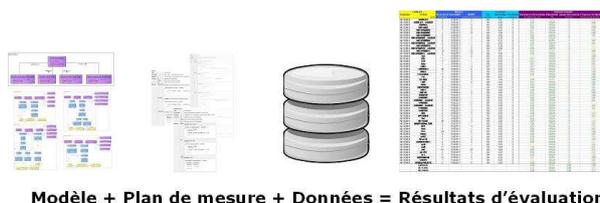


Figure 14 : Représentation de la phase d'évaluation

2.6 Cinquième phase : Exploitation

La dernière phase du processus de gestion des indicateurs de qualité vise à exploiter les résultats issus de la phase d'évaluation.

C'est ici qu'on étudiera les résultats de l'évaluation, ce qui permettra de faire des constatations :

- sur le modèle MoCQA issu de la phase de modélisation et éventuellement identifier des changements à y apporter,
- sur le plan de mesure issu de la phase d'opérationnalisation et éventuellement identifier des changements à y apporter,
- sur les objectifs de qualité à atteindre, ce qui peut donner lieu à un déclassement de certains indicateurs jugés inutiles et/ou à l'identification de nouveaux objectifs à modéliser,
- sur la qualité des produits logiciels évalués et éventuellement identifier des actions correctives à apporter sur les produits.

L'ensemble de ces constatations se transforme ainsi en un plan d'actions à réaliser, comme le montre la figure ci-dessous, ceci dans un souci d'amélioration continue.

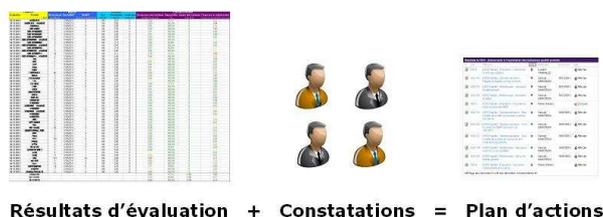


Figure 15 : Représentation de la phase d'exploitation

C'est généralement ici qu'on comprend mieux l'aspect itératif de l'approche MoCQA, l'ensemble des phases pouvant être remises en question.

A noter l'existence d'un « scénario idéal » dans le cas où aucune action d'amélioration n'est identifiée : dans ce cas, on bouclera de manière itérative sur les deux dernières phases afin d'évaluer régulièrement la qualité des produits logiciels concernés et ainsi de s'assurer de la non régression des indicateurs.

Chapitre 3 : L'environnement

3.1 Présentation de l'organisme

Le déploiement dont il est question ici a été réalisé au sein du Service de Production informatique D443 de la Direction générale opérationnelle de l'Agriculture, des Ressources naturelles et de l'Environnement, faisant partie du Service public de Wallonie. Etant donné la complexité de la structure, il semble opportun d'en dire plus sur le sujet.

L'objectif n'est pas de proposer une vue exhaustive des compétences du Service public de Wallonie, mais de situer dans quel contexte le sujet du présent mémoire est déployé. Le lecteur aura donc conscience que les entités administratives présentées ne le sont pas parce que plus importantes que les autres, mais bien parce que faisant partie de ce contexte.

Pour faciliter la lecture et la compréhension, les sous-entités administratives qui sont décrites plus loin dans le chapitre sont indiquées en **gras** dans la liste et en couleur sur la figure. Chaque entité présentée est sous-titrée avec son emplacement dans la structure du Service public de Wallonie, ceci afin de permettre au lecteur de « cartographier » plus efficacement chaque entité.

3.1.1 Le Service public de Wallonie (SPW)

SPW (environ 10 000 agents)

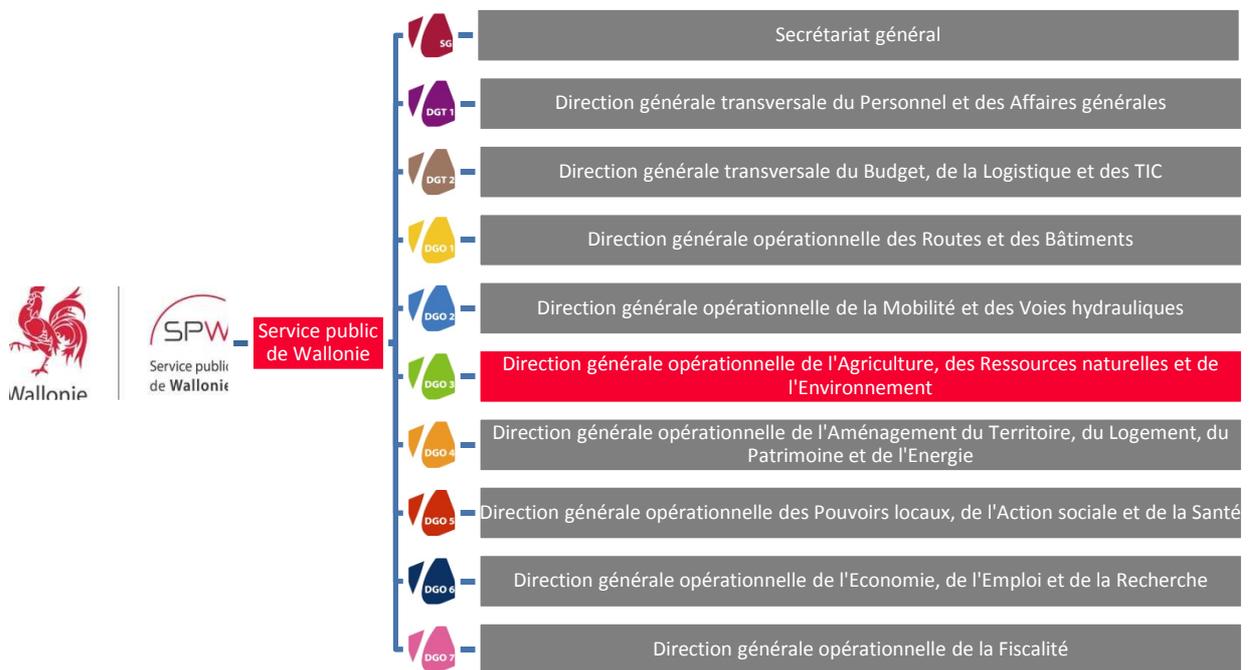


Figure 16: Organigramme du SPW

Le Service public de Wallonie se compose d'un Secrétariat général, de deux Directions générales transversales et de sept Directions générales opérationnelles.

Le Secrétariat général regroupe les stratégies transversales, la communication, l'audit, la géomatique et les fonds structurels.

Les deux Directions générales transversales offrent aux Directions générales opérationnelles un encadrement et une aide dans des matières dites transversales (personnel, juridique, budgétaire, logistique, informatique,...). Ces deux directions sont :

- la Direction générale transversale du Personnel et des Affaires générales et
- la Direction générale transversale du Budget, de la Logistique et des TIC.

Chaque direction générale opérationnelle gère des matières et des compétences spécifiques, en lien direct avec les besoins et les attentes des citoyens, des entreprises, des associations et des pouvoirs locaux. Les sept directions générales sont :

- la Direction générale opérationnelle des Routes et des Bâtiments,
- la Direction générale opérationnelle de la Mobilité et des Voies hydrauliques,
- la **Direction générale opérationnelle de l'Agriculture, des Ressources naturelles et de l'Environnement**,
- la Direction générale opérationnelle de l'Aménagement du territoire, du Logement, du Patrimoine et de l'Energie,
- la Direction générale opérationnelle des Pouvoirs locaux, de l'Action sociale et de la Santé,
- la Direction générale opérationnelle de l'Economie, de l'Emploi et de la Recherche et
- la Direction générale opérationnelle de la Fiscalité.

3.1.2 La Direction générale opérationnelle de l'Agriculture, des Ressources naturelles et de l'Environnement (DGARNE ou DGO3)

SPW → DGARNE (environ 2400 agents)

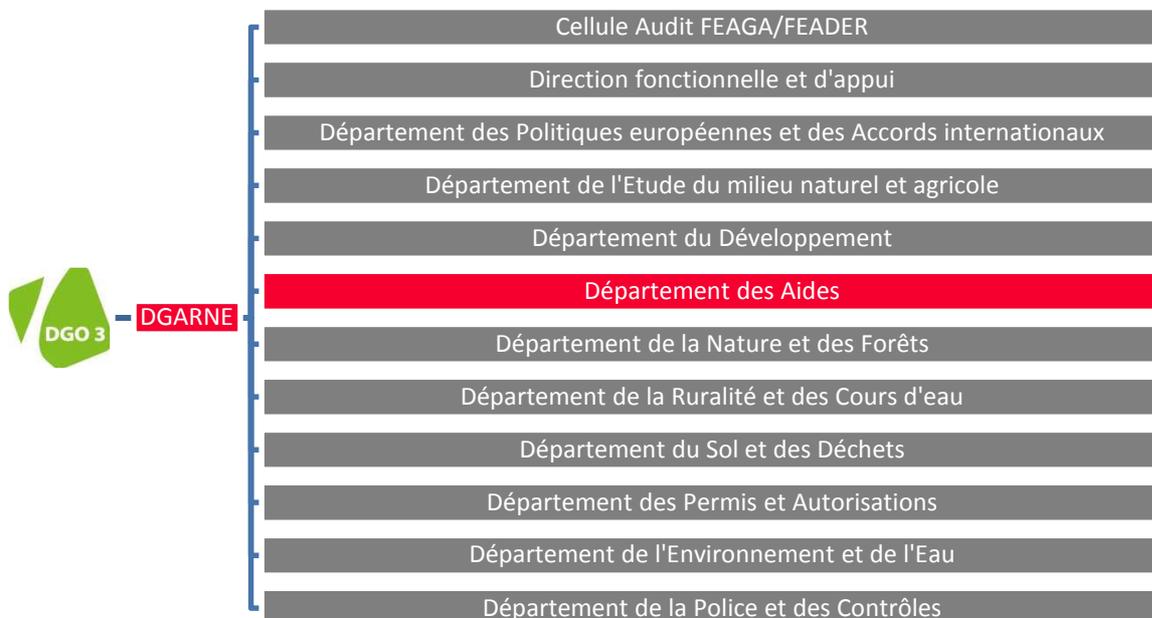


Figure 17: Organigramme de la DGARNE

La Direction générale opérationnelle de l'Agriculture, des Ressources naturelles et de l'Environnement est issue de la fusion mi-2008 de deux Directions générales antérieures : celle de l'Agriculture et celle des Ressources naturelles et de l'Environnement. La DGARNE a pour mission fondamentale le développement durable de la Région wallonne.

La DGARNE est structurée comme suit :

- la Cellule Audit FEAGA/FEADER,
- la Direction fonctionnelle et d'appui,
- le Département des Politiques européennes et des Accords internationaux,
- le Département de l'Etude du milieu naturel et agricole,
- le Département du Développement,
- le **Département des Aides**,
- le Département de la Nature et des Forêts,
- le Département de la Ruralité et des Cours d'eau,
- le Département du Sol et des Déchets,
- le Département des Permis et Autorisations,
- le Département de l'Environnement et de l'Eau et
- le Département de la Police et des Contrôles.

A noter l'existence du **Service de Support informatique et informationnel** qui ne fait actuellement pas partie de l'organigramme officiel de la DGARNE : il s'agit d'une structure interne décidée par le Directeur général et placée sous la dépendance de celui-ci. Ce service est néanmoins décrit plus loin.

3.1.3 Le Département des Aides (D4)

SPW → DGARNE → D4 (environ 340 agents)



Figure 18: Organigramme de la D4

Le Département des Aides assure les fonctions de paiement et de comptabilisation des aides et subventions de l'Union européenne, régionales ou cofinancées et participe au suivi et au maintien de l'agrément de la Direction générale au titre d'organisme payeur pour le compte des fonds agricoles de l'Union européenne. Il assure également le suivi des enquêtes européennes en matière de contrôle de gestion des fonds européens.

Le Département des Aides est structuré comme suit :

- la Direction des Droits et des Quotas,
- la Direction des Surfaces agricoles,
- la Direction des Structures agricoles,
- la **Direction de l'Octroi des Aides agricoles**,
- la Direction d'Ath,
- la Direction de Ciney,
- la Direction de Huy,
- la Direction de Libramont,
- la Direction de Malmedy,
- la Direction de Thuin et
- la Direction de Wavre.

3.1.4 La Direction de l'Octroi des Aides agricoles (D44)

SPW → DGARNE → D4 → D44 (environ 100 agents)



Figure 19: Organigramme de la D44

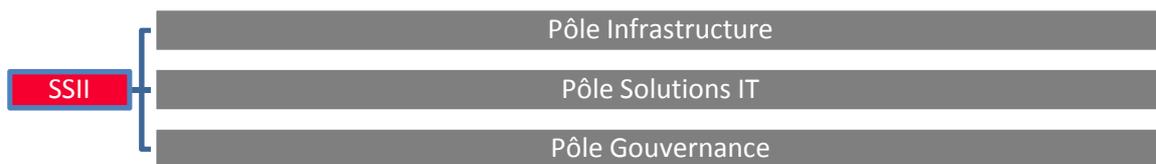
La Direction de l'Octroi des Aides agricoles assure le support juridique transversal concernant les matières liées à l'agriculture, le support à la gestion du Département, le contrôle interne de gestion pour l'organisme payeur, la liquidation des aides, la comptabilisation des opérations financières et le rapportage à l'Union, le suivi des audits externes, la disponibilité des infrastructures spécifiques et de l'informatique métier du Système intégré de gestion et de contrôles (SIGEC) et la production des applications informatiques « métier ».

La Direction de l'Octroi des Aides agricoles est structurée comme suit :

- le Service de Gestion,
- le Service de Support à la Gestion,
- le Service de Comptabilité des Aides et
- le **Service de Production informatique.**

3.1.5 Le Service de Support informatique et informationnel (SSII)

SPW → DGARNE → SSII (7 agents)



Le Service de Support informatique et informationnel est en charge de la gouvernance du système d'information de la DGARNE. Dans ce sens, le SSII est garant de la continuité, de la qualité et de la sécurité du système d'information auprès de la Direction générale. Afin d'assurer son impartialité et son indépendance, le SSII est placé sous la dépendance hiérarchique du Directeur général de la DGARNE.

3.1.6 Le Service de Production informatique (D443 ou SPI)

SPW → DGARNE → D4 → D44 → D443 (environ 70 agents)



Figure 20: Organigramme de la D443

Le Service de Production informatique est la structure interne de développement des logiciels. Il agit sur base de bons de commande internes établis par le Pôle Solutions IT du SSII et validés par le Comité de Direction de la DGARNE.

Les engagements du SPI se résument au niveau de la réalisation du développement des applications métier, de l'assistance à la maîtrise d'œuvre et de la qualité des solutions développées.

Actuellement, le SPI assure toujours l'existence d'une informatique « métier » à l'Organisme payeur wallon (OPW), ceci afin de pouvoir maintenir les conditions d'agrément comme Organisme payeur.

Le Service de Production informatique est structuré comme suit :

- le Pôle du Développement,
- le **Pôle d'Appui au Développement**,
- le Pôle Sécurité,
- le Pôle Exploitation et
- le Service Desk.

Le Pôle du Développement constitue le plus gros des effectifs du SPI, ses missions principales étant de produire les solutions logicielles « métier » et d'en assurer la maintenance.

Le Pôle d'Appui au Développement est décrit plus en détail ci-après.

Le Pôle Sécurité a pour mission principale de définir et sensibiliser à la politique de sécurité de l'Organisme payeur wallon (ISO 27002).

Le Pôle Exploitation prend en charge la gestion de l'infrastructure et le support pour les serveurs, bases de données, réseaux, environnements,... de l'ensemble de la D4.

Le Service Desk est le point d'entrée unique pour prendre en charge l'ensemble des problèmes IT de la D4 (support technique de premier niveau, interventions,...).

3.1.7 Le Pôle d'Appui au Développement (PAD)

SPW → D4 → D44 → D443 → PAD (13 agents)



Figure 21: Organigramme du PAD

Le Pôle d'Appui au Développement a pour objectifs de diminuer les temps de développement et d'augmenter la qualité des productions du SPI, ceci en définissant et en mettant en place les méthodes de travail, les outils et les procédures pour chacun des métiers du développement.

Le PAD est structuré en deux cellules opérationnelles et en trois cellules des représentants du métier :

- la Cellule d'Architecture,
- la **Cellule d'Assurance qualité informatique**,
- la Cellule des Développeurs,
- la Cellule des Chefs de projet et
- la Cellule des Analystes.

La Cellule d'Architecture vise à étudier de nouveaux standards du marché en matière d'architecture logicielle, à compléter le cycle de développement logiciel, à permettre le lien entre les analystes métier et les développeurs et enfin à mettre en place une politique d'architecture orientée service (SOA). Les trois domaines d'activité de la Cellule d'Architecture sont l'architecture système, l'architecture applicative et les analyses détaillées de conception. Elle est également en charge de la gestion des outils communs du SPI.

La Cellule d'Assurance qualité informatique est décrite plus en détail ci-après.

Les trois cellules des représentants du métier (développeurs, chefs de projet et analystes) visent à avoir un représentant de chaque métier lié au développement aux réunions de coordination du PAD et à fournir un service de support aux métiers liés (par exemple, la Cellule des Analystes fournit du support aux analystes en matière de modélisation logicielle avec UML). Ces cellules organisent régulièrement des réunions permettant aux personnes d'un même métier de partager leurs expériences.

3.1.8 La Cellule d'Assurance qualité informatique (AQI)

SPW → D'GARNE → D4 → D44 → D443 → PAD → AQI (5 agents)

La Cellule d'Assurance qualité informatique a pour objectifs de développer, gérer et promouvoir la qualité informatique au sein du SPI. Sa création a été officialisée en janvier 2009 et constitue la plus ancienne cellule du PAD.

Les trois domaines d'activité de la cellule AQI sont la gestion des méthodes & standards du SPI, l'expertise en techniques de test logiciel et le support qualité informatique. Ces trois domaines continuent d'évoluer régulièrement grâce à des activités d'amélioration continue et des projets de mises en place d'activités plus conséquentes.

Un quatrième domaine d'activité est en cours de mise en place : la gestion des indicateurs de qualité dans le cadre de laquelle s'inscrit le présent mémoire. Un projet d'implémentation des indicateurs de qualité a permis de coordonner la déclinaison de l'approche MoCQA aux spécificités du SPI : mise en place des premiers indicateurs, établissement des services proposés, formalisation des procédures et détermination des outils adéquats. Une fois ce projet clôturé, la cellule AQI aura une activité de gestion transversale visant à régulièrement évaluer et améliorer les indicateurs existants, à identifier de nouveaux besoins et à faire des rapports d'interprétation des indicateurs évalués.

A titre indicatif, la Cellule AQI est composée de cinq membres pour une charge moyenne d'environ 2,7 équivalents temps plein.

3.2 La démarche d'amélioration qualité

Le présent projet de déploiement apporte sa pierre à l'édifice dans la démarche d'amélioration globale entreprise au sein de la DGARNE. Cette démarche repose sur les principes du développement durable et de la maîtrise des opérations.

Le développement durable est au cœur de la démarche ; ce principe impose d'évaluer et de rechercher l'amélioration à l'aune des trois piliers du développement durable : l'écologique, le social et l'économique.

La maîtrise des opérations consiste en la mise en place d'un système de management respectant des normes internationalement reconnues : le Système Intégré de Maîtrise des Opérations (SIMO). Ce système concentre un ensemble cohérent d'outils et de documents qui permettent d'améliorer l'efficacité et l'efficience de l'ensemble des agents dans la réalisation des missions confiées à la DGARNE.

La mise en œuvre des missions de la DGARNE nécessite le déploiement d'un système de management permettant la réalisation des missions attribuées par le pouvoir politique dans une recherche permanente d'efficacité et d'efficience.

Il s'agit de déployer une approche de management basée sur l'analyse des processus mis en œuvre au sein de la DGARNE, leur optimisation et leur amélioration permanente dans un esprit de collaboration avec les organes chargés d'assister les Directions générales du SPW dans leur effort de modernisation.

La démarche de management par la qualité totale, entreprise par la DGARNE, est traduite et supportée par le SIMO. Cette démarche se veut à la fois intégrée, collective et rigoureuse. Elle vise l'excellence dans l'exécution des missions qui lui sont confiées. Le Comité de Direction a opté pour la certification progressive de certains éléments clés du SIMO pour plusieurs normes et référentiels reconnus internationalement, comme ISO 9001 par exemple.

3.2.1 Le Système Intégré de Maîtrise des Opérations (SIMO)

Les principes fondamentaux du Système Intégré de Maîtrise des Opérations sont :

- le « **leadership** » (décider, planifier, piloter l'amélioration, communiquer, motiver),
- l'**approche processus** (gestion des ressources, orientation client, valeur ajoutée, optimisation, gestion des flux d'information, gestion des impacts négatifs, gestion du risque et identification des opportunités),
- l'**amélioration continue** (basée sur la *roue de Deming* : planification, mise en œuvre, vérification, action d'amélioration) et
- l'**approche intégrée** (définition des objectifs, planification, mise en œuvre, évaluation, actions correctives et préventives, revues de Direction et de Département).

Sans entrer dans des détails qui n'apporteraient rien au sujet étudié dans le présent mémoire, il est utile de savoir que l'un des objectifs du SIMO est d'assurer la qualité des produits, ce qui justifie l'existence du présent paragraphe : la qualité des produits logiciels étant bien entendu concernée et contribuant à la qualité des produits des processus métier.

3.2.2 L'amélioration continue de la qualité logicielle

Depuis 2007, la hiérarchie de la D443 a pris conscience du besoin d'améliorer la qualité de ses productions : sans entrer dans les détails de l'historique ayant conduit à la situation de départ, on qualifiera l'informatique jusqu'à cette époque d'*artisanale*, sans que ce terme soit compris de manière péjorative. La qualité des réalisations dépendait principalement de la qualité des ressources humaines responsables de les produire. A titre d'exemple, certaines analyses fonctionnelles étaient formalisées en UML, d'autres avec Merise, d'autres en langage non formel,...

La prise de conscience de 2007 s'est concrétisée principalement par la création d'un *Bureau d'Architecture*, dépendant de l'équivalent de l'époque du *Pôle du Développement* actuel. Ce Bureau d'Architecture avait pour principal objectif de définir des méthodologies liées aux métiers du développement et de former les ressources à l'application de ces méthodologies. La D443 constituait donc ainsi une équipe de ressources dont le travail n'était pas directement productif mais visait à améliorer la productivité des autres ressources.

C'est ainsi que des standards internes à la D443 ont vu le jour : référentiel de gestion de projet, dénomination de variables, modélisation logicielle avec UML, techniques de test,... Le Bureau d'Architecture s'est vu remanié dans sa structure en 2009 avec la création progressive de cellules (dont la cellule AQI est la première née) et ses missions se sont étendues, notamment pour tendre vers une informatique plus moderne. Début 2011, le Bureau d'Architecture a été officiellement remplacé par le Pôle d'Appui au Développement qui est décrit plus avant.

Même s'il reste beaucoup de chemin à parcourir, les avancées depuis les cinq dernières années en termes de technologie, de formalisme, de structure, de bonnes pratiques et de mentalité sont encourageantes, tout cela en conservant la productivité nécessaire à la réalisation des projets commandités par les utilisateurs.

3.2.3 Le projet d'implémentation des indicateurs de qualité sur les produits

Dans l'optique de l'amélioration continue de la qualité logicielle (voir section précédente), une proposition de projet a été soumise à la hiérarchie D443 en octobre 2010 par la cellule AQI, visant l'implémentation d'indicateurs de qualité sur nos produits logiciels (nom de code QUALITE_PRODUIITS).

Au départ, l'idée derrière ce projet était de se diriger vers l'étude des normes ISO 9126, ISO 14598 et ISO 250xx qui concernent la qualité des produits logiciels, plutôt par méconnaissance des autres possibilités existantes que par certitude de l'applicabilité de ces normes.

En décembre 2010, une sélection de projets a été soumise à Naji Habra, spécialiste du domaine de la qualité logicielle aux FUNDP, par Samuel Hanoteau qui cherchait un sujet de mémoire qui serait utile à la recherche, productif pour la D443 et compatible avec ses centres d'intérêt. Le choix s'est vite porté sur le projet QUALITE_PRODUIITS qui pourrait permettre de déployer l'approche MoCQA en environnement professionnel. Un accord « Win-Win » a été établi entre la D443 et Samuel Hanoteau et le projet a été officiellement accepté en janvier 2011.

Il fut estimé que l'approche MoCQA était adéquate à déployer au sein de la D443 pour les raisons principales suivantes :

- la définition de modèles sur mesure en fonction de l'environnement est particulièrement compatible avec l'approche habituelle de la hiérarchie D443 et des membres du Pôle d'Appui au Développement ;
- la possibilité de se tourner vers les modèles de qualité ISO (9126 ou 25000) par la suite sans devoir remettre en cause le choix de l'approche MoCQA ;
- le besoin à moyen terme de mettre en place une activité de contrôle qualité, pour laquelle des indicateurs de qualité constituent un premier pas objectif et
- le peu de ressources disponibles au sein de la cellule AQI qui demande une approche structurée et itérative pour la mise en place des indicateurs.

3.3 Présentation de l'étudiant

Sorti diplômé de la Haute Ecole d'Enseignement Supérieur de Namur en juin 2000, Samuel Hanoteau a opté pour le métier de consultant informatique, ceci afin d'acquérir de l'expérience au fil des missions proposées par son employeur.

Enchaînant d'abord des missions de développement et d'analyse fonctionnelle, Samuel s'est progressivement formé au domaine de la gestion de projet. Tout au long de son expérience, Samuel a souvent eu des difficultés à travailler de manière non formalisée, chose qu'il est arrivé qu'on lui impose « dans l'urgence » : le besoin d'écrire et d'appliquer des procédures reproductibles, la nécessité de documenter, l'importance de tester formellement et l'envie de satisfaire tous les intervenants (qu'ils soient utilisateurs, commanditaires, commerciaux, développeurs,...) ont souvent guidé ses actes.

Début 2008, alors que Samuel venait de clôturer un projet dont il avait la responsabilité, une nouvelle mission s'est présentée au SPW : on demandait une ressource connaissant la gestion de projet, les méthodes d'analyse fonctionnelle et les techniques de test logiciel : très vite, Samuel s'est épanoui dans cet environnement où le challenge est omniprésent. Enormément de bonnes pratiques à mettre en place et surtout le besoin de qualité dû à la structure complexe de la D443 (rarement implémenté dans de petites structures) font que la motivation est toujours la même, après plus de quatre années de travail.

Samuel a d'abord intégré un projet pour lequel il a mis en place des bonnes pratiques en matière de tests unitaires, pour ensuite intégrer le Bureau d'Architecture (décrit plus haut) la même année. Une Cellule Qualité a été créée fin 2008 pour devenir, début 2009, la Cellule d'Assurance qualité informatique que l'on connaît et que Samuel coordonne.

Depuis début 2012, Samuel est également Contact Qualité D443, ce qui implique qu'il est le relais D443 au sein du réseau qualité mis en place dans le cadre du SIMO (voir plus avant), dépassant donc le domaine de la qualité informatique pour toucher le domaine de la qualité d'entreprise.

Chapitre 4 : Le déploiement

Le présent chapitre vise à expliquer le déploiement de l'approche MoCQA en lui-même : la prise de connaissance de l'approche, les difficultés rencontrées, les résultats des premières itérations,... Il constitue déjà un premier retour d'expérience sur la faisabilité du déploiement de l'approche.

A noter que l'ensemble des livrables de ce déploiement se trouve dans l'annexe 2 du présent document, parfois sous forme obscurcie pour des causes de confidentialité des données. Il est important de garder à l'esprit qu'il s'agit ici d'une synthèse du déploiement, le détail se trouvant dans les livrables de l'annexe 2.

4.1 Prise de connaissance de l'approche MoCQA

La prise de connaissance de l'approche MoCQA elle-même a commencé mi 2011 au moyen des quelques documents disponibles sur le domaine [Vanderose 2010], [Vanderose, Kamseu & Habra 2010], [Vanderose, Habra & Kamseu 2011].

Rapidement, la complexité des termes « universitaires » très théoriques par rapport à l'approche plus concrète et pragmatique du milieu « professionnel » a identifié le besoin d'une vulgarisation de l'approche afin de la rendre accessible à un plus large panel d'environnements. La réponse à ce besoin est développée dans le chapitre suivant.

Cette complexité, ainsi que le fait que l'approche MoCQA soit toujours à l'état de développement au sein des FUNDP, ont rendu relativement lente la progression de la première itération du projet de déploiement, notamment sur les premières phases et a parfois donné lieu à des retours en arrière sur des phases déjà réalisées pour conserver un tout cohérent.

Il a été décidé de travailler en mode itératif, à savoir se limiter à un seul indicateur dans un premier temps. La motivation de travailler avec une seule itération pour commencer était la volonté de réaliser l'ensemble du processus MoCQA au moins une fois pour fin 2011, de sorte de pouvoir obtenir des résultats concrets et exploitables le plus tôt possible, ce qui a permis d'avoir un premier feedback pour le présent sujet de mémoire mais aussi de rassurer la hiérarchie D443 sur l'applicabilité de la méthode.

4.2 Première itération

4.2.1 Phase 1 : acquisition

Cette phase est probablement la plus difficile puisqu'elle suppose de déterminer par où commencer. Avec une population d'environ 70 agents dans le Service de Production Informatique et 2400 utilisateurs dans la DGARNE (et donc autant d'intervenants théoriques possibles), il n'était pas envisageable de choisir une approche exhaustive.

Un échantillon de 5 intervenants a été sélectionné pour recueillir leurs besoins en termes de qualité sur les produits logiciels. Ces cinq intervenants sont :

- le responsable du Pôle d'Appui au Développement,
- le responsable du Pôle du Développement,
- le responsable du Pôle Service Desk,
- le coordinateur de la Cellule d'Architecture et
- le coordinateur de la Cellule d'Assurance Qualité Informatique.

Une interview entre l'étudiant et chacun des intervenants a eu lieu permettant d'identifier 26 objectifs de qualité au total.

Un tableau de l'ensemble de ces objectifs a été établi et une réunion avec la hiérarchie du SPI a permis de sélectionner un objectif à déployer lors de la première itération. L'objectif sélectionné était issu d'un besoin exprimé par le Pôle Service Desk, à savoir l'objectif « Fiabilité ». Cet objectif a été sélectionné plutôt qu'un autre car il semblait moins complexe de premier abord et que les entités mesurables liées semblait déjà correctement identifiées.

C'est lors de cette première phase que la réflexion sur la typologie des intervenants, expliquée au chapitre 5, a eu lieu.

4.2.2 Phase 2 : modélisation

La phase 2 a consisté à instancier le méta-modèle MoCQA une première fois (pour l'étudiant) afin de modéliser l'objectif « Fiabilité ». Rapidement, une structuration de l'objectif en 4 sous-objectifs a eu lieu, en fonction des besoins exprimés lors de la phase 1 et des données à disposition :

- Survenance des incidents,
- Disponibilité,
- Impact des incidents et
- Fréquence de déploiement.

Le formalisme utilisé pour la représentation est la notation UML dédiée aux diagrammes d'objets. La figure ci-dessous illustre la représentation MoCQA de cette structuration.

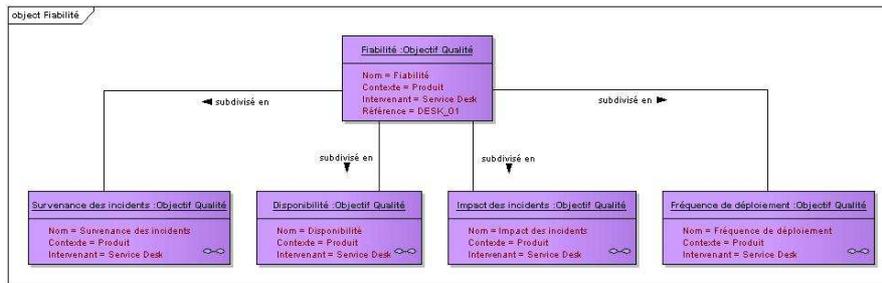


Figure 22 : Structure de l'objectif de qualité "Fiabilité"

Chaque sous-objectif a fait ensuite l'objet d'une modélisation MoCQA. La figure suivante illustre le sous-objectif « Surveillance des incidents ».

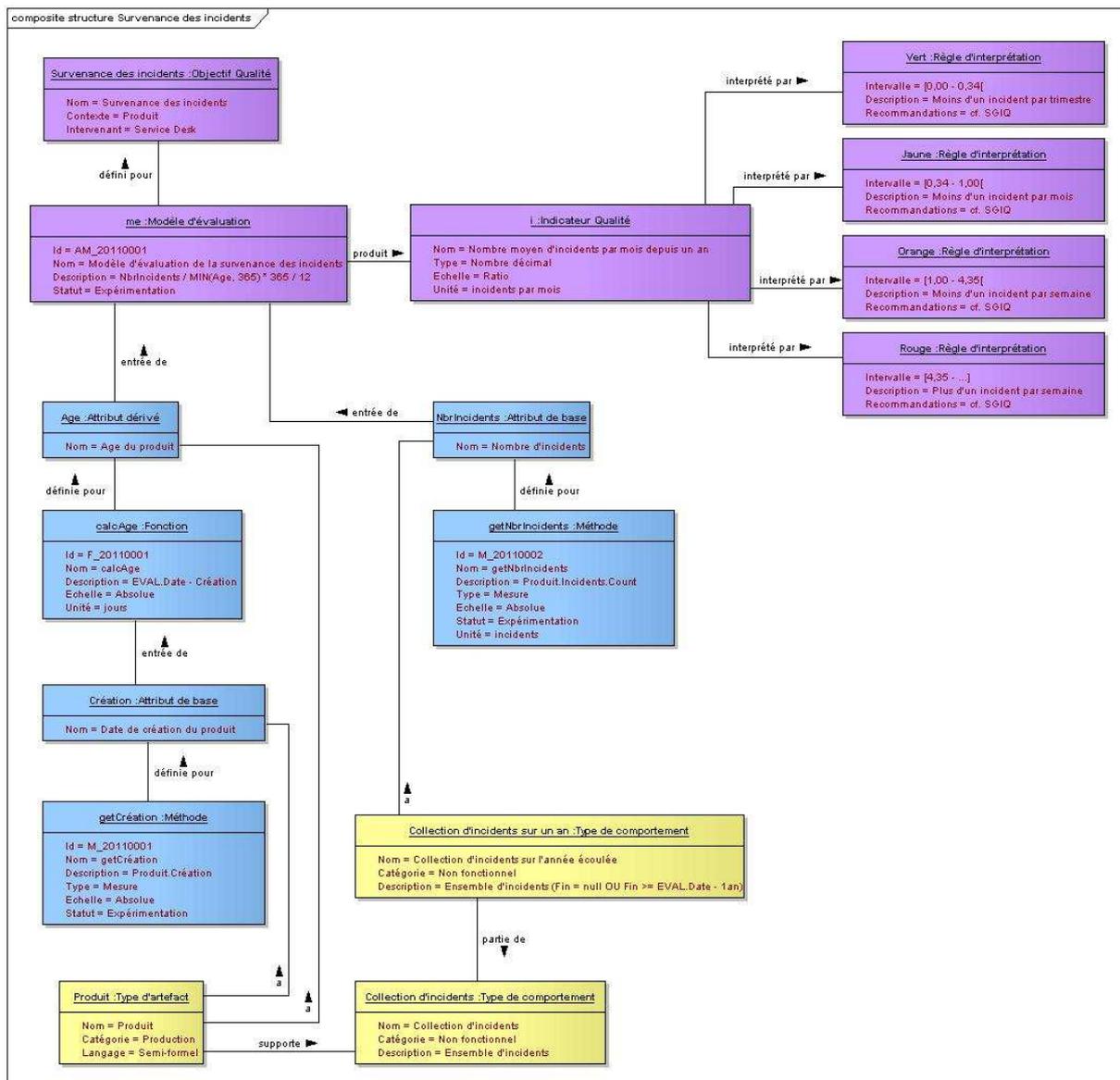


Figure 23 : Modèle MoCQA "Surveillance des incidents"

Les 4 sous-objectifs ont ainsi donné naissance à 4 sous-modèles MoCQA : c'est lors de cette étape que de nombreuses questions sont arrivées, ayant donné lieu à une majorité des contributions reprises au chapitre suivant.

4.2.3 Phase 3 : opérationnalisation

Cette phase visait à déterminer comment acquérir les attributs de base nécessaires à l'évaluation des 4 indicateurs de qualité modélisés lors de la phase de modélisation.

A cette fin, chaque attribut s'est vu attribuer une fiche expliquant comment le mesurer, la plupart du temps sous forme d'une requête SQL à faire sur un serveur particulier.

Afin de permettre une avancée plus rapide dans l'itération, aucune des mesures ne s'est vue automatisée, chaque requête devant être exécutée « manuellement ».

4.2.4 Phase 4 : évaluation

Sans doute la phase la plus stimulante de la première itération, l'évaluation allait enfin permettre d'obtenir des résultats exploitables et présentables à la hiérarchie SPI.

La majorité des métriques portant sur des incidents (à savoir *toute plainte remontée au Service Desk par un utilisateur concernant un produit logiciel particulier*), la question de l'échantillon de produits logiciels concernés par l'évaluation a été résolue très rapidement : l'évaluation porterait sur l'ensemble des produits SPI, à savoir les applications métiers qui y sont développées, les frameworks/services/composants propriétaires qui y sont développés, ainsi que les progiciels/outils qui y sont utilisés⁴.

Un tableau de bord (au format Excel) a été créé pour récolter les résultats de cette évaluation. Ce tableau était conforme au modèle MoCQA, chaque indicateur y figurant pouvant être lié au modèle pour une meilleure compréhension des mécanismes mis en œuvres pour obtenir sa valeur.

4.2.5 Phase 5 : exploitation

Pour cette phase d'exploitation, un comité de suivi des indicateurs a été créé au sein du SPI, composé d'un noyau fixe et d'invités en fonction des résultats exploités. Une première réunion de ce comité a eu lieu pour exploiter les résultats de l'évaluation « Fiabilité ».

14 constatations ont été faites, chacune donnant lieu à une action déterminée et encodée dans le système de gestion des tâches⁵ du SPI, avec à chaque fois un responsable désigné. Ces actions concernaient principalement des erreurs de modélisation ou des problèmes d'organisation empêchant une évaluation totalement correcte. Pour des raisons de confidentialité, ces constatations ne peuvent être étudiées ici plus en profondeur.

Pour cette première itération, le plan d'actions a porté uniquement sur les indicateurs eux-mêmes (modélisation, etc.) et n'a pas donné lieu à des actions portant sur les produits logiciels SPI, le projet étant jugé encore trop peu mature pour être communiqué à ce stade. Il est toutefois très intéressant de signaler que certains résultats jugés faibles renforçaient un « ressenti » négatif de la part de la hiérarchie sur la qualité de certains produits. Ce point est développé dans le chapitre suivant.

⁴ Concernant les progiciels et outils utilisés au sein du SPI, un périmètre a été défini afin de se limiter aux progiciels et outils *choisis* par le SPI (en opposition aux progiciels et outils *imposés* par le SPW au SPI).

⁵ Pour information : Atlassian Jira.

4.3 Itérations suivantes

Début 2012, d'autres itérations ont suivi, généralement de manière plus rapide.

Une itération a porté sur l'objectif « Respect du cycle de vie » qui consiste à s'assurer que les produits développés au sein du SPI respectent les différents livrables attendus tout au long du cycle de développement du SPI (analyse fonctionnelle, architecture, tests, etc.). A l'heure de la rédaction du présent document, cette itération est en attente d'une première évaluation.

Une autre itération a porté sur l'objectif « Réception CeSAM » qui consiste en une aide à la décision lors du passage d'une application métier du mode de gestion « projet » en mode de gestion « maintenance », dont les cellules CeSAM⁶ sont responsables. A noter que cette itération particulière a été réalisée dans le cadre du travail du cours *Qualité des produits et des processus* de Naji Habra et ne devrait pas être évaluée comme faisant partie du présent mémoire.

Enfin, une nouvelle évaluation sur l'objectif « Fiabilité » a eu lieu après réalisation de la majorité des actions identifiées.

4.4 Communications

L'aspect psychologique lié à l'interprétation des indicateurs (voir chapitre 5) a rapidement identifié que la communication était très importante concernant le déploiement des indicateurs.

C'est pourquoi plusieurs communications ciblées en fonction du profil ont eu lieu au sein du SPI afin de commencer une sensibilisation à la démarche et de recueillir des premiers avis intéressants à la fois pour les prochaines communications à plus large public et pour un retour d'expérience aux FUNDP.

4.5 Suite

Il est évident que la gestion des indicateurs de qualité sur les produits logiciels du SPI ne s'arrête pas là, l'expérience étant un succès. Bien que le projet touche à sa fin lors de l'écriture du présent mémoire, l'activité, elle, continuera et constituera un nouveau domaine d'activités pour la cellule AQI.

Un standard interne de gestion des indicateurs, en quelque sorte une instanciation du guide de déploiement fourni dans l'annexe 1, sera réalisé.

Une réflexion aura lieu concernant l'automatisation de certaines mesures, ainsi que la manière de présenter les indicateurs (avec droits d'accès en fonction du profil de la ressource).

⁶ Les deux Cellules des Systèmes Applicatifs Métiers (CeSAM) sont responsables de la maintenance corrective et évolutive des applications métiers gérées par le SPI.

Chapitre 5 : Contributions

Le présent chapitre vise à synthétiser les contributions du mémoire : c'est ici qu'on décrit le niveau de satisfaction des deux premiers objectifs du travail, à savoir, pour rappel :

- une participation à la validation de l'approche MoCQA via une utilisation concrète dans un environnement professionnel, ce qui permettra de s'assurer de la faisabilité de celle-ci, de donner un retour d'expérience aux FUNDP et d'éventuellement suggérer des adaptations et
- la rédaction d'un guide de déploiement MoCQA destiné à toute entreprise désireuse de se lancer dans un tel déploiement.

C'est également ici que sont développées les différentes réflexions qui ont dû être menées tout au long du projet.

5.1 Retour d'expérience sur MoCQA

Dès la première itération du déploiement MoCQA, plusieurs questions sont survenues. La majorité avait déjà été étudiée par les FUNDP, mais certaines n'avaient pas été envisagées : elles sont donc reprises ici, accompagnées de la solution utilisée lors du déploiement.

5.1.1 Différence entre les notions de projet et de produit

La méthodologie MoCQA et le méta-modèle qui l'accompagne définissent que l'évaluation de la qualité porte sur un **projet**, défini comme l'ensemble des artefacts, comportements et dérivations d'un logiciel, cela tout au long de son existence.

Cette compréhension du terme *projet* est différente de l'usage qui en est fait dans la majorité des entreprises⁷, où la définition donnée par les méthodes de gestion de projet est employée. Le guide PMBOK définit notamment un projet comme étant « une entreprise temporaire initiée dans le but de créer un produit, un service ou un résultat unique » [PMI 2009].

C'est d'ailleurs cette définition qui est employée pour le terme *projet* au sein de l'environnement de déploiement dont il est question ici. A cause de ce fait, il a été décidé d'employer le terme **produit** lors du déploiement. C'est également ce terme qui est utilisé dans le présent mémoire.

Sans rentrer dans les détails, on peut résumer qu'une évaluation MoCQA ne porte pas que sur un livrable logiciel, mais aussi sur l'ensemble des processus exécutant des transformations sur ce logiciel, ainsi que les différents comportements de ce logiciel.

Bien que ce point ait fait l'objet de nombreuses discussions entre Benoît Vanderose et Samuel Hanoteau, aucune solution de conciliation n'a été trouvée. Les deux parties reconnaissent que le terme *projet* est trop limité dans le temps et que le terme *produit* est trop limité dans la cible de l'évaluation. Il serait intéressant de trouver (ou d'inventer) un terme réunissant les deux concepts.

⁷ Information basée sur l'expérience de l'étudiant qui a travaillé, au moment de la rédaction de ce mémoire, avec un peu plus d'une dizaine d'entreprises différentes.

5.1.2 Typologie des intervenants

La première étape de la méthodologie MoCQA consiste à identifier les intervenants pour le produit ciblé par l'évaluation. A chaque fois qu'une évaluation est faite pour un nouveau produit, on doit identifier ces intervenants et recueillir leurs besoins en termes de qualité.

Comme on peut le constater dans le chapitre 3 décrivant l'environnement de déploiement, certains agents sont impliqués dans un périmètre limité de produits, tandis que d'autres agents sont impliqués, de par leur mission de support, dans l'ensemble des produits. Pour ces derniers, il est clair que leurs besoins en termes de qualité sont identiques d'un produit à un autre.

Il a donc été décidé de déterminer une typologie des intervenants :

- les **intervenants applicatifs**, dont les besoins sont, ou peuvent être, différents d'un produit à un autre et
- les **intervenants génériques**, dont les besoins sont identiques d'un produit à un autre.

A titre d'exemple, on pourra réunir dans les intervenants applicatifs le responsable informatique du produit, ses utilisateurs, le commanditaire, le chef de projet (si le produit est un livrable d'un projet en cours), les développeurs, les analystes,...

On réunira dans les intervenants génériques le chef de service, les responsables qualité, sécurité, architecture, exploitation,...

5.1.3 Révision du méta-modèle

Lors de l'étude de la sémantique du méta-modèle, certaines incohérences (syntaxiques UML et sémantiques) ont été relevées. Elles ne sont pas listées ici mais il est intéressant de noter que ce déploiement a permis de les relever, probablement du fait d'un regard nouveau sur le méta-modèle.

Egalement durant le projet, le méta-modèle a continué à évoluer dans le temps (ajouts d'attributs, changement de noms,...) : l'intégration par l'étudiant constituait également une révision donnant un feedback rapide à Benoît Vanderose.

5.1.4 Notion temporelle

Lors de la première itération pour l’objectif « Fiabilité », certains indicateurs se basent sur une notion temporelle. Par exemple, un indicateur donne le « nombre moyen d’incidents par mois depuis un an ». La question s’est posée de comment modéliser le fait de devoir récupérer les incidents vieux de maximum un an.

La figure suivante montre comment sont structurés le produit et les incidents au format MoCQA :



Figure 24 : Modélisation MoCQA d'une collection d'incidents

On peut y voir qu’un produit supporte une collection d’incidents qui est composée d’incidents.

Une première solution visant à définir un attribut « Nombre d’incidents depuis un an » a été envisagée, ce qui a donné la modélisation suivante :

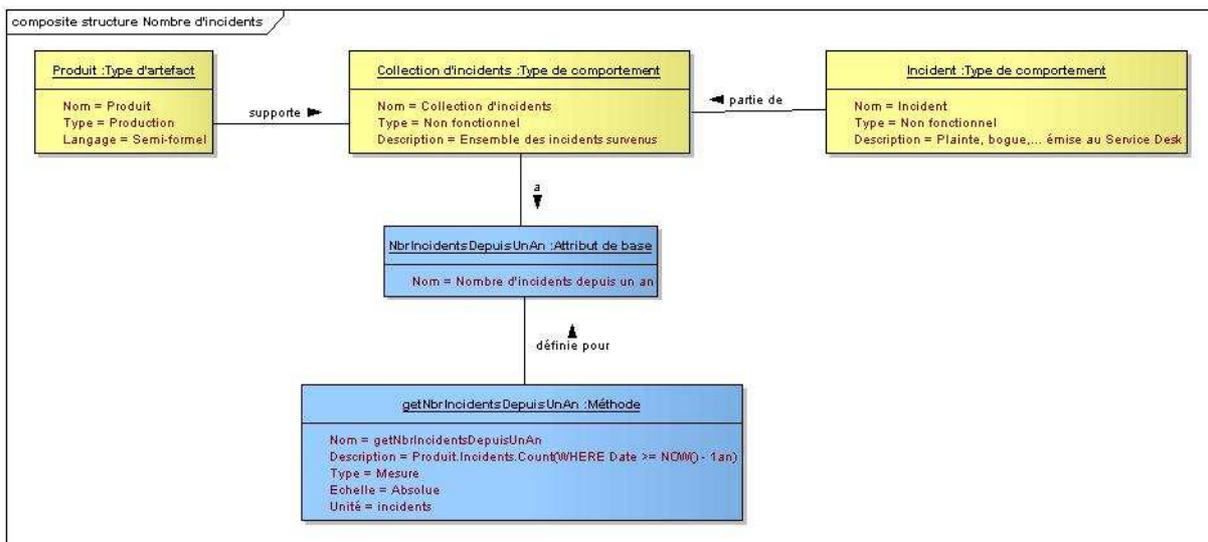


Figure 25 : Modélisation MoCQA de la mesure du nombre d'incidents (solution 1)

L’avantage de cette solution est qu’il n’y a pas de modifications à apporter aux concepts orientés produit mais a l’inconvénient que l’attribut « Nombre d’incidents depuis un an » ne permet pas d’être réutilisé dans un autre modèle pour une autre période de temps (par exemple, recueillir le nombre d’incidents depuis 3 mois demanderait la définition d’un nouvel attribut, accompagné de sa propre méthode de mesure).

De plus, cette solution détourne quelque peu la philosophie de MoCQA qui énonce qu’une entité mesurable (ici, la collection d’incidents) a des attributs (ici, le nombre d’incidents). Le fait de définir l’attribut comme étant « depuis un an » fait qu’il ne s’agit pas vraiment d’une mesure sur la collection d’incidents, mais sur une partie de celle-ci. Cette réflexion a amené à une autre solution, modélisée dans la figure suivante :

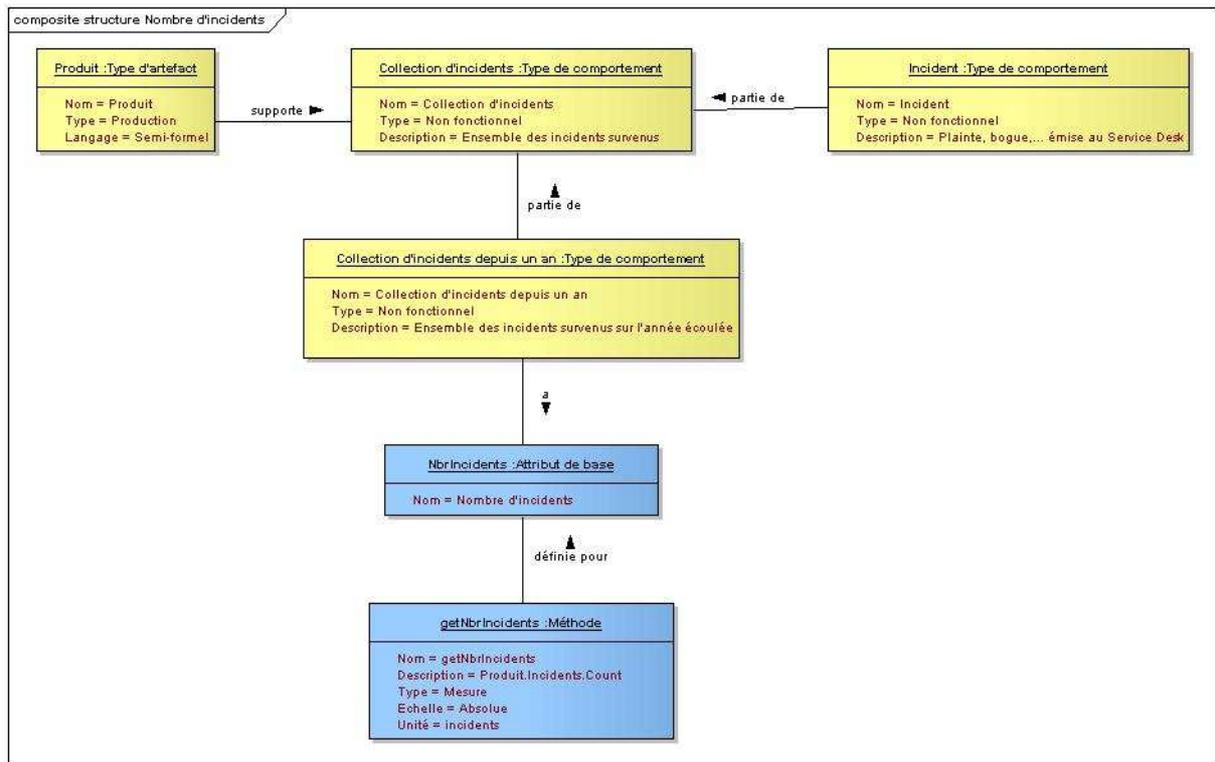


Figure 26 : Modélisation MoCQA de la mesure du nombre d'incidents (solution 2)

Cette solution introduit une nouvelle entité mesurable : la collection d'incidents depuis un an, qui est un sous-ensemble (lien « partie de ») de la collection d'incidents. Elle a l'avantage de pouvoir réutiliser l'attribut « Nombre d'incidents » (débarrassé de sa notion temporelle) : en cas de besoin du même attribut sur une autre période de temps, il suffit de définir un nouveau sous-ensemble de la collection d'incidents.

En restant au niveau de la modélisation, on pourrait ne pas se rendre compte de l'avantage de la seconde solution, puisqu'il y a finalement une entité supplémentaire. C'est dans le plan de mesure et l'éventuelle automatisation que l'avantage ressortira puisqu'une seule méthode générique pour récupérer le nombre d'incidents sur un ensemble donné devra être définie.

5.2 Guide de déploiement de l'approche MoCQA

Le second objectif du présent mémoire vise à rédiger un guide de déploiement de l'approche MoCQA en environnement professionnel : le besoin, identifié rapidement en début de déploiement, vient de la différence de langage utilisé au sein de l'université et au sein de l'environnement professionnel.

En effet, le langage universitaire se doit d'être très précis et le plus générique possible, tandis que le langage professionnel est généralement plus concret et axé sur la pratique.

Lors de ses premières lectures des articles concernant MoCQA, l'étudiant se faisait une certaine idée de la complexité de la méthodologie. Par la suite, lorsque les concepts ont commencé à être compris, il a constaté que la complexité réelle était inférieure à la première idée qu'il s'en faisait.

Sans support « humain », la seule lecture des articles a donc été jugée comme insuffisante par l'étudiant et son promoteur, ce qui a amené à l'identification du présent objectif, à savoir « vulgariser » les concepts universitaires afin de favoriser l'utilisation de l'approche MoCQA dans un environnement professionnel.

Le guide reprend une explication de ce qu'est MoCQA et de comment le déployer : une description détaillée du méta-modèle permettant de comprendre la signification de chaque concept et de chaque attribut est donnée. Le processus de gestion des indicateurs de qualité est également décrit. Enfin, des bonnes pratiques et lignes de conduite sont proposées, sortant directement de l'expérience acquise lors du déploiement.

Ce guide est joint au présent mémoire dans l'annexe 1.

5.3 Utilisation de la langue française

L'environnement dans lequel le déploiement a eu lieu étant une entité administrative de la Région wallonne, le français est la langue officielle qui y est utilisée.

C'est ainsi qu'une version parallèle du méta-modèle MoCQA a été réalisée par l'étudiant, reflet du *MoCQA metamodel* traduit en français. Cette traduction a eu l'avantage pour l'étudiant de bien prendre en main le méta-modèle et d'en cerner les détails.

A noter qu'afin de permettre son utilisation dans l'environnement de déploiement, le guide de déploiement a été également rédigé en français.

5.4 Cycle de développement au format MoCQA

Le SPI où le déploiement a eu lieu étant dans une logique ALM (Application Lifecycle Management), les modèles MoCQA qui sont et seront développés porteront généralement sur des types d'entité mesurable communs, le cycle de vie des produits au sein de l'organisme étant standardisé. A titre d'exemple, un code source sera toujours une partie du produit développé et sera toujours composé de classes qui sont elles-mêmes composées de méthodes, etc.

Afin de conserver une cohérence des types d'entité mesurable au travers des différents modèles MoCQA réalisés, le cycle de vie des produits a été modélisé au format MoCQA, c'est-à-dire uniquement avec des concepts orientés produit dans un même schéma. Cela a l'avantage de toujours utiliser les mêmes termes pour un même concept (exemple : un type d'entité mesurable « analyse fonctionnelle » portera toujours ce nom alors que, en l'absence d'une formalisation claire, on pourra une fois modéliser « analyse fonctionnelle », une autre fois « analyse », ou encore « analyse des exigences », etc.).

D'un côté pratique lors de la modélisation MoCQA, cette schématisation permet également de réutiliser rapidement des types d'entité mesurable, le *qualiticien*⁸ ayant à sa disposition une « bibliothèque » des types d'entité mesurable existants.

Il a été envisagé, en cours de projet, de définir totalement le cycle de vie des produits SPI au format MoCQA, mais cette solution a finalement été jugée improductive. Il a plutôt été décidé de créer ce modèle « transversal » progressivement dans le temps, au fur et à mesure que les modèles MoCQA réalisés ont besoin des types d'entité mesurable.

Ci-dessous (Figure 27), le modèle du domaine des produits du SPI⁹ est proposé à titre informatif. Ensuite, les mêmes concepts sont présentés au format MoCQA pour les produits (Figure 28), les produits développés (Figure 29) et les applications métier (Figure 30).

Un « produit développé » étant une spécialisation d'un « produit » et une « application métier » étant une spécialisation d'un « produit développé », trois diagrammes ont été réalisés au format MoCQA pour représenter chaque concept.

⁸ Le terme *qualiticien* est ici l'appellation du profil métier des membres de la cellule d'Assurance Qualité Informatique.

⁹ Allégué ici pour ne présenter que les concepts déjà utilisés au sein des modèles MoCQA développés

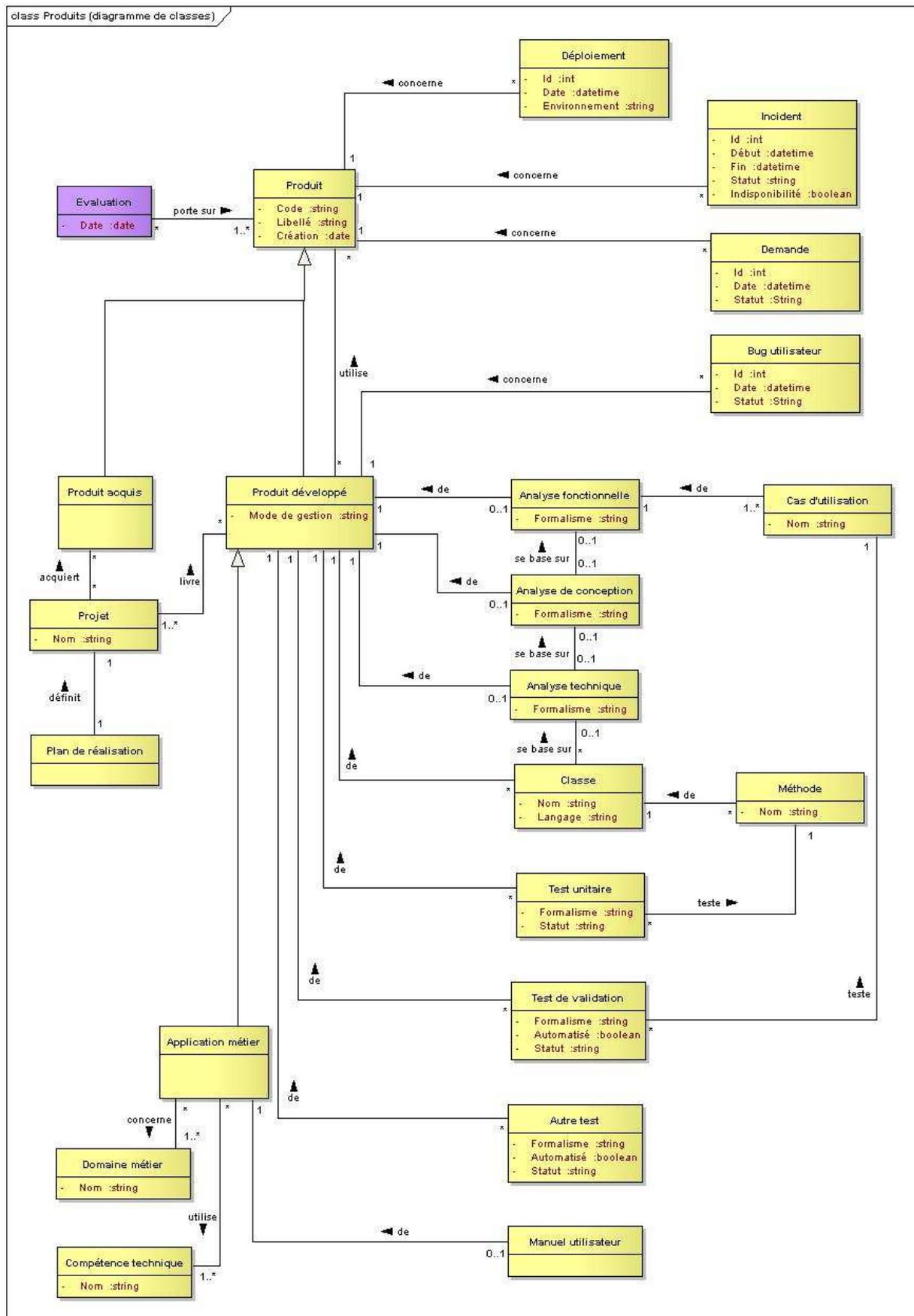


Figure 27 : Modèle du domaine des produits SPI

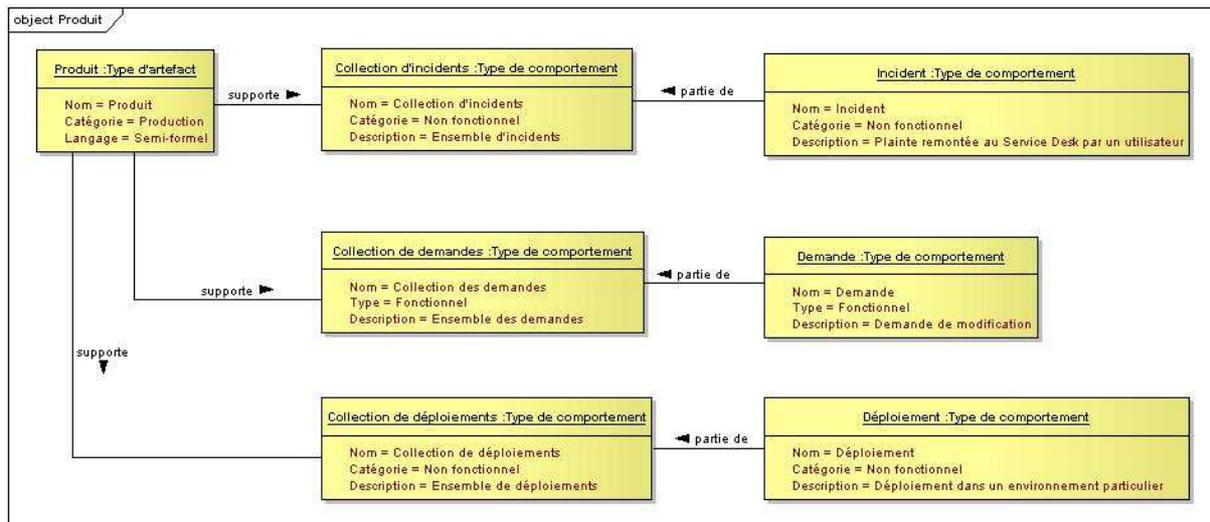


Figure 28 : Modèle MoCQA des entités mesurables liées au Produit

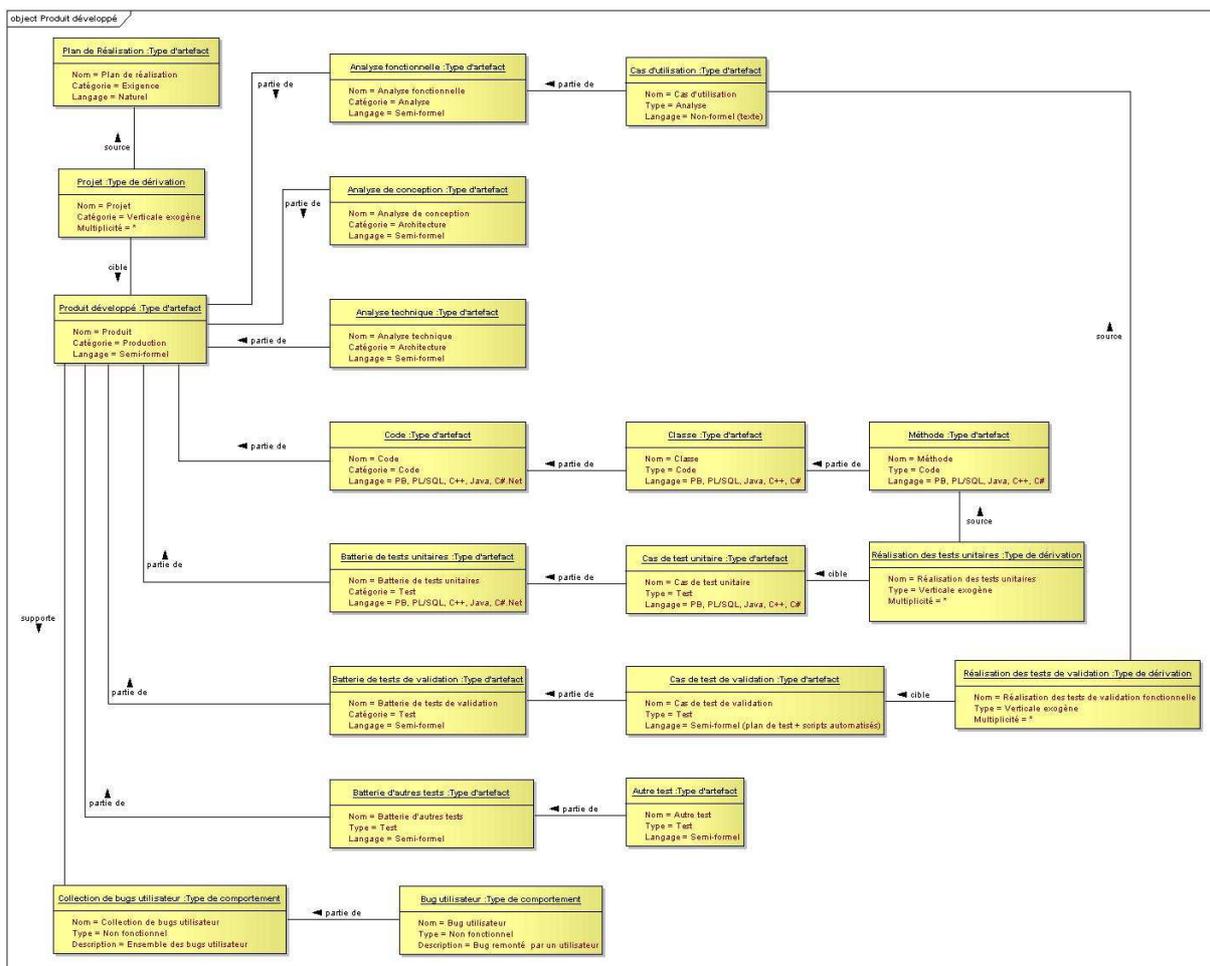


Figure 29 : Modèle MoCQA des entités mesurables liées au Produit développé

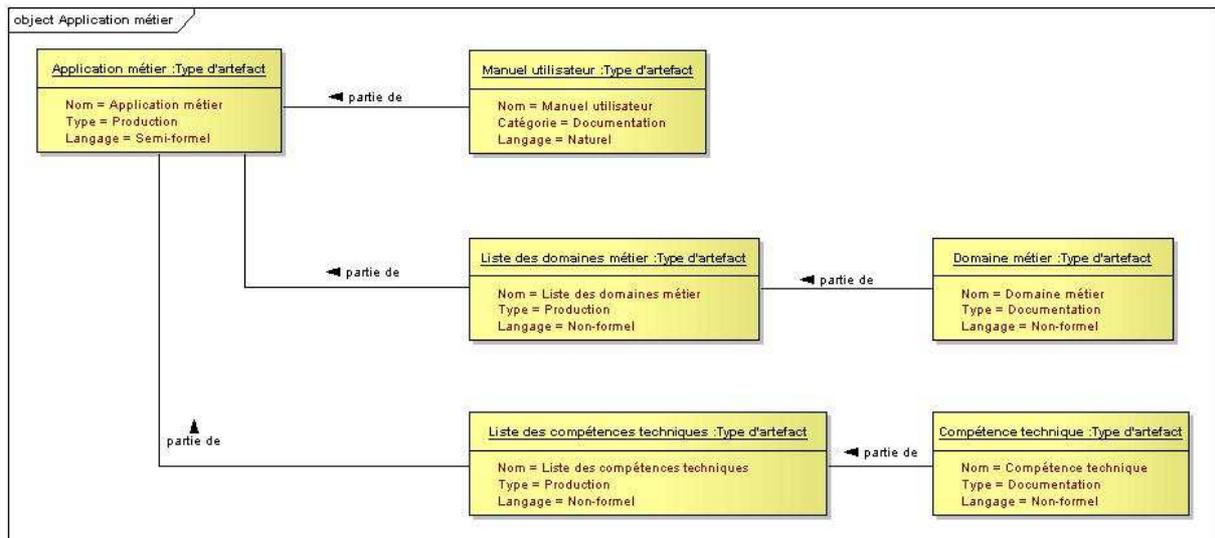


Figure 30 : Modèle MoCQA des entités mesurables liées à l'Application métier

5.5 Evaluation du coût

Etant réalisé en partie dans le cadre d'un projet du SPI, les coûts des activités ont été enregistrés.

L'étudiant a bien pris garde de séparer les activités de prise en main de l'approche MoCQA des activités de production réelle liées aux indicateurs de qualité.

Ces coûts sont repris dans l'annexe 3. On peut y constater que la réalisation de la première itération sur les indicateurs de qualité a coûté moins de 10 J.H, ce qui ne semble pas excessif du point de vue du SPI. En effet, une crainte régulière concernant le domaine de la qualité est son coût excessif.

Bien que l'expérience pourra apporter plus d'informations à ce sujet, l'étudiant estime que cet ordre de coût sera conservé pour une forte majorité des indicateurs. Il est évident que selon la complexité de ceux-ci, le coût sera nuancé.

5.6 Réflexion sur la psychologie du déploiement

Au-delà de l'application de l'approche MoCQA elle-même, la mise en place d'indicateurs de qualité au sein d'une entreprise peut générer de la résistance par les différentes ressources humaines. En effet, la nature humaine fait qu'on aime rarement être évalué voire contrôlé, car cela peut être pris pour un manque de confiance ou un ciblage des défaillances des individus.

C'est pourquoi plusieurs éléments clés ont été identifiés, dont il est important de tenir compte lors de l'évaluation d'un produit ou des communications réalisées aux ressources humaines. Ces éléments sont décrits ci-après.

5.6.1 Interprétation des indicateurs

Un indicateur n'étant, au final, qu'une information, la plupart du temps un nombre, son interprétation pourrait être différente d'une personne à l'autre. C'est pourquoi il est important de définir des règles d'interprétation les plus claires et les plus complètes possibles. Et en fonction des circonstances et du contexte, un même indicateur pourra être interprété différemment.

La figure ci-dessous reprend un extrait des tableaux d'interprétation de l'indicateur « Nombre moyen d'incidents par mois depuis un an » du modèle « Fiabilité » :

Intervalles

Code couleur	Intervalle de valeurs	Description
Vert	$0,00 \leq i < 0,34$	moins d'un incident par trimestre
Jaune	$0,34 \leq i < 1,00$	moins d'un incident par mois
Orange	$1,00 \leq i < 4,35$	moins d'un incident par semaine
Rouge	$4,35 \leq i$	plus d'un incident par semaine

Recommandations

Interprétations possibles d'une valeur verte	Action
Peu ou pas d'incidents concernent ce produit.	Continuer comme ça !
Peu ou pas d'incidents sont remontés au Service Desk.	Sensibiliser les utilisateurs du produit.
Peu ou pas d'incidents sont encodés par le Service Desk.	Demander au Service Desk d'encoder les incidents.
Le produit est obsolète et n'est plus utilisé.	Le faire supprimer de la liste des produits.
Les valeurs mesurées et/ou les calculs d'évaluation sont erronés.	Décrire l'erreur à la cellule AQL.
Autre ?	Contacteur la cellule AQL.
...	...
Interprétations possibles d'une valeur rouge	Action
Trop d'incidents concernent ce produit.	Mener des actions correctives prioritaires.
Trop d'incidents sont remontés par erreur au Service Desk.	Sensibiliser et/ou former les utilisateurs du produit.
Les valeurs mesurées et/ou les calculs d'évaluation sont erronés.	Décrire l'erreur à la cellule AQL.
Autre ?	Contacteur la cellule AQL.

Figure 31 : Extrait des tableaux d'interprétation "Fiabilité"

Premièrement, des tranches de valeurs sont définies, permettant de regrouper des valeurs avec des mêmes règles d'interprétation. Par convention au sein de l'organisme, un code couleur a été affecté à chaque tableau de règles, « Vert » étant théoriquement le résultat attendu et « Rouge » étant théoriquement un mauvais résultat.

Ensuite, pour chaque tranche de valeurs, un ensemble de règles est défini avec, pour chaque règle identifiée, une recommandation. Il est important d'essayer d'identifier toutes les causes possibles de cette valeur de l'indicateur. Par exemple, on pourrait se contenter d'un résultat « vert » sans se poser de questions, mais on peut voir dans l'exemple ci-dessus que cette approche n'est pas forcément pertinente : un faible nombre d'incidents par mois peut venir du fait que des incidents ne

sont pas remontés ou pas encodés, alors qu'ils existent bel et bien ! A l'inverse, un résultat « rouge » ne devra pas forcément indiquer que c'est l'équipe responsable du produit qui a mal fait son travail.

Afin d'être complet, les deux dernières interprétations possibles de chaque tableau seront toujours les mêmes, à savoir que l'équipe qualité (ici la cellule AQI) peut toujours avoir fait une erreur lors de l'évaluation ou qu'une interprétation possible n'a pas été envisagée, le tableau devant alors être complété.

Pour en terminer avec l'interprétation, il est essentiel que le lecteur d'un indicateur l'interprète en connaissance de cause et ait conscience qu'un indicateur ne se suffit pas à lui-même. C'est à partir d'un certain nombre d'indicateurs différents sur un même produit qu'on aura un *aperçu* de la qualité de celui-ci. Par exemple, si 3 indicateurs sont mesurés et que 2 sont en « rouge », il ne faut pas directement en déduire que le produit est mauvais également. Par contre, si 50 indicateurs sont mesurés et que 49 d'entre eux sont en « rouge », à ce moment, on peut raisonnablement penser qu'il existe au moins un problème sur le produit évalué.

5.6.2 Cible de l'évaluation

Un autre aspect très important à communiquer aux ressources humaines responsables des produits évalués est la cible de l'évaluation.

Il faut rassurer les gens sur le fait qu'on évalue des produits et non des ressources, en ajoutant à cela la communication sur l'interprétation des indicateurs expliquée plus haut.

Il n'est pas de la responsabilité d'une équipe qualité¹⁰ (ici la cellule AQI), d'évaluer des personnes. Lors de l'évaluation des ressources humaines, la hiérarchie pourra bien sûr se servir des indicateurs mais cela ne doit pas constituer la seule source d'informations pour réaliser l'évaluation.

A noter que MoCQA permet de supporter cet aspect grâce, notamment, à la modélisation des dérivations : celles-ci incluent les activités réalisées, ce qui permet de pointer les moments où il y a une perte de qualité, sans directement incriminer les ressources humaines qui en sont responsables.

5.6.3 Participation des intervenants

Lors de quelques présentations des premiers indicateurs de qualité au SPI, certaines ressources ont exprimé leurs craintes par rapport à la subjectivité des indicateurs.

En effet, l'un des arguments poussant l'utilisation de l'approche MoCQA pour mettre en place des indicateurs de qualité est que cela tend vers l'objectivité puisque les indicateurs sont basés sur des métriques et la manière de les acquérir est formalisée et accessible aux intéressés.

Cela rend plus objective l'évaluation d'un indicateur mais la crainte exprimée venait du choix des indicateurs qui a semblé très subjectif : pourquoi avoir commencé par la fiabilité ? Pourquoi mesurer le nombre moyen d'incidents par mois ? Etc.

¹⁰ En tout cas, pour ce qui est du SPI, où le déploiement est réalisé.

Cette crainte est tout à fait justifiée et vient du fait que les personnes la formulant n'avaient pas été impliquées dans la définition des objectifs de qualité. C'est pourquoi il est important de faire participer l'ensemble des intervenants, afin de favoriser une acceptation, au moins partielle, du choix des indicateurs. Cela va du chef de service à l'opérateur en passant par le développeur, l'analyste, l'architecte, le testeur, etc.

Il est également important de présenter les indicateurs comme étant un outil à destination de chacun : le développeur aura accès à des indicateurs lui permettant d'avoir une idée de la qualité de son code, etc. Ainsi, cette philosophie devrait faciliter l'acceptation par les ressources, une approche itérative comme MoCQA permettant d'intégrer les retours des intervenants applicatifs, ceci pour améliorer les indicateurs de qualité et ainsi provoquer un consensus.

5.6.4 Support de la hiérarchie

Bien que cela soit important pour toute activité d'un service, le support de la hiérarchie est primordial pour la mise en place d'indicateurs de qualité. En effet, le risque de résistance au changement étant élevé, voire inévitable, un support fort de la hiérarchie permet de montrer aux ressources que les indicateurs sont utiles et sont utilisés.

D'un autre côté, il faut que la hiérarchie rassure en démontrant qu'elle a compris la philosophie d'interprétation des indicateurs expliquée plus haut, ceci via une bonne communication.

5.6.5 Diffusion des résultats d'évaluation

Une autre crainte des ressources auxquelles les premiers indicateurs ont été présentés est la comparaison avec les collègues.

Cela dépend de la politique de l'organisme où est effectué le déploiement. Au SPI, il a été décidé que seules certaines ressources auraient accès à l'entièreté des indicateurs (typiquement : la hiérarchie). Un responsable de produit aura accès uniquement aux indicateurs concernant son produit, ce qui évitera d'éveiller la curiosité et de comparer ses résultats à ceux des autres équipes.

Quid des utilisateurs des produits ? La communication sur l'interprétation des indicateurs étant primordiale, on pourrait donner un accès aux indicateurs pour les utilisateurs. Au sein du SPI, il a été décidé que ce ne serait pas le cas car cela constitue un risque de donner des informations permettant de rejeter certains problèmes transversaux sur le service informatique. De plus, réaliser une communication efficace auprès de 2400 agents aurait un coût non négligeable et peut-être pas rentable. Cette décision est propre au SPI et pourra être différente dans un autre contexte : avant toute décision, il est important d'analyser les impacts possibles de la diffusion des résultats.

5.7 Réflexion sur la force des indicateurs

Une constatation également intéressante est apparue à la fin de la première itération, après définition du plan d'actions (lors de la phase d'exploitation). En analysant certains constats et les actions liées, on s'est rendu compte que ces problèmes étaient déjà connus.

Par exemple, le constat le plus critique qui a été fait lors de la première exploitation est que le SPI ne disposait pas d'une liste uniformisée des produits : chaque système (gestion des incidents, gestion des déploiements, sécurité, ...) avait sa propre liste de produits, ce qui générait des incohérences dans le tableau de résultats des indicateurs car il comportait des applications en double voire en triple (avec des noms différents), des applications non présentes et des applications obsolètes. Ce problème avait déjà été évoqué de nombreuses fois lors des réunions de coordination du PAD mais aucune action concrète n'avait été réalisée pour pallier à ce problème. Grâce à la phase d'exploitation de l'approche MoCQA, une action d'uniformisation des codes produits a été ajoutée au plan d'action, avec un responsable assigné, ce qui donne beaucoup plus de poids au besoin que de simples discussions en réunion.

A noter que le SPI utilise l'outil Atlassian Jira pour le suivi des tâches, ce qui ajoute au poids des actions, plus qu'un fichier Excel ou équivalent qui peut se perdre ou vite devenir obsolète.

La force des indicateurs est également que tant qu'on n'a pas apporté des actions correctives, étant donné qu'on réitère des évaluations régulièrement, l'indicateur montrera toujours le problème. Il faudra donc toujours bien s'en occuper à un moment ou à un autre.

Un autre constat portait sur les résultats de certains produits. La hiérarchie avait évidemment une idée des produits posant problème mais n'avait pas ou peu de critères objectifs pour en juger. Les résultats sur ces produits étaient dans le « rouge » ce qui, rappelons-le, n'est pas toujours synonyme de problèmes mais ces résultats étaient compatibles avec l'idée que la hiérarchie s'était faite des produits « à problèmes ».

Dans ce cas, les indicateurs fixent un ressenti via des critères objectifs, ce qui constitue une autre force des indicateurs de qualité.

5.8 Participation à la validation

Comme toute théorie universitaire, il est intéressant de pouvoir démontrer la faisabilité de l'approche MoCQA, puisque toujours à l'état de développement au sein des FUNDP.

L'étudiant n'a pas la prétention de pouvoir valider ou pas l'approche MoCQA mais le déploiement dont il a été question ici participe à la validation.

Au niveau de la méthodologie MoCQA, elle s'est révélée applicable :

- les 5 étapes ont été suivies et n'ont pas révélé de manque,
- plusieurs actions avaient pour but de repasser par l'étape de modélisation,
- plusieurs actions avaient pour but de repasser par l'étape d'opérationnalisation,
- le retour vers l'étape d'évaluation a été réalisé,
- l'identification du besoin d'un nouvel indicateur a eu lieu une fois,
- seul le déclassement d'un indicateur n'a pas eu lieu (mais on peut penser sans risque que cela ne posera pas de problèmes).

Au niveau du méta-modèle MoCQA, il s'est révélé instanciable :

- l'ensemble des concepts a été instancié au moins une fois,
- la majorité des possibilités a été utilisée. Elles ne seront pas listées ici car trop nombreuses mais on peut citer quelques possibilités identifiées qui n'ont pas été utilisées :
 - la notion de version dans le type de dérivation et le type d'artéfact,
 - la notion de maturité dans le type d'artéfact,
 - le fait que plusieurs types d'artéfact soient la source ou la cible d'un seul type de dérivation,
 - les types de dérivation horizontale et les types de dérivation endogène,
 - les comportements fonctionnels,
 - la notion de statut qui est resté continuellement en « expérimentation »,
 - la possibilité de définir plusieurs méthodes ou plusieurs fonctions pour un seul attribut et
 - la possibilité qu'un objectif de qualité soit une entrée de plusieurs modèles d'évaluation différents.

Autre élément fournissant une aide à la validation, il faut garder à l'esprit que l'approche MoCQA est déployée au sein du SPI et les activités de gestion et de développement des indicateurs vont continuer à être réalisées au-delà du présent mémoire. Les indicateurs mis en place ont déjà permis d'identifier des problèmes de qualité qui ont été suivis jusqu'à leur résolution.

Des communications ont été effectuées actuellement auprès d'une quinzaine de ressources humaines et les concepts ont été compris.

La question peut se poser de savoir en quoi la mise en place d'indicateurs de qualité avec l'approche MoCQA peut garantir qu'on atteint l'objectif ultime de qualité. Effectivement, la garantie n'existe pas (puisque les indicateurs n'amènent que des recommandations d'actions correctives) mais il en va de même avec les autres modèles de qualité.

On pourrait également poser la question de la pertinence des indicateurs, MoCQA prônant la définition d'indicateurs « sur mesure » contrairement à la majorité des modèles existants : à nouveau, aucune garantie de pertinence des indicateurs n'est donnée. De même, les autres modèles ne donnent aucune garantie, sauf pour les normes ISO qui pourraient être considérées comme telles si on a confiance en la notion de certification.

On peut remarquer que les indicateurs mis en place dans les premières itérations n'auraient probablement pas pu être évalués au moyen d'un des modèles de qualité cités au chapitre 1, ce qui tend à confirmer que la philosophie d'adaptation à l'environnement est appropriée.

Pour obtenir une validation, d'autres expériences du type de celle-ci dans d'autres environnements devraient être réalisées. On pense à des entreprises plus petites, où les moyens accordés pour la qualité sont peut-être moins importants. On pense également à des entreprises du secteur privé où les enjeux sont différents.

Enfin, l'une des facettes de l'approche MoCQA était de pouvoir être appliquée dans tout environnement, même vierge de toute méthode de gestion du cycle de vie (pas d'analyse, pas d'architecture, etc.). Bien que ce soit théoriquement possible, l'étudiant émet des doutes sur la faisabilité pratique dans ce type d'environnement, le raisonnement étant le suivant : si l'entreprise ne met pas de moyens en œuvre pour définir des méthodes de gestion du cycle de vie, il y a fort à parier qu'elle ne mettra pas de moyens en œuvre pour la définition d'indicateurs de qualité.

Chapitre 6 : Futurs sujets de recherche

Tout au long du projet, certains points ont été soulevés mais n'ont pas été étudiés afin de rester concentré sur les objectifs du présent mémoire. Ces points sont repris ici : ils sont de complexité variable et certains pourraient donner lieu à des futurs sujets de mémoire, tandis que d'autres seront étudiés par le SPI directement.

6.1 Déploiement dans d'autres environnements

Le présent déploiement concerne un organisme public de grande taille. Il serait intéressant de tenter l'expérience dans des entreprises privées et des environnements de plus petite taille, ceci afin de s'assurer de la généralité de l'approche MoCQA.

6.2 Utilisation du MoCQA Toolkit

Etant donnée l'indisponibilité du *MoCQA Toolkit* pour accomplir le présent déploiement, avoir un retour d'expérience en environnement professionnel sur celui-ci serait intéressant. Dans la continuité, le guide de déploiement en environnement professionnel pourrait être étendu avec des bonnes pratiques d'utilisation des outils.

De plus, il serait envisageable d'en faire l'utilisation au sein du SPI : un sujet de mémoire intéressant pour un étudiant serait d'implémenter le *MoCQA Toolkit* en tenant compte de l'existant.

A noter qu'en l'absence d'outils propriétaires, la modélisation s'effectue via l'outil Enterprise Architect, ce qui donne lieu à une utilisation un peu fastidieuse, l'outil n'étant pas dédié à l'approche MoCQA.

6.3 Intégration de la norme ISO/IEC 25000 via MoCQA

La question se pose au sein du SPI de la pertinence de se baser sur la norme ISO/IEC 25000 pour définir les indicateurs de qualité. Dans ce doute, le déploiement de l'approche MoCQA a permis de se rassurer puisque ne fermant pas la porte à l'intégration d'une norme ISO.

Un futur sujet de mémoire pourrait être d'intégrer cette norme via la méthodologie MoCQA (et son méta-modèle) au sein du SPI.

6.4 Indicateurs de qualité sur les « projets »

La cible des évaluations dont il est question dans le présent mémoire est le produit.

Il serait possible d'utiliser l'approche MoCQA, peut-être moyennant quelques adaptations mineures du méta-modèle, pour mettre en place des indicateurs de qualité sur des projets¹¹.

Ce sujet sera prochainement étudié au sein du SPI.

6.5 Automatisation et présentation des indicateurs

Afin d'éviter le surcoût d'une réévaluation régulière de manière manuelle, il est envisagé d'automatiser une partie des métriques. La question de la présentation des indicateurs via une interface accessible aux ressources (avec gestion de droits d'accès) est également posée.

Un projet va prochainement être proposé au SPI afin de répondre à ces besoins.

6.6 Mise en place d'une activité de contrôle qualité informatique

La cellule AQI réalise des activités d'assurance qualité via la gestion de méthodes et standards internes, l'expertise en matière de techniques de test et le support qualité informatique : ces trois domaines d'activité mettent en place les moyens permettant de s'assurer de la qualité.

Une activité de contrôle qualité permettrait de vérifier l'utilisation de ces moyens par les équipes du SPI : actuellement, le SPI ne réalise pas (ou très peu de manière non systématique) cette activité de contrôle qualité.

L'un des objectifs à moyen terme est de mettre en place cette activité, via la création d'une cellule « CQI » (Contrôle Qualité Informatique), sœur de la cellule AQI. Bien que cela ne suffise pas pour atteindre cet objectif¹², la mise en place des indicateurs de qualité constitue un premier pas dans ce sens. L'avantage de cette démarche est que, étant donné que la manière d'évaluer les indicateurs est consultable et que chaque intervenant va théoriquement être impliqué, les indicateurs devraient être mieux perçus et acceptés par les équipes dont le produit est évalué.

A cette fin, des indicateurs de qualité visant à évaluer le respect des méthodes et standards internes seront mis en place via l'approche MoCQA.

¹¹ Au sens PMBOK du terme.

¹² Cette mise en place demandera, entre autres, la mise en place d'une équipe, la définition de procédures, une communication adéquate aux ressources humaines, etc.

6.7 Indicateurs identifiant un besoin de ressources humaines

Une question intéressante qui a été soulevée est de pouvoir identifier qu'un manque de qualité vient d'un manque de ressources humaines dans une équipe. En effet, il arrive qu'on justifie le manque de qualité avec le fait qu'on n'a pas assez de ressources pour accomplir le travail.

Il serait intéressant de se pencher sur la question de savoir à quel moment le problème est résolu avec les ressources disponibles et à quel moment le problème ne peut être résolu qu'avec l'augmentation de ressources.

Ce sujet est proposé sans certitude de la possibilité d'y répondre.

6.8 Diffusion du guide de déploiement

Le guide de déploiement de l'approche MoCQA en environnement professionnel est actuellement rédigé en français uniquement. Afin de favoriser sa diffusion et son utilisation, une traduction vers l'anglais serait intéressante à réaliser.

Conclusion

Le présent mémoire porte sur le déploiement de l'approche MoCQA en environnement professionnel et vise trois objectifs.

Le premier objectif est une utilisation concrète de l'approche afin de s'assurer de l'applicabilité de celle-ci : la faisabilité a été montrée via le suivi de la méthodologie MoCQA et l'instanciation multiple de son méta-modèle. Le présent déploiement a permis de donner rapidement du feedback aux FUNDP, de répondre à certaines questions qui n'avaient pas encore été traitées (typologie des intervenants, notion temporelle, traduction en français, coût de déploiement,...) et d'amener des réflexions tout au long du projet (différence entre projet et produit, psychologie du déploiement, force des indicateurs,...). L'étudiant a amené un regard neuf, voire peut-être « naïf » étant donné sa méconnaissance initiale vis-à-vis du domaine de la métrologie, sur l'approche MoCQA en abordant les tâches de manière concrète, pratique, et orientée sur les résultats.

Toutefois, le méta-modèle MoCQA n'a pas été utilisé à 100% de ses possibilités mais, dans les cas réalisés, n'a pas montré de limite empêchant de modéliser une situation particulière.

Le second objectif est la rédaction d'un guide de déploiement à destination des professionnels de l'informatique. Ce guide est proposé en annexe 1 et propose une « vulgarisation » des concepts universitaires accompagnée de bonnes pratiques identifiées lors d'un déploiement.

Néanmoins, le déploiement décrit dans le présent mémoire s'est déroulé dans un organisme public de grande taille : on peut donc se demander si l'approche MoCQA est applicable au sein d'une entreprise privée et/ou une organisation de plus petite taille. Le guide de déploiement est donc naturellement orienté « grande entreprise », bien qu'un effort ait été fait lors de sa rédaction pour garder à l'esprit les entreprises de taille plus modeste.

Enfin, le dernier objectif est l'identification des possibilités d'extension et/ou de recherches concernant l'approche MoCQA. On propose ici un ensemble de sujets à étudier par l'organisme où a eu lieu le déploiement (automatisation, indicateurs de qualité projets,...) et quelques sujets de réflexion (utilisation du *MoCQA Toolkit*, intégration de la norme ISO 25000,...) pour les FUNDP, certains pouvant éventuellement faire l'objet d'un futur mémoire.

L'ensemble des pistes de recherche future n'a probablement pas été identifié mais les quelques réflexions proposées, si étudiées, permettront d'augmenter les connaissances et de tendre vers une validation de l'approche MoCQA. Cette approche étant composée de trois éléments principaux, seuls deux d'entre eux ont été utilisés (méthodologie et méta-modèle) : l'utilisation en environnement professionnel du kit d'outils MoCQA permettrait de s'assurer de l'entière applicabilité de l'approche MoCQA.

Sans avoir la prétention de pouvoir valider l'approche MoCQA sur base unique du présent déploiement, l'étudiant pense que l'approche a de bonnes raisons de l'être : ne fut-ce que parce que l'aventure ne s'arrête pas avec ce mémoire puisque l'approche MoCQA va continuer à être utilisée au sein de l'environnement professionnel dont il est question ici.

Références

Bibliographie

Victor R. Basili, Gianluigi Caldiera, H. Dieter Rombach: *The Goal Question Metric Approach*, 1994
[Basili, Caldiera & Rombach 1994]

Samuel Hanoteau: *Guide de déploiement de l'approche MoCQA en environnement professionnel*, 2012
[Hanoteau 2012]

ISO/IEC 9126-1: *Software engineering - Product quality - Part 1: Quality model*, 2001
[ISO/IEC 2001]

ISO/IEC 25000: *Software engineering - Software product Quality Requirements and Evaluation (SQuaRE) - Guide to SQuaRE*, 2005
[ISO/IEC 2005]

Tom Mens, Leandro Doctors, Naji Habra, Benoît Vanderose, Flora Kamseu: *Etudes empiriques sur l'évolution de la qualité logicielle*, 2011
[Mens, Doctors, Habra, Vanderose & Kamseu 2011]

Maryoly Ortega, María Pérez, Teresita Rojas: *Construction of a Systemic Quality Model for Evaluating a Software Product*, 2003
[Ortega, Pérez & Rojas 2003]

Project Management Institute: *Guide du corpus des connaissances en management de projet (Guide PMBOK)*, 2009
[PMI 2009]

Object Management Group: *OMG Unified Modeling Language (OMG UML) Version 2.4.1*, 2011
[UML 2011]

Benoît Vanderose: *Towards a Model-Centric Quality Assessment: the MoCQA approach*, 2010
[Vanderose 2010]

Benoît Vanderose, Flora Kamseu, Naji Habra: *Towards a Model-Centric Quality Assessment*, 2010
[Vanderose, Kamseu & Habra 2010]

Benoît Vanderose, Naji Habra, Flora Kamseu: *Operationalization of a Model-Centric Quality Assessment (MoCQA) framework*, 2011
[Vanderose, Habra & Kamseu 2011]

Webographie

Service Public de Wallonie : <http://spw.wallonie.be/>

Portail de la Wallonie : <http://www.wallonie.be/>

Références internes DGARNE

Les références suivantes ne sont pas accessibles au public mais sont néanmoins citées par respect pour leurs auteurs.

Claude Delbeuck: *Manuel de management DGO3*, Document interne DGARNE, 2011

Intranet du Service Public de Wallonie

Intranet DGARNE

Intranet Confluence Informatique D443

Annexe 1 : Guide de déploiement

Facultés Universitaires Notre-Dame de la Paix, Namur
Faculté d'informatique

Approche MoCQA

Guide de déploiement en environnement professionnel

Samuel Hanoteau

Version 1.0 du 14/05/2012

1 Introduction

1.1 Objectif

L'approche MoCQA (*Model-Centric Quality Assessment*) propose un cadre générique pour la mise en place de modèles de qualité produisant des indicateurs de qualité sur les produits logiciels.

Etant développée au sein des Facultés Universitaires Notre-Dame de la Paix de Namur, l'approche a fait l'objet de plusieurs expérimentations, dont une en environnement professionnel. Ce déploiement hors université a rapidement identifié le besoin d'une approche plus pragmatique par rapport aux théories universitaires : une sorte de « vulgarisation » des concepts a ainsi donné naissance au présent guide.

L'objectif de ce guide est donc d'expliquer aux professionnels de l'informatique, désireux de mettre en place des indicateurs de qualité sur leurs produits logiciels, ce qu'est et ce que propose l'approche MoCQA en y joignant des exemples concrets d'utilisation et des bonnes pratiques de déploiement, fruits de l'expérience acquise par l'auteur.

1.2 Plan de lecture

Le présent guide de déploiement est structuré en différentes sections, dont un résumé est proposé ci-dessous. Le lecteur souhaitant comprendre l'approche MoCQA pourra aborder ces sections de manière séquentielle. Le guide se veut également un outil de référence dont une section particulière pourra être consultée en cas de besoin lors d'un déploiement.

La section *Contenu* vise à structurer le document via une table des matières et une table des illustrations et à donner une signification aux abréviations utilisées dans le document.

La section *Indicateurs de qualité avec MoCQA* vise à présenter l'approche d'une manière générale ainsi que sa finalité.

Les sections *Acquisition*, *Modélisation*, *Opérationnalisation*, *Evaluation* et *Exploitation* visent à détailler chaque phase spécifique de la méthodologie MoCQA.

La section *Communication* aborde l'aspect de la communication autour du déploiement des indicateurs de qualité, qui peut être très important en fonction de l'environnement où a lieu le déploiement.

Enfin, la *Conclusion* vise à faire le point sur le contenu du présent guide.

1.3 Prérequis à la lecture

Afin de pouvoir comprendre et instancier le méta-modèle MoCQA, une connaissance du formalisme des diagrammes de classes et des diagrammes d'objets UML est nécessaire.

La notation BPMN est utilisée pour représenter le processus principal mais la connaissance de celle-ci constitue uniquement un plus, la notation étant reconnue assez intuitive.

1.4 Illustration par un cas réel

Pour les chapitres 4 à 8 décrivant les différentes phases de la mise en place d'un indicateur de qualité, un cas réel sera utilisé pour illustrer la théorie énoncée.

Ce cas est tiré de l'expérimentation réalisée par l'auteur au sein d'un service informatique d'environ 70 personnes réalisant et maintenant des produits pour environ 2400 utilisateurs. Il est donc possible que les descriptions soient très orientées « grande entreprise » : l'auteur a toutefois porté attention à rester générique dans ses explications.

Afin de bien l'identifier, on prend comme convention que l'exemple est présenté en encadré grisé, précédé de la mention « Exemple » en gras souligné, comme illustré ci-dessous.

Exemple : ceci est un cas réel.

1.5 Références

Sont reprises ci-dessous les références ayant servi à l'élaboration du présent guide, ainsi que les références liées à l'approche MoCQA.

- *The Goal Question Metric Approach*, Victor R. Basili, Gianluigi Caldiera, H. Dieter Rombach, 1994
- *ISO/IEC 9126-1: Software engineering - Product quality - Part 1: Quality model*, ISO/IEC, 2001
- *Towards a Model-Centric Quality Assessment: the MoCQA approach*, Benoît Vanderose, 2010
- *Towards a Model-Centric Quality Assessment*, Benoît Vanderose, Flora Kamseu, Naji Habra, 2010
- *OMG Unified Modeling Language (OMG UML) Version 2.4.1*, Object Management Group, 2011
- *Business Process Model and Notation (BPMN) Version 2.0*, Object Management Group, 2011
- *Operationalization of a Model-Centric Quality Assessment (MoCQA) framework*, Benoît Vanderose, Naji Habra, Flora Kamseu, 2011
- *Etudes empiriques sur l'évolution de la qualité logicielle*, Tom Mens, Leandro Doctors, Naji Habra, Benoît Vanderose, Flora Kamseu, 2011
- *Déploiement de l'approche MoCQA en environnement professionnel*, Samuel Hanoteau, 2012

2 Contenu

2.1 Table des matières

1 Introduction	2
1.1 Objectif.....	2
1.2 Plan de lecture	2
1.3 Prérequis à la lecture	2
1.4 Illustration par un cas réel	3
1.5 Références	3
2 Contenu.....	4
2.1 Table des matières.....	4
2.2 Table des illustrations	5
2.3 Abréviations	6
3 Indicateurs de qualité avec MoCQA.....	7
3.1 Indicateurs qualité produits.....	7
3.2 Maturité du cycle de vie logiciel	9
3.3 Support hiérarchique	9
3.4 Approche MoCQA	10
3.5 Par où commencer ?	11
4 Acquisition.....	12
4.1 Présentation de la démarche.....	12
4.2 Identification des objectifs de qualité.....	13
4.3 Identification de l'existant	13
4.4 Structuration des objectifs.....	13
4.5 Compte rendu de l'interview	14
4.6 Priorisation des objectifs.....	15
4.7 Le mot de la fin.....	15
5 Modélisation	16
5.1 Méta-modèle MoCQA.....	16
5.2 Modélisation de la structure des objectifs et des indicateurs.....	23
5.3 Modélisation des entités mesurables	24
5.4 Modélisation des attributs.....	25
5.5 Finalisation du modèle.....	26
5.6 Réflexion au sujet des entités mesurables	29
5.7 Le mot de la fin.....	29

6	Opérationnalisation	30
6.1	Méthodes et fonctions.....	30
6.2	Modèles d'évaluation	31
6.3	Le mot de la fin.....	31
7	Evaluation.....	32
7.1	Application du plan de mesure	32
7.2	Opportunité d'automatisation.....	32
7.3	Présentation des résultats	32
7.4	Accès aux résultats.....	33
7.5	Le mot de la fin.....	33
8	Exploitation	34
8.1	Qui exploite les résultats d'évaluation ?.....	34
8.2	Procédure d'exploitation	34
8.3	Importance du plan d'actions	35
8.4	Le mot de la fin.....	35
9	Communication.....	36
10	Conclusion.....	37

2.2 Table des illustrations

Figure 1	: Processus général de gestion d'un indicateur de qualité	11
Figure 2	: Représentation de la phase d'acquisition.....	15
Figure 3	: Exemple simple de modèle	16
Figure 4	: Exemple simple de méta-modèle	17
Figure 5	: Méta-modèle MoCQA	18
Figure 6	: Modèle "Fiabilité": structure des objectifs de qualité.....	23
Figure 7	: Modèle "Surveillance des incidents" - Indicateur de qualité	24
Figure 8	: Modèle "Surveillance des incidents" - Concepts orientés produit.....	24
Figure 9	: Modèle "Surveillance des incidents" - Concepts orientés mesures	26
Figure 10	: Modèle "Surveillance des incidents" - Concepts orientés qualité.....	27
Figure 11	: Modèle "Surveillance des incidents" - Règles d'interprétation.....	28
Figure 12	: Représentation de la phase de modélisation	29
Figure 13	: Représentation de la phase d'opérationnalisation.....	31
Figure 14	: Représentation de la phase d'évaluation	33
Figure 15	: Représentation de la phase d'exploitation	35

2.3 Abréviations

Abréviation	Signification
ALM	Application Lifecycle Management
BPMN	Business Process Model and Notation
FUNDP	Facultés Universitaires Notre-Dame de la Paix
MoCQA	Model-Centric Quality Assessment
SGIQ	Système de Gestion des Indicateurs Qualité
UML	Unified Modeling Language
XML	eXtensible Markup Language

3 Indicateurs de qualité avec MoCQA

Dans ce chapitre, on décrit ce que sont et la finalité des indicateurs de qualité sur les produits. Ensuite, une note sur la maturité du cycle de vie au sein de l'environnement professionnel cible les prérequis nécessaires pour commencer un déploiement MoCQA. On attire ensuite l'attention sur le besoin du support de la hiérarchie. Enfin, l'approche MoCQA est présentée de manière générale.

3.1 Indicateurs qualité produits

L'objectif de la démarche présentée dans ce guide est de mettre en place des indicateurs de qualité sur des produits logiciels que l'on appellera, par raccourci, des *indicateurs qualité produits*. Mais que sont-ils ? Prenons chaque terme séparément afin de pouvoir l'expliquer de manière efficace.

3.1.1 Qu'est-ce qu'un indicateur ?

Un indicateur est une donnée concrète pouvant être interprétée, ce qui donne généralement lieu à une action.

Prenons un exemple simple : on constate que de la fumée s'échappe par la porte d'une pièce. On en déduit qu'un incendie s'est déclaré et on appelle les pompiers. Cet exemple illustre très bien la définition donnée ci-dessus puisqu'on retrouve chaque concept énoncé :

Il y a de la fumée
(indicateur)
→
Il y a un incendie
(interprétation)
→
On appelle les pompiers
(action)

Ici, la fumée est une donnée concrète : on peut la voir. A l'inverse, l'incendie ici est uniquement une interprétation de l'indicateur de fumée qui a donné lieu à l'appel des pompiers. Mais la fumée indique-t-elle forcément qu'un feu s'est déclaré ? Non : ce pourrait être un fumigène, de la vapeur d'eau, etc. Cela attire l'attention sur une première leçon à retenir : **un indicateur peut être interprété de multiples façons !**

De même, un incendie est-il uniquement déterminé par la présence de fumée ? A nouveau, la réponse est non : il faut qu'il y ait des flammes, de la chaleur, etc. C'est ainsi qu'on identifie une seconde leçon à retenir : **un indicateur ne se suffit pas à lui-même !**

Evidemment, en fonction de l'interprétation faite d'un indicateur, l'action sera différente.

3.1.2 Qu'est-ce que la qualité ?

Sans doute la question à laquelle il est le plus difficile d'apporter une réponse : si on interrogeait dix personnes sur le sujet, il est probable qu'on obtienne dix définitions différentes. On préférera une définition la plus générique possible : **la qualité est l'ensemble des moyens mis en œuvre visant à satisfaire les intervenants**. On comprend ici par intervenant toute personne ayant une implication avec l'objet dont on évalue la qualité : la satisfaction de l'ensemble des intervenants est donc chose rare, voire parfois contradictoire.

Prenons l'exemple du développement d'un logiciel où les intervenants peuvent être très nombreux : on pense bien sûr à l'utilisateur final dont la satisfaction est primordiale mais on doit également penser aux développeurs, aux analystes, au chef de projet, au responsable de la future maintenance, à la hiérarchie, au commanditaire, aux cellules de contrôle, etc.

Au vu du nombre potentiellement élevé d'intervenants, on peut facilement se rendre compte que la perception de la qualité par chacun d'entre eux sera différente. Le commanditaire voudra que son logiciel fonctionne et permette aux utilisateurs d'être plus productif, l'utilisateur voudra un logiciel l'aidant dans son travail et agréable à utiliser, le développeur voudra un code propre respectant les bonnes pratiques d'ingénierie du logiciel, la hiérarchie voudra que le client soit content et que le coût soit minimal, etc.

C'est là qu'on identifie une nouvelle leçon : **chaque intervenant a une perception de la qualité qui lui est propre.**

3.1.3 Qu'est-ce qu'un produit ?

Un produit est une entité logicielle ayant un cycle de vie propre. Il s'agit donc naturellement des applications développées mais aussi des éventuels frameworks propriétaires, des services et composants distribués,..., ainsi que des logiciels dont l'entreprise a fait l'acquisition. On constate donc que la cible sur laquelle porte une évaluation peut très vite devenir assez large.

Un produit est composé de différentes entités mesurables, à savoir des éléments sur lesquels au moins un attribut peut être mesuré. Elles peuvent être de trois types différents :

- Les *artéfacts*, qui constituent généralement la majorité des entités mesurables, sont les éléments du produit. Par exemple, le code source, l'analyse, la documentation, l'exécutable, la batterie de tests,... sont autant d'artéfacts d'un produit.
- Les *comportements* que l'on peut constater durant le cycle de vie du produit. On peut citer, par exemple, le calcul d'un taux de change ou un incident survenant sur le produit ou encore le contentement d'un utilisateur.
- Les *dérivations* qui sont les transformations d'un artéfact en un autre artéfact. Un exemple simple pourrait être la dérivation d'un artéfact « code source » vers un artéfact « exécutable » qui est tout simplement la « compilation ».

D'une entreprise à une autre, et d'un produit à un autre, les entités mesurables seront différentes (bien que probablement proches), ceci dépendant du cycle de vie mis en place et du niveau de standardisation de l'environnement.

3.1.4 Quelle est la finalité des indicateurs qualité produits ?

Dans la réalité, nous avons un « ressenti » personnel sur certains produits. Qui n'a jamais entendu et/ou dit lui-même « *Cette application est vraiment bien faite !* » ou « *Ce logiciel, c'est du n'importe quoi !* », etc. ? Souvent, un même produit pourra être ressenti différemment par des personnes différentes : citons par exemple la « guerre » des systèmes d'exploitation ayant lieu sur de nombreux forums, où les fans de l'un dénigrent l'autre et inversement.

Sur quoi ce ressenti se base-t-il ? Sur notre expérience personnelle, sur notre perception de la qualité, peut-être aussi sur notre humeur du jour... On peut clairement parler ici de *subjectivité*.

L'idée visée par la démarche présentée ici est de **mettre en place un ensemble d'indicateurs de qualité pour évaluer les produits sur base de métriques**. Comme on l'a vu plus haut, un indicateur ne se suffit pas à lui-même : il est donc plus qu'utile d'en définir un ensemble afin d'avoir un aperçu de la qualité, le résultat de cet aperçu n'étant pas garanti à 100% puisque tout indicateur peut être interprété différemment. Mais le fait de se baser sur des éléments mesurables (et donc beaucoup plus concrets qu'un simple ressenti) fait qu'on tend vers une certaine *objectivité* de l'évaluation.

Les indicateurs permettent de disposer d'une vue d'ensemble dynamique de la qualité d'un ou de plusieurs produit(s) et sont destinés à tous les types de profil qui les utiliseront comme un nouvel **outil**. La finalité n'est pas d'évaluer les ressources humaines associées au produit, mais le produit lui-même.

3.2 Maturité du cycle de vie logiciel

La mise en place d'indicateurs qualité produits sera plus ou moins simple en fonction du niveau de maturité du cycle de vie logiciel au sein de l'environnement visé. Prenons trois exemples fictifs pour illustrer ce point.

La société X faisant partie des leaders du développement de logiciels a un cycle de vie clairement défini : chaque activité est attribuée à un profil spécifique et a une entrée et une sortie bien définies. Recueil des exigences, analyse fonctionnelle, conception de l'architecture, développement, tests, déploiement, acceptation,... sont des étapes réalisées selon des règles formalisées au sein de l'entreprise, se basant sur des standards reconnus du marché.

La société Y est une petite entreprise effectuant, entre autres activités, du développement de logiciel. Elle a mis en place une équipe de professionnels de l'informatique qui « touchent à tout » : il n'est pas rare d'y voir la même personne réaliser chaque activité de manière itérative. Les ressources connaissent certaines bonnes pratiques du marché, qu'elles agrémentent de leur expérience personnelle.

Le particulier Z est une personne ayant une activité complémentaire visant à développer des logiciels pour des clients trouvés via le « bouche à oreille ». L'analyse des exigences se limite souvent à quelques notes sur une feuille : Z se lance généralement dans le développement au plus vite, ceci étant plus une passion qu'une source de revenus pour lui. La phase de tests est très limitée et sans doute réalisée par le client lui-même uniquement.

Ces trois cas ayant chacun leurs avantages et inconvénients que l'on ne jugera pas ici, la maturité de leur cycle de vie logiciel est clairement différente. La société X aura sans doute rapidement des résultats productifs sur la façon d'améliorer ses processus et de cibler plus objectivement les besoins de formation, les manques en ressources au sein de certaines équipes, etc. La société Y se rendra peut-être compte que les indicateurs donnent des résultats assez inégaux d'un produit à un autre et que, donc, standardiser son cycle de vie serait plus productif. Enfin, bien qu'il soit probable qu'il ne fasse jamais la démarche de mettre en place des indicateurs de qualité, Z se rendra compte, s'il le fait, que respecter des bonnes pratiques reconnues n'est pas une perte de temps.

Ces trois exemples permettent d'annoncer un constat : il est possible que les premiers résultats donnés par les indicateurs de qualité identifient des manques organisationnels globaux liés au cycle de vie logiciel, plutôt que des défauts isolés. Ceci est tout à fait normal et ne doit pas rebuter au début : ces manques doivent, au contraire, être traités avec attention, ce qui implique le besoin d'un support hiérarchique, comme indiqué ci-après.

3.3 Support hiérarchique

La mise en place d'indicateurs qualité produits peut engendrer de la résistance au changement, plus ou moins forte en fonction de la taille et de la politique de l'entreprise, mais aussi en fonction de la manière de communiquer autour de cette activité. Bien qu'on réalise des évaluations sur des *produits*, il y a des ressources humaines liées à la création de ces produits : développeurs, analystes, chef de projet,... sont autant de personnes pouvant se sentir visées par ces évaluations, ce qui risque de constituer plus un frein qu'un moteur pour l'exploitation des indicateurs.

Il est donc primordial, particulièrement dans les grandes entreprises, de bénéficier d'un support fort de la hiérarchie. D'un côté, celle-ci doit affirmer qu'elle promeut l'activité qui est réalisée à sa demande. D'un autre côté, la hiérarchie doit également rassurer sur le fait qu'elle est consciente de la finalité des indicateurs et que l'évaluation des ressources humaines ne sera pas réalisée sur base unique des résultats d'évaluation des produits réalisés par ces ressources.

3.4 Approche MoCQA

L'approche MoCQA (*Model-Centric Quality Assessment*) est le fruit d'un projet de recherche au sein des Facultés Universitaires Notre-Dame de la Paix de Namur. Comme son nom l'indique, cette approche vise l'évaluation de la qualité centrée sur les modèles.

3.4.1 Qu'est-ce qu'un modèle de qualité ?

La norme ISO/IEC 9126 définit un modèle de qualité comme étant **l'ensemble des caractéristiques, et des relations entre elles, fournissant les bases pour la spécification des exigences de qualité et l'évaluation de la qualité**. Un modèle de qualité fournit donc des spécifications permettant à l'équipe de développement d'appréhender le niveau de qualité exigé des livrables qu'elle produit et à l'évaluateur de la qualité de disposer d'un support dans son travail d'évaluation.

L'approche MoCQA vise à créer des modèles de qualité adaptés à l'environnement de l'entreprise.

3.4.2 Que propose MoCQA ?

L'approche MoCQA est composée de trois éléments principaux :

- Un **méta-modèle** qui permettra, une fois instancié, de créer les modèles de qualité (appelés aussi *modèles MoCQA*) : ce méta-modèle sera expliqué plus loin dans le chapitre sur la modélisation.
- Une **méthodologie** qui guide le qualiticien dans sa démarche de création et de gestion des indicateurs qualité produits et qui est le sujet principal du présent guide de déploiement.
- Un **kit d'outils** dont il ne sera pas question ici, le kit n'étant pas assez abouti et donc non exploitable au moment de la rédaction du présent document.

La particularité de l'approche MoCQA est que les modèles de qualité, comme ils sont réalisés « sur mesure », s'adaptent à l'environnement dans lequel ils sont réalisés, contrairement aux modèles de qualité « prédéfinis » (ISO 9126, ISO 25000, FURPS,...). A noter que le méta-modèle MoCQA a été étudié pour permettre l'intégration de ces modèles de qualité « prédéfinis », de sorte qu'utiliser l'approche MoCQA ne ferme aucune porte par rapport aux modèles existants du marché.

3.4.3 Le cycle de vie d'un indicateur de qualité

La méthodologie MoCQA définit 5 phases successives formant un cycle, lui-même itératif, pour le cycle de vie d'un indicateur de qualité.

Le processus porte sur un produit en particulier ou sur un ensemble de produits dans le cas où on souhaite mettre en place des indicateurs transversaux à tous les produits, comme ce sera souvent le cas dans les grandes entreprises où une philosophie ALM (*Application Lifecycle Management*) est adoptée.

La figure ci-dessous montre le processus général de gestion d'un indicateur.

On peut y voir les 5 phases successives :

- l'acquisition qui vise à identifier l'objectif de qualité à atteindre,
- la modélisation qui vise à créer un modèle de qualité (le « quoi »),
- l'opérationnalisation qui vise à créer un plan de mesures lié au modèle de qualité (le « comment »),
- l'évaluation qui vise à appliquer le modèle de qualité et le plan de mesures pour donner des résultats et
- l'exploitation qui vise à étudier les résultats de l'évaluation et à émettre des constatations.

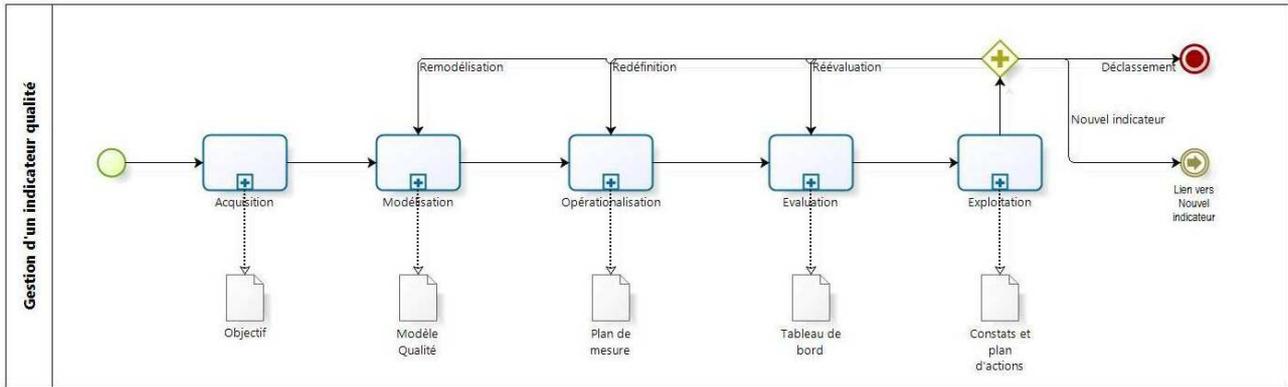


Figure 1 : Processus général de gestion d'un indicateur de qualité

La dernière étape (l'exploitation) donne lieu à un plan d'actions (selon l'interprétation des indicateurs), chaque action pouvant concerner le modèle de qualité ou le plan de mesures, demander une réévaluation, identifier le besoin d'un nouvel indicateur (auquel cas une nouvelle instance du processus doit être démarrée), voire à déclasser l'indicateur.

Chaque phase est décrite en détails dans les 5 chapitres à venir.

Il est important à ce stade de garder à l'esprit que chaque indicateur aura son propre cycle de vie. Il faudra donc maintenir une certaine activité sur chaque indicateur au fur et à mesure qu'ils sont mis en place.

3.5 Par où commencer ?

Le déploiement de l'approche MoCQA dans une entreprise demande, comme toute nouvelle activité, de réaliser les choses dans le bon ordre.

Tout d'abord, il est essentiel, comme expliqué à la section 3.3, que la personne ou l'équipe chargée du déploiement soit missionnée clairement par la hiérarchie (à l'exception des personnes qui sont leur propre patron) et que la hiérarchie soit informée au moins des grandes lignes de la démarche.

Ensuite il vaut mieux commencer « petit » et procéder par itérations progressives.

La première phase d'acquisition vise à interviewer les intervenants. On l'a vu, le nombre d'intervenants peut rapidement devenir assez conséquent. Si c'est le cas, mieux vaut sélectionner un échantillon d'intervenants et, si possible, choisir des personnes n'ayant pas un a priori négatif sur le domaine de la qualité en général. Ici, deux approches différentes sont possibles : soit on sélectionne des intervenants plutôt basés sur l'organisationnel (exemples : chef de service, responsable qualité,...), soit on sélectionne des intervenants ayant un travail plus concret sur les produits (exemples : développeurs, analystes,...).

De même, il faut déterminer sur quoi va porter la première évaluation. Si on est dans une logique ALM, on voudra des indicateurs sur l'ensemble des produits de l'entreprise : mieux vaut, dans une première itération, se pencher sur un échantillon de produits seulement. Si on n'est pas dans une logique ALM, on peut se contenter d'un seul produit dans un premier temps.

4 Acquisition

La première phase consiste à identifier les objectifs de qualité à atteindre pour les produits logiciels à évaluer.

Si ce n'est déjà fait, on commence par identifier les intervenants ayant un intérêt dans la qualité des produits logiciels. A noter qu'il existe deux types d'intervenants :

- les **intervenants applicatifs**, dont les besoins sont, ou peuvent être, différents d'un produit à un autre et
- les **intervenants génériques**, dont les besoins sont identiques d'un produit à un autre.

Une fois la liste des intervenants établie, on sollicite une interview auprès de chacun. Le mieux est d'interviewer séparément chaque intervenant, sauf si le ou les objectif(s) de qualité à atteindre doivent être obligatoirement communs.

Exemple : le cas illustré se déroulant dans un service informatique très structuré ayant une logique ALM, un échantillon de cinq intervenants a été sélectionné : le responsable du développement, le responsable de l'appui au développement, le responsable du service desk, le responsable de l'architecture et le responsable de l'assurance qualité.

Bien que la méthodologie MoCQA n'oblige rien à ce niveau, on conseille que l'interview d'un intervenant se déroule comme suit.

4.1 Présentation de la démarche

Il semble essentiel de commencer l'interview par une sensibilisation à la démarche : présenter le contexte dans lequel les indicateurs vont être utilisés, ce sur quoi ils portent et surtout l'utilité que ces indicateurs devraient avoir pour l'intervenant.

Si l'intervenant a déjà été interviewé dans le cadre d'une autre itération, on peut bien sûr alléger voire omettre cette étape.

4.2 Identification des objectifs de qualité

L'idée est ici que l'intervenant réponde à la question « Qu'est-ce qui fait qu'un produit est de qualité pour moi vis-à-vis de mon métier ? ». Dépendant de la personne interviewée, cette étape peut s'avérer difficile. Si tel est le cas, le qualitatif peut aiguiller en donnant des exemples ou en supposant les objectifs de la personne mais il faut toutefois éviter de répondre à la place de l'intervenant : il s'agit de ses objectifs et non de ceux du qualitatif (qui pourra, soit dit en passant, « s'auto-interviewer » à un moment donné puisqu'il est lui-même un intervenant).

Exemple : le responsable du service desk interviewé a identifié 3 objectifs de qualité liés à son travail, à savoir la fiabilité des produits, la performance des produits et la satisfaction des utilisateurs. Quand l'un de ces objectifs n'est pas atteint pour un produit, l'utilisateur contacte directement le service desk qui est la première ligne de front pour subir le mécontentement d'un utilisateur : ces objectifs sont donc parfaitement en adéquation avec le profil métier de l'intervenant.

Comme l'illustre l'exemple ci-dessus, un objectif doit être lié au métier de l'intervenant mais ce n'est pas pour autant un objectif que l'intervenant doit atteindre lui-même : ce sont des objectifs pour les produits.

Il est également intéressant de demander à l'intervenant de fixer des priorités au sein de ses objectifs.

4.3 Identification de l'existant

Il se peut que l'entreprise, ou l'intervenant de sa propre initiative, dispose déjà de certaines métriques, développées ou accessibles via un outil particulier. Il est important de les identifier dans le but de pouvoir éventuellement les réutiliser.

Exemple : ici, le responsable du service desk a mentionné qu'un outil de gestion permettait d'encoder des incidents (à savoir les plaintes des utilisateurs concernant un produit particulier) : cet outil, appelé *TrackIT!*, pourra donc constituer une source éventuelle pour la mesure de certains attributs.

4.4 Structuration des objectifs

L'exercice est ici de structurer chaque objectif en sous-objectifs si besoin. En effet, certains objectifs sont parfois trop généralistes pour pouvoir être ciblés concrètement.

Exemple : on a demandé au responsable du service desk ce qu'il entendait par *Fiabilité* et rapidement, quatre sous-objectifs sont apparus : la *survenance des incidents*, la *disponibilité*, l'*impact des incidents* et la *fréquence de déploiement*. Ces sous-objectifs devenaient évidemment plus concrets et, à ce stade, on entrevoit la façon de les évaluer.

A noter qu'une structuration à plus de deux niveaux est possible si le contexte le demande.

4.5 Compte rendu de l'interview

Après l'interview, il est important de consigner les informations recueillies dans un compte rendu, le canevas important peu. Néanmoins, la constitution d'un canevas particulier est une bonne idée car il peut servir comme guide lors de l'interview.

Exemple : ci-dessous un extrait du compte rendu de l'interview du responsable du service desk reprenant les informations transversales et le premier objectif de qualité.

Informations transversales	
Identifiant	DESK
Date de l'interview	19/08/2011
Intervenant	Pierre Girboux
Fonction	Responsable du Service Desk
Type d'intervenant	Générique
Validation	Validé (22/08/2011)
Remarques additionnelles	Non

Objectifs qualité	
Fiabilité	
Identifiant	DESK_01
Priorité	1
Description	<p>Les solutions logicielles livrées doivent être fiables et stables. Voici les sous-objectifs identifiés pour cet objectif:</p> <ul style="list-style-type: none"> • Survenance des incidents (un nombre élevé d'incidents peut indiquer un manque de fiabilité) • Disponibilité (une application doit être disponible pour ses utilisateurs au maximum de temps) • Impact des incidents (un incident doit au minimum impacter la production) • Fréquence de déploiement (un produit déployé fréquemment en production n'est pas rassurant)
Standards adoptés	Non
Parties concernées	<ul style="list-style-type: none"> • Livrables
Mesures existantes	<ul style="list-style-type: none"> • Nombre d'incidents liés à une application. <ul style="list-style-type: none"> • Oui: via l'outil Track-It • Niveau de confiance: élevé • Evaluation: à la demande
Entités mesurables	<ul style="list-style-type: none"> • Taux de disponibilité moyen de l'application (pourcentage) • Temps d'indisponibilité moyen après incident (nombre entier, unité: minute) • Nombre de déploiements moyen (nombre entier, unité: déploiement/mois)

Le compte rendu est envoyé à l'intervenant en lui demandant une validation des objectifs identifiés (dans un délai raisonnable, généralement annoncé).

L'intervenant doit ensuite valider les objectifs : selon la politique de l'entreprise, cette validation se fera de manière plus ou moins formelle.

4.6 Priorisation des objectifs

Chaque interview d'un intervenant identifie généralement plusieurs objectifs de qualité. Comme chaque objectif va théoriquement faire l'objet d'une modélisation MoCQA, il est important de fixer des priorités sur chaque objectif pour déterminer lesquels seront étudiés en premier. Les critères de priorisation sont dépendants du contexte : besoin à court terme, complexité présumée d'une modélisation, métriques déjà existantes, etc. Lors des premières itérations, on recommande de sélectionner des objectifs qui semblent plus simples à modéliser.

La priorisation peut se faire soit via accord entre les intervenants ou par arbitrage direct par la hiérarchie.

Exemple : l'interview des cinq intervenants a donné une liste de 26 objectifs de qualité. L'objectif Fiabilité du service desk a reçu la priorité la plus forte pour deux raisons : l'outil *TrackIT!* semblait contenir toutes les informations nécessaires aux mesures et l'objectif semblait plus simple à modéliser.

4.7 Le mot de la fin

A la fin de cette phase d'acquisition, on a ainsi dégagé les principaux objectifs, éventuellement déjà structurés, comme l'indique la figure ci-dessous. On a normalement identifié un premier objectif sur lequel se pencher : on peut ainsi passer à l'étape de la modélisation MoCQA décrite dans le chapitre suivant.



Figure 2 : Représentation de la phase d'acquisition

5 Modélisation

La seconde phase vise à instancier le méta-modèle MoCQA pour créer un modèle de qualité relatif aux objectifs identifiés dans la phase d'acquisition.

Concrètement, pour des raisons de lisibilité, on peut choisir de représenter un modèle MoCQA via plusieurs schémas différents, chaque schéma portant sur un « sous-modèle » particulier.

C'est ici qu'on déterminera :

- les indicateurs de qualité permettant d'évaluer l'atteinte des objectifs,
- les différentes entités mesurables sur lesquelles vont porter les mesures,
- les différents attributs qui devront être capturés ou calculés et,
- les règles d'interprétation des indicateurs de qualité.

Pour ce faire, il est important de comprendre le méta-modèle MoCQA et ce qu'il permet de faire. Ensuite, l'instanciation du méta-modèle se fait en 4 étapes. Enfin, on complète en définissant des règles d'interprétation.

5.1 Méta-modèle MoCQA

On aborde ici peut-être l'aspect le plus théorique : le méta-modèle MoCQA peut paraître rebutant au début mais on apprend vite à jongler avec ses concepts.

5.1.1 Qu'est-ce qu'un méta-modèle ?

Un méta-modèle, c'est un modèle d'un modèle. Le méta-modèle MoCQA est donc le modèle d'un modèle MoCQA. Le méta-modèle définit les concepts qui devront être présents dans le modèle résultant de son instanciation.

Un exemple étant plus parlant, prenons le cas d'un diagramme de classes UML (dont la connaissance du formalisme est l'un des prérequis à la compréhension de ce guide) :

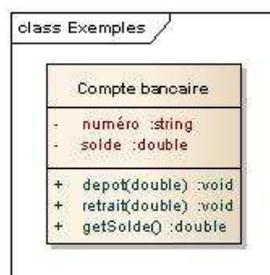


Figure 3 : Exemple simple de modèle

On peut y voir qu'un compte bancaire a deux attributs (numéro et solde) et trois opérations (depot, retrait et getSolde). Il s'agit d'un modèle au format d'un diagramme de classes UML.

Le méta-modèle (simplifié pour l'exemple) d'un diagramme de classes UML est proposé dans la figure ci-dessous. On peut y voir qu'une classe a un libellé, des attributs et des opérations. Les attributs sont constitués d'un libellé, d'un type et d'une visibilité, de même que les opérations qui prennent en plus en entrée des paramètres ayant eux-mêmes un libellé et un type.

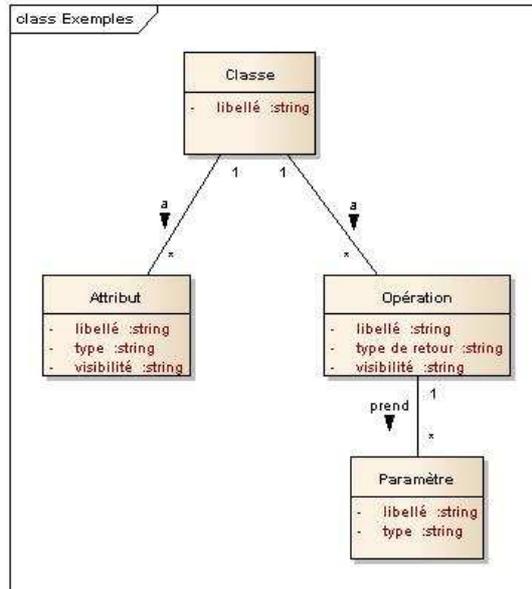


Figure 4 : Exemple simple de méta-modèle

5.1.2 Quel outil utiliser pour instancier le méta-modèle MoCQA ?

L'instanciation d'un méta-modèle peut se présenter sous différents formats. Le formalisme choisi ici pour représenter les modèles MoCQA est celui des diagrammes d'objets UML. A cette fin, l'outil *Enterprise Architect* a été utilisé pour instancier le méta-modèle mais tout outil permettant de créer ce type de diagramme fera probablement l'affaire. L'avantage de l'utilisation de ce type d'outil est qu'on peut créer de nouveaux objets via « cliquer-déposer » depuis le méta-modèle MoCQA vers le diagramme d'objets cible.

Pour rappel, un kit d'outils MoCQA est actuellement en cours de développement au sein des FUNDP et proposera un formalisme propriétaire, probablement plus adapté pour représenter un modèle de qualité MoCQA. Mais le formalisme des diagrammes d'objets UML permet d'instancier le méta-modèle sans limitation et de comprendre les concepts MoCQA présentés ici.

5.1.3 Présentation du méta-modèle MoCQA

Le méta-modèle MoCQA est représenté dans la figure ci-dessous.

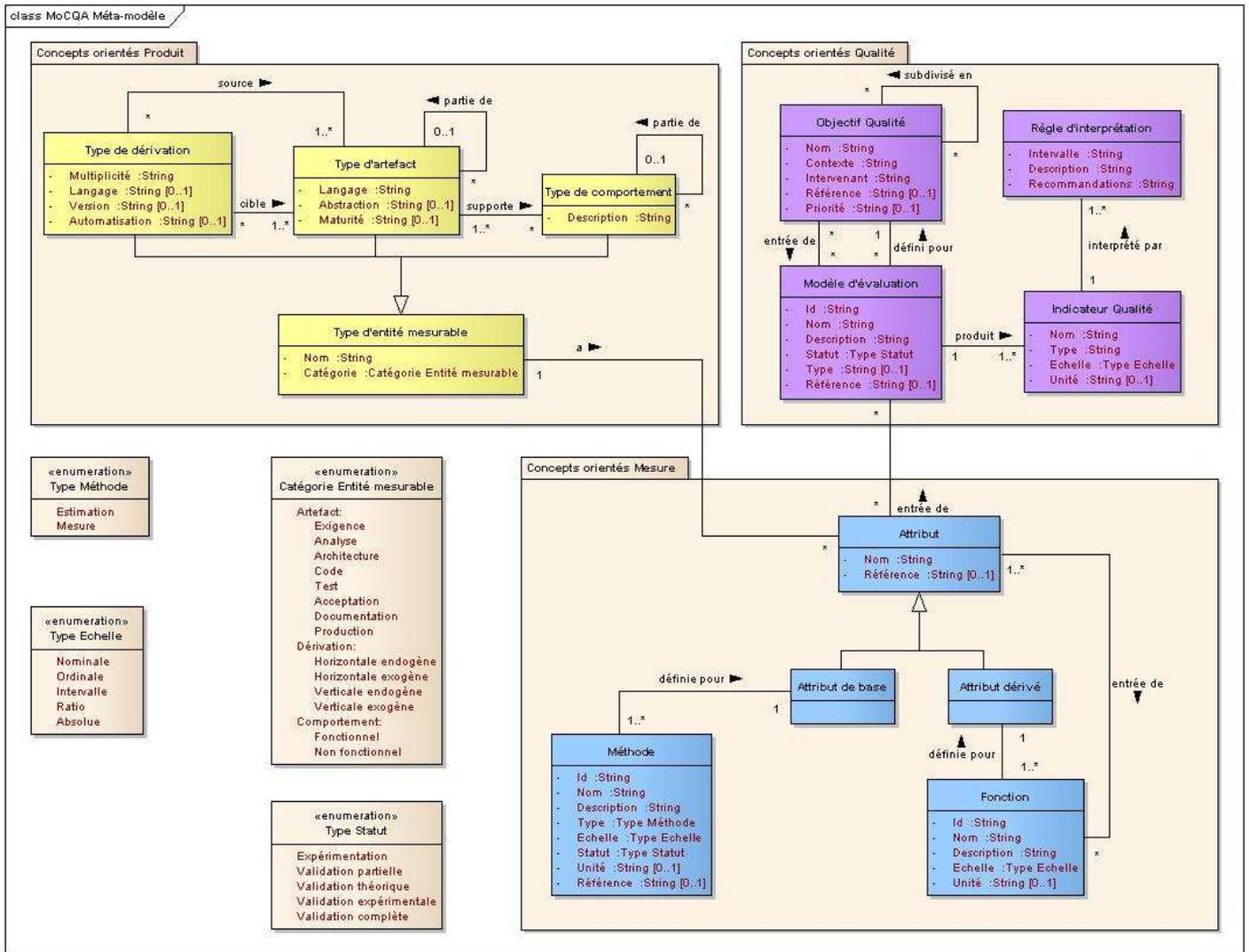


Figure 5 : Méta-modèle MoCQA

Le méta-modèle MoCQA est structuré en trois parties distinctes reliées entre elles :

- les concepts orientés Qualité (représentés en mauve),
- les concepts orientés Mesure (représentés en bleu) et
- les concepts orientés Produit (représentés en jaune).

Les codes couleurs présentés ici ont été établis dans l'unique but de faciliter la lecture des futurs modèles de qualité, chaque objet instancié du méta-modèle conservant la couleur de son type.

On présente également des énumérations représentant les valeurs possibles pour certains types.

Chaque concept est présenté dans les sections ci-après.

5.1.4 Concepts orientés Qualité

Objectif qualité	
Définition :	Un objectif (de) qualité est un facteur ou une caractéristique de qualité, généralement haut niveau, pour lequel un intervenant a un intérêt.
Attributs :	Nom : Nom reflétant de manière adéquate le concept qualité qu'il concerne.
	Contexte : Définit la cible de l'évaluation, à savoir les éléments pour lesquels l'évaluation de l'objectif qualité est pertinente (exemples : produit, application métier, progiciel,...).
	Intervenant : Personne ou groupe de personnes intéressé par cet objectif qualité (exemple : hiérarchie, développeur, équipe spécifique,...).
	Référence : Référence (bibliographie, site web,...) dont l'objectif qualité est inspiré.
	Priorité : Niveau de priorité de l'objectif qualité pour l'intervenant.
Relations :	Subdivisé en : Un objectif qualité peut être subdivisé en plusieurs objectifs qualité.
	Entrée de : Un objectif qualité peut être une entrée de modèles d'évaluation définis pour d'autres objectifs qualité.
Contraintes :	Si l'objectif qualité n'a pas de subdivision, alors il a forcément au moins un modèle d'évaluation qui le définit.

Modèle d'évaluation	
Définition :	Un modèle d'évaluation est une collection de déclarations (équations, descriptions de processus, algorithmes,...) utilisant les mesures ou estimations produites par différents attributs en vue de produire un ou plusieurs indicateur(s) qualité pour l'objectif liée.
Attributs :	Id : Identifiant permettant de pointer vers un fichier descriptif du modèle d'évaluation.
	Nom : Nom reflétant de manière adéquate le modèle d'évaluation qu'il concerne.
	Description : Description succincte du modèle d'évaluation.
	Statut : Indique le degré de confiance par rapport à ce modèle.
	Type : Type donné à titre informatif (non encore documenté par les FUNDP).
	Référence : Référence (bibliographie, site web,...) dont le modèle d'évaluation est inspiré.
Relations :	Produit : Un modèle d'évaluation produit au moins un indicateur qualité.
	Défini pour : Un modèle d'évaluation est défini pour un objectif qualité.
Contraintes :	Un modèle d'évaluation prend en entrée au minimum un objectif qualité ou un attribut.
	Si le modèle d'évaluation produit plusieurs indicateurs qualité, ceux-ci doivent avoir la même sémantique.

Indicateur qualité	
Définition :	Un indicateur (de) qualité est une donnée concrète résultant de l'application du modèle d'évaluation lié et représentant un objectif qualité.
Attributs :	Nom : Nom significatif.
	Type : Type de la valeur avec éventuellement l'intervalle de valeurs accepté (exemple : nombre, pourcentage,...).
	Echelle : Type d'échelle utilisé.
	Unité : Unité à associer à l'indicateur.
Relations :	Interprété par : Un indicateur qualité est interprété par au moins une règle d'interprétation, généralement plusieurs.

Règle d'interprétation		
Définition :	Une règle d'interprétation vise à lier la valeur d'un indicateur à une signification en rapport avec l'objectif qualité.	
Attributs :	Intervalle :	Tranche de valeurs interprétée (exemple : de 0 à 0,05).
	Description :	Description significative de l'intervalle (exemple : maximum un incident par mois).
	Recommandations :	Ensemble des recommandations pour l'intervalle donné (avec causes possibles et actions correspondantes).

5.1.5 Concepts orientés Mesure

Attribut		
Définition :	Un attribut est une propriété mesurable.	
Attributs :	Nom :	Nom reflétant de manière adéquate la caractéristique ciblée (exemple : montant).
	Référence :	Référence (bibliographie, site web,...) dont l'attribut est inspiré.
Relations :	Entrée de :	Un attribut peut être une entrée d'un ou plusieurs modèle(s) d'évaluation.
	Entrée de :	Un attribut peut être une entrée d'une ou plusieurs fonction(s).

Attribut de base (spécialisation de Attribut)		
Définition :	Un attribut de base est un attribut obtenu directement sans aucun traitement.	
Contraintes :	Lors de l'évaluation de l'attribut de base, si celui-ci a plusieurs méthodes qui le définissent, une seule doit être considérée comme active.	

Attribut dérivé (spécialisation de Attribut)		
Définition :	Un attribut dérivé est un attribut obtenu au moyen d'au moins un autre attribut.	
Contraintes :	Lors de l'évaluation de l'attribut dérivé, si celui-ci a plusieurs fonctions qui le définissent, une seule doit être considérée comme active.	

Méthode		
Définition :	Une méthode est une description logique de la technique d'évaluation d'un attribut de base.	
Attributs :	Id :	Identifiant permettant de pointer vers un fichier descriptif de la méthode.
	Nom :	Nom reflétant de manière adéquate la méthode.
	Description :	Description succincte de la méthode.
	Type :	Permet de définir s'il s'agit d'une méthode subjective (Estimation) ou objective (Mesure).
	Echelle :	Type d'échelle utilisé.
	Statut :	Indique le degré de confiance par rapport à cette méthode.
	Unité :	Unité à associer au résultat de la méthode.
	Référence :	Référence (bibliographie, site web,...) dont la méthode est inspirée.
Relations :	Définie pour :	Une méthode est définie pour un attribut de base.

Fonction		
Définition :	Une fonction permet de déterminer un attribut dérivé sur base des attributs en entrée.	
Attributs :	Id :	Identifiant permettant de pointer vers un fichier descriptif de la fonction.
	Nom :	Nom reflétant de manière adéquate la fonction.
	Description :	Description succincte de la fonction.
	Echelle :	Type d'échelle utilisé.
	Unité :	Unité à associer au résultat de la fonction
Relations :	Définie pour :	Une fonction est définie pour un attribut dérivé.

5.1.6 Concepts orientés Produit

Type d'entité mesurable		
Définition :	Un type d'entité mesurable est une propriété directe du produit que l'on peut mesurer.	
Attributs :	Nom :	Nom significatif définissant le type d'entité mesurable (exemples : diagramme de séquence, documentation,...).
	Catégorie :	Catégorie du type d'entité mesurable en fonction de sa spécialisation.
Relations :	A :	Un type d'entité mesurable peut avoir un ou plusieurs attribut(s).

Type d'artéfact (spécialisation de Type d'entité mesurable)		
Définition :	Un type d'artéfact est un élément du produit (exemples : diagramme, code, document,...).	
Attributs :	Langage :	Langage dans lequel le type d'artéfact est exprimé (exemples : UML, C++, langage naturel, langage semi-formel,...).
	Abstraction :	Version relative du type d'artéfact, 0 étant la version actuelle (exemple : -1 est la version précédente).
	Maturité :	Niveau atteint par le type d'artéfact par rapport au cycle de vie (exemples : classe modélisée, classe codée, classe testée,...).
Relations :	Partie de :	Un type d'artéfact peut être une partie d'un autre type d'artéfact (exemple : une méthode est une partie d'une classe).
	Supporte :	Un type d'artéfact peut supporter un ou plusieurs type(s) de comportement.

Type de dérivation (spécialisation de Type d'entité mesurable)		
Définition :	Transformation d'un type d'artéfact en un autre type d'artéfact (exemple : une compilation est une dérivation d'un « code » vers un exécutable).	
Attributs :	Multiplécité :	Nombre entier (ou * pour plusieurs) indiquant le nombre de types d'artéfact issus du type de dérivation.
	Langage :	Langage dans lequel le type de dérivation est exprimé (exemple : XSLT).
	Version :	Version relative au type de dérivation, 0 étant la version actuelle (exemple : -1 est la version précédente).
	Automatisation :	Indique comment le type de dérivation est automatisé.
Relations :	Source :	Un type de dérivation a pour source au moins un type d'artéfact.
	Cible :	Un type de dérivation a pour cible au moins un type d'artéfact.

Type de comportement (spécialisation de Type d'entité mesurable)		
Définition :	Comportement à l'utilisation du produit (exemples : qualité à l'utilisation, calcul d'un taux de change, incident,...).	
Attributs :	Description :	Description du type de comportement.
Relations :	Partie de :	Un type de comportement peut être une partie d'un autre type de comportement (exemple : un incident est une partie d'une collection d'incidents).

5.1.7 Enumérations

Type Méthode	
Estimation :	Méthode subjective.
Mesure :	Méthode objective.

Type Echelle	
Nominale :	Libellés sans ordre (exemple : masculin/féminin).
Ordinale :	Classification ordonnée où les écarts de valeur n'ont pas d'importance (exemple : cotes de restaurant).
Intervalle :	Classification ordonnée sans zéro significatif où les écarts de valeur ont de l'importance (exemple : température).
Ratio :	Classification ordonnée avec zéro significatif où les écarts de valeur ont de l'importance (exemple : hauteur).
Absolue :	Ratio avec transformation (exemple : compteur).

Catégorie Entité mesurable	
<i>Type d'entité mesurable = Type d'artéfact</i>	
Exigence :	Artéfact lié à la récolte des exigences.
Analyse :	Artéfact lié à l'analyse fonctionnelle.
Architecture :	Artéfact lié à l'architecture.
Code :	Artéfact lié à la programmation.
Test :	Artéfact lié à la phase de vérification.
Acceptation :	Artéfact lié à la phase de validation.
Documentation :	Artéfact lié à la documentation.
Production :	Artéfact lié à la phase de production.
<i>Type d'entité mesurable = Type de dérivation</i>	
Horizontale endogène :	Transformation ne changeant pas le niveau de détails et ne changeant pas le méta-modèle utilisé (exemple : refactoring).
Horizontale exogène :	Transformation ne changeant pas le niveau de détails et changeant le méta-modèle utilisé (exemple : méthode Java en méthode C#).
Verticale endogène :	Transformation changeant le niveau de détails et ne changeant pas le méta-modèle utilisé (exemple : méthode non testée en méthode testée).
Verticale exogène :	Transformation changeant le niveau de détails et changeant le méta-modèle utilisé (exemple : compilation).
<i>Type d'entité mesurable = Type de comportement</i>	
Fonctionnel :	Comportement fonctionnel.
Non fonctionnel :	Comportement système.

Type Statut	
Expérimentation :	En phase d'expérimentation.
Validation partielle :	Validé mais utilisation contournée non validée (exemple : détournement du nombre cyclomatique de McCabe).
Validation théorique :	Validé par une démonstration scientifique.
Validation expérimentale :	Validé suite à une expérimentation réelle.
Validation complète :	Validé totalement.

5.1.8 Par où commencer l'instanciation ?

L'expérience a montré que la création d'un modèle de qualité MoCQA peut se faire en quatre étapes : mieux vaut commencer par modéliser la structure des objectifs, continuer avec les entités mesurables suivies des attributs pour revenir à la fin sur les objectifs de qualité pour les compléter. Ces quatre étapes sont décrites dans les sections suivantes.

5.2 Modélisation de la structure des objectifs et des indicateurs

La première étape de la modélisation est de « traduire » la structure des objectifs, déjà identifiée lors de la phase d'acquisition, au format MoCQA. A cette fin, un seul concept est utile : le concept « Objectif qualité » qui peut être subdivisé en plusieurs autres concepts du même type.

Exemple : on l'a vu dans le chapitre précédent, on a structuré l'objectif *Fiabilité* en quatre sous-objectifs : *Survenance des incidents*, *Disponibilité*, *Impact sur les incidents* et *Fréquence de déploiement*. Le résultat de l'instanciation du méta-modèle MoCQA pour représenter cette structure est représenté par la figure ci-dessous.

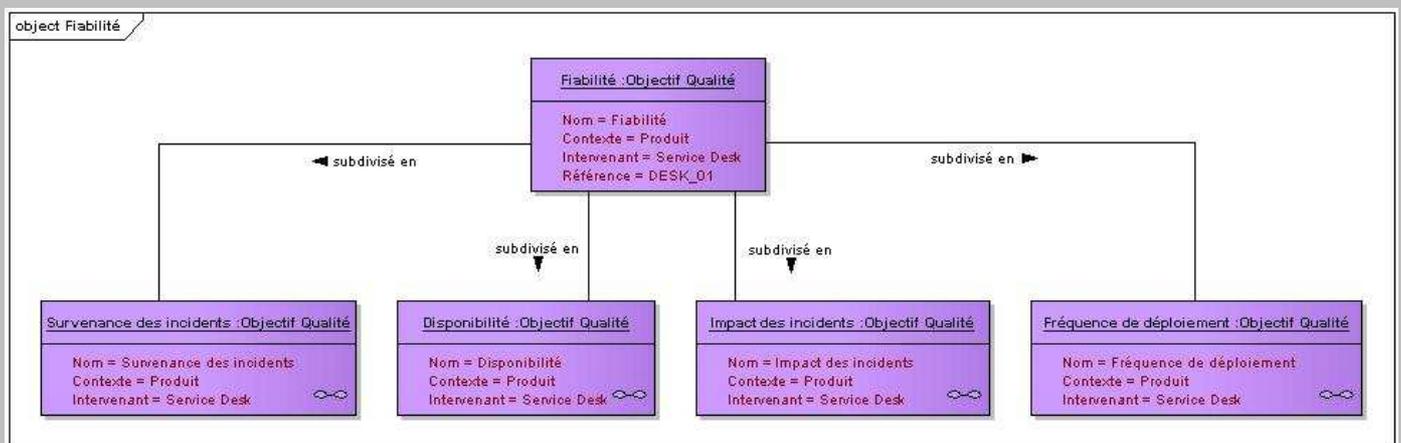


Figure 6 : Modèle "Fiabilité": structure des objectifs de qualité

L'étape suivante vise à compléter le modèle MoCQA (ou à créer des sous-modèles MoCQA) avec chaque sous-objectif.

A partir d'un (sous-)objectif de qualité, on définit son modèle d'évaluation : dans un premier temps, ses attributs peuvent rester vides, l'essentiel étant de modéliser ce qu'il produit, à savoir un indicateur de qualité. C'est ici qu'on va se demander ce qui va représenter concrètement (par une valeur) l'indicateur lié à l'objectif de qualité.

Exemple : on aborde donc le premier sous-objectif, à savoir la *Survenance des incidents*. Que veut-on représenter ici ? Après quelques recherches, on décide qu'un indicateur pertinent serait le nombre moyen d'incidents par mois depuis un an. Contrairement à l'objectif de qualité, l'indicateur est plus concret et on sait ce qu'on va devoir obtenir comme type de valeur.

On représente donc l'indicateur, lié à l'objectif de qualité par un modèle d'évaluation qu'on définira plus tard.

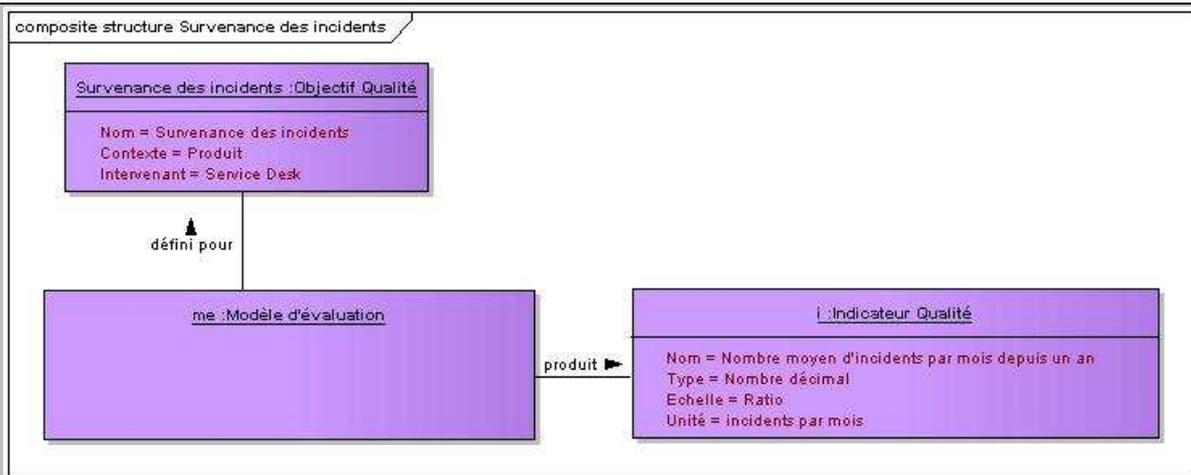


Figure 7 : Modèle "Surveillance des incidents" - Indicateur de qualité

A noter que plusieurs indicateurs peuvent être produits par un modèle d'évaluation : dans ce cas, il est important que les indicateurs produits par un modèle d'évaluation aient la même sémantique.

5.3 Modélisation des entités mesurables

Cette étape consiste à instancier les concepts orientés produit (représentés en jaune dans le méta-modèle). On l'a vu à la section 3.1.3, une entité mesurable peut être spécialisée en trois types différents : un artefact, une dérivation ou un comportement. A noter qu'on n'instanciera pas la classe *Type d'entité mesurable* elle-même mais toujours une de ses trois spécialisations.

L'objectif est d'identifier les éléments qui vont être mesurés et leurs relations entre eux. Bien que ce ne soit pas obligatoire, il est recommandé de toujours représenter l'artefact représentant ce sur quoi porte l'évaluation (le produit). On se limitera aux types d'entité mesurable qui sont pertinents pour le modèle créé, à savoir ceux pour lesquels un attribut est évalué ou qui facilitent la compréhension du modèle.

Exemple : on commence le modèle Surveillance des incidents par représenter le produit évalué. Sur celui-ci surviennent des incidents. Comme on va devoir disposer du nombre d'incidents, c'est sur l'ensemble des incidents (la collection) que va porter l'un de nos attributs : cette collection doit donc être représentée.

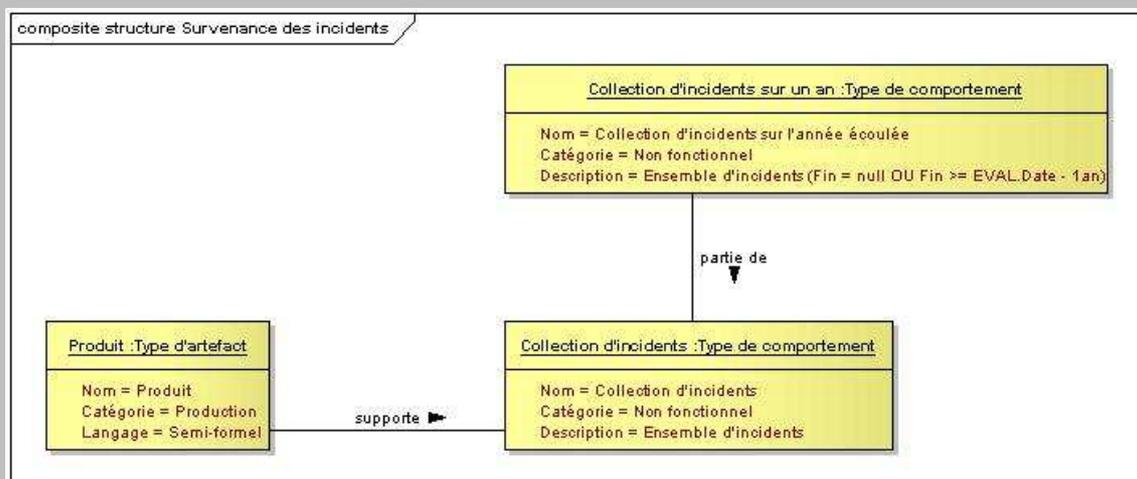


Figure 8 : Modèle "Surveillance des incidents" - Concepts orientés produit

On peut remarquer qu'on a représenté la collection d'incidents ainsi qu'une sous-collection d'incidents reprenant uniquement les incidents depuis un an, ce qui permet de représenter la notion temporelle dans ce cas-ci.

5.4 Modélisation des attributs

A cette étape, on dispose des entités mesurables et de l'indicateur final à évaluer : on doit donc identifier les attributs dont on aura besoin pour évaluer cet indicateur. On va donc modéliser l'ensemble des attributs à mesurer ou calculer sur les entités mesurables.

On distingue ici deux spécialisations d'attributs :

- les attributs de base dont la valeur peut être acquise directement sur l'entité mesurable et
- les attributs dérivés qui sont le résultat d'un traitement sur base d'au moins un autre attribut.

A noter qu'on n'instanciera pas la classe *Attribut* elle-même mais toujours une de ses deux spécialisations.

Chaque attribut doit être accompagné de la manière employée pour l'obtenir : une méthode accompagne un attribut de base et une fonction accompagne un attribut dérivé.

Exemple : pour calculer le nombre d'incidents moyen, on a évidemment besoin du nombre d'incidents : il s'agit d'un attribut de base qu'a la collection d'incidents.

Cela pourrait suffire si l'ensemble des produits évalués existent depuis plus d'un an puisqu'on diviserait le nombre d'incidents par 12 pour avoir son nombre d'incidents moyen par mois. Mais ce ne serait pas équitable pour un produit de moins d'un an : il faut donc connaître l'âge du produit évalué.

Pour ce faire, on acquiert un attribut de base contenant sa date de création, qui est une entrée à une fonction déterminant un attribut dérivé contenant son âge.

La figure suivante montre le résultat obtenu. On peut remarquer que l'âge est calculé en faisant la différence entre la date de création du produit et la date de l'évaluation : cette dernière est plus intéressante à utiliser que la date du jour car la réexécution d'une même évaluation donnerait ainsi des résultats toujours identiques.

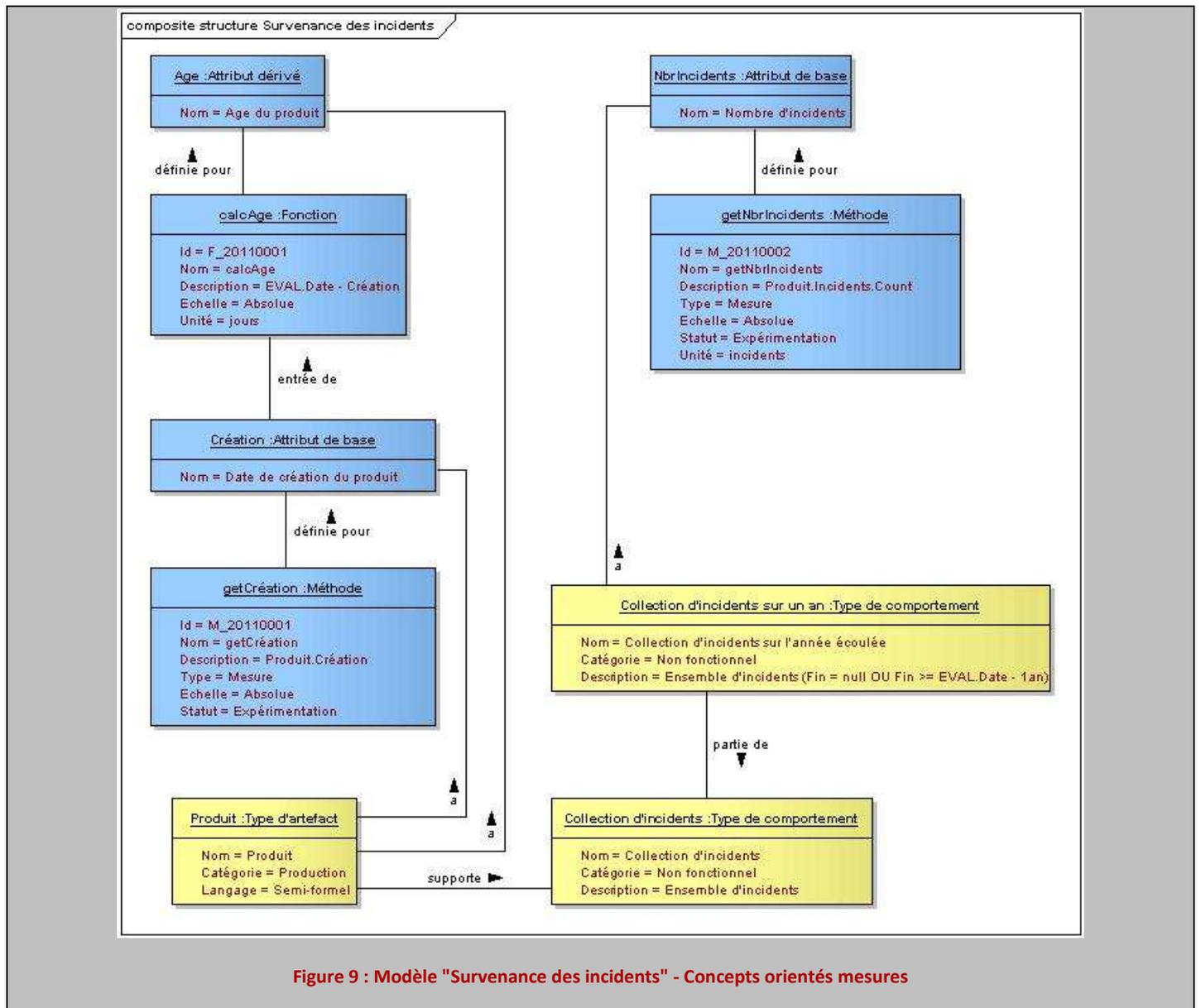


Figure 9 : Modèle "Surveillance des incidents" - Concepts orientés mesures

Dans cette étape de modélisation des attributs, il se peut qu'on identifie certains manques dans ce qui a été réalisé auparavant, typiquement un type d'entité mesurable manquant, etc.

5.5 Finalisation du modèle

On connaît, à ce stade, les attributs à mesurer et l'indicateur à produire : il reste à les lier avec un modèle d'évaluation. Ce modèle vise simplement à dire ce qu'on doit faire des attributs en entrée pour produire l'indicateur défini. Pour des cas simples, il s'agit d'une simple formule ; pour des cas plus complexes, il peut s'agir d'un algorithme ou d'un ensemble de procédures.

Une fois le modèle d'évaluation défini, ce n'est pas encore terminé : il reste la question délicate de l'interprétation de l'indicateur de qualité. On va donc définir des règles d'interprétation de cet indicateur : tout d'abord en créant des tranches de valeurs représentant généralement les niveaux d'atteinte de l'objectif.

Bien que rien ne soit imposé à ce sujet, on recommande de définir un code couleur lié à chaque intervalle de valeurs : on pourrait imaginer « vert » quand c'est bien et « rouge » quand ce ne l'est pas. On peut également affiner ceci en ajoutant « jaune » quand il reste un petit effort à fournir et « orange » quand on est à la limite à ne pas

dépasser. Ces choix sont dépendants de la culture de l'entreprise mais l'utilisation de couleurs pourra être répercutée dans le tableau d'évaluation des indicateurs et permettra une meilleure lisibilité de celui-ci.

Le choix des intervalles n'est pas chose facile : souvent, l'exploitation des premiers résultats amènera une remise en question de ce choix afin que les tranches de valeur soient les plus représentatives possibles du niveau de qualité attendu.

Exemple : on complète le modèle d'évaluation en spécifiant que l'indicateur est calculé en divisant le nombre d'incidents par l'âge du produit (avec limite à un an) puis en ramenant la moyenne sur un mois.

Quatre règles d'interprétation sont définies : on décide qu'une valeur représentant moins d'un incident par trimestre est le niveau attendu de qualité. On décline ensuite les niveaux inférieurs comme indiqué sur la figure suivante.

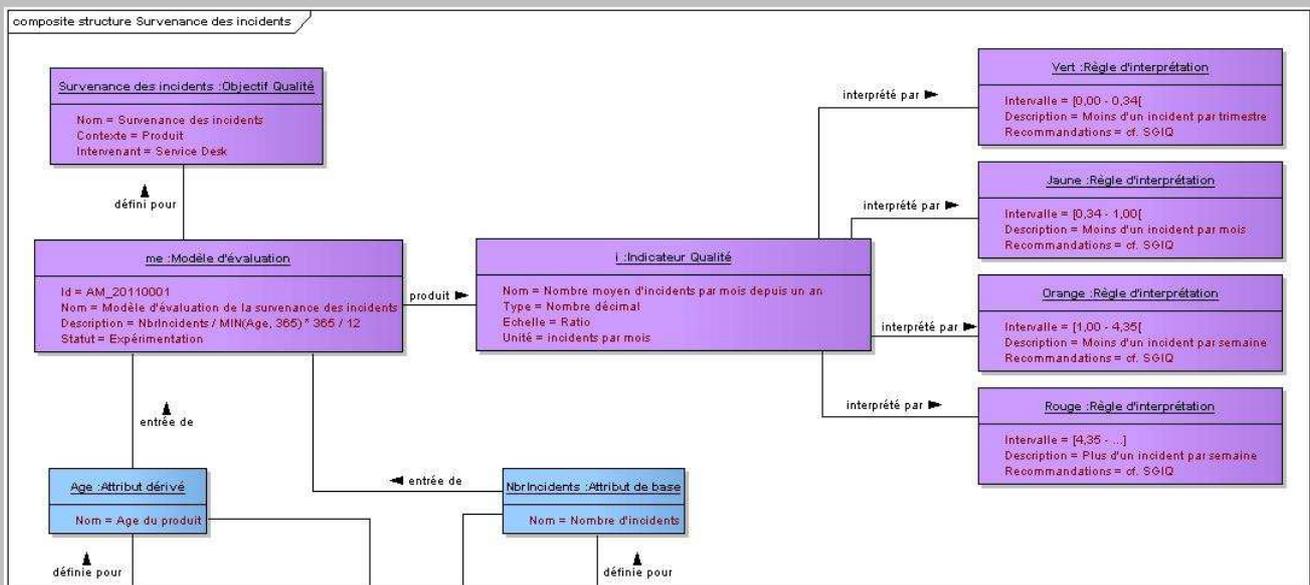


Figure 10 : Modèle "Survenance des incidents" - Concepts orientés qualité

Ensuite, pour chaque règle d'interprétation, on associe un ensemble de recommandations. La première recommandation est généralement la plus simple : c'est l'interprétation la plus intuitive de l'indicateur, à savoir que si on est dans le « vert », on doit continuer dans ce sens et si on est dans le « rouge », on doit améliorer la situation.

Mais ce n'est généralement pas la seule interprétation possible de l'indicateur, comme on l'a vu dans la section 3.1.1. Il faut donc tenter d'identifier un maximum de situations amenant à une telle valeur et, pour chacune, identifier l'action appropriée. L'exemple ci-dessous permettra probablement de mieux illustrer ces concepts.

Enfin, il ne faut pas oublier que l'indicateur peut être erroné ou qu'une interprétation possible n'a pas été identifiée, auquel cas l'action correspondante est de signaler ce défaut au qualicien afin de mener des actions correctives.

Exemple : on peut voir ci-dessous les règles d'interprétation de l'indicateur de *Survenance des incidents*. Pour chaque tranche de valeur est présenté un tableau reprenant chaque interprétation possible identifiée, ainsi que les actions jugées appropriées pour chaque cas.

Règles d'interprétation

Nombre moyen d'incidents par mois depuis un an

Intervalle

Code couleur	Intervalle de valeurs	Description
Vert	0,00 <= I < 0,34	moins d'un incident par trimestre
Jaune	0,34 <= I < 1,00	moins d'un incident par mois
Orange	1,00 <= I < 4,35	moins d'un incident par semaine
Rouge	4,35 <= I	plus d'un incident par semaine

Recommandations

Interprétations possibles d'une valeur verte	Action
Peu ou pas d'incidents concernant ce produit.	Continuer comme ça !
Peu ou pas d'incidents sont remontés au Service Desk.	Sensibiliser les utilisateurs du produit.
Peu ou pas d'incidents sont encodés par le Service Desk.	Demander au Service Desk d'encoder les incidents.
Le produit est obsolète et n'est plus utilisé.	Le faire supprimer de la liste des produits.
Les valeurs mesurées et/ou les calculs d'évaluation sont erronés.	Décrire l'erreur à la cellule AQI.
Autre ?	Contactez la cellule AQI.
Interprétations possibles d'une valeur jaune	Action
Quelques incidents concernant ce produit.	En fonction de la criticité du produit, mener des actions correctives.
Quelques incidents sont remontés au Service Desk mais pas tous.	Sensibiliser les utilisateurs du produit.
Quelques incidents sont encodés par le Service Desk mais pas tous.	Demander au Service Desk d'encoder les incidents.
Les valeurs mesurées et/ou les calculs d'évaluation sont erronés.	Décrire l'erreur à la cellule AQI.
Autre ?	Contactez la cellule AQI.
Interprétations possibles d'une valeur orange	Action
Beaucoup d'incidents concernant ce produit.	Mener des actions correctives.
Beaucoup d'incidents sont remontés par erreur au Service Desk.	Sensibiliser et/ou former les utilisateurs du produit.
Les valeurs mesurées et/ou les calculs d'évaluation sont erronés.	Décrire l'erreur à la cellule AQI.
Autre ?	Contactez la cellule AQI.
Interprétations possibles d'une valeur rouge	Action
Trop d'incidents concernant ce produit.	Mener des actions correctives prioritaires.
Trop d'incidents sont remontés par erreur au Service Desk.	Sensibiliser et/ou former les utilisateurs du produit.
Les valeurs mesurées et/ou les calculs d'évaluation sont erronés.	Décrire l'erreur à la cellule AQI.
Autre ?	Contactez la cellule AQI.

Figure 11 : Modèle "Surveillance des incidents" - Règles d'interprétation

Si l'indicateur est dans le « vert », il est probable que c'est parce qu'il n'y a pas ou peu d'incidents : l'action appropriée dans ce cas est tout simplement un mot d'encouragement. Mais il se peut que des incidents existent mais qu'ils ne soient tout simplement pas signalés par les utilisateurs du produit, auquel cas il faut les sensibiliser à l'importance de remonter les incidents. C'est ainsi que plusieurs interprétations sont identifiées pour chaque situation.

A noter les deux dernières interprétations possibles pour chaque tranche de valeurs qui identifient la possibilité que les valeurs soient erronées ou qu'on a oublié une interprétation possible : dans ce cas, la cellule d'Assurance Qualité Informatique en charge de la gestion des indicateurs de qualité est contactée.

5.6 Réflexion au sujet des entités mesurables

Afin de conserver une cohérence des types d'entité mesurable au travers des différents modèles MoCQA réalisés, le cycle de vie des produits peut être modélisé au format MoCQA, à savoir uniquement des concepts orientés produit dans un même schéma. Cela a l'avantage de toujours utiliser les mêmes termes pour un même concept (exemple : un type d'entité mesurable « analyse fonctionnelle » portera toujours ce nom alors que, en l'absence d'une formalisation claire, on pourra une fois modéliser « analyse fonctionnelle », une autre fois « analyse », ou encore « analyse des exigences », etc.).

D'un côté pratique lors de la modélisation MoCQA, cette schématisation permet également de réutiliser rapidement des types d'entité mesurable, le qualicien ayant à sa disposition une « bibliothèque » des types d'entité mesurable existants. En fonction de l'outil utilisé, cela peut constituer un gros gain de temps si l'outil permet de faire du « cliquer-déposer » d'un modèle à un autre.

Ce choix est dépendant de l'environnement où on déploie l'approche MoCQA et de la portée des indicateurs : si l'entreprise est dans une logique ALM, il est probable que ce soit profitable à moyen terme. Si on décide de modéliser le cycle de vie, on peut le faire de deux façons :

- soit complètement, en une seule fois, à condition que le cycle de vie soit clairement défini au sein de l'entreprise,
- soit progressivement, au fur et à mesure de l'utilisation de nouveaux types d'entité mesurable, ce qui permet une approche plus itérative et généralement plus productive.

5.7 Le mot de la fin

Les différentes étapes de la phase de modélisation proposées ici ne sont qu'une suggestion principalement pour les débutants : il se peut qu'avec l'expérience, le qualicien réalise le modèle dans un ordre différent, l'essentiel étant que tout s'y retrouve au final.

La phase de modélisation constitue probablement la phase la plus complexe, les phases suivantes étant plus intuitives pour les professionnels de l'informatique. A la fin de cette phase de modélisation, on dispose d'un modèle de qualité formalisant ce qu'on doit mesurer (le « quoi »), comme l'indique la figure ci-dessous. On peut ainsi passer à l'étape d'opérationnalisation, visant à décrire comment on va effectuer les mesures, décrite dans le chapitre suivant.

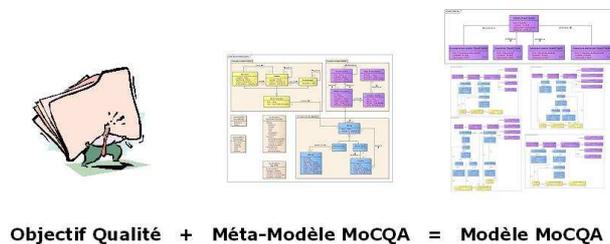


Figure 12 : Représentation de la phase de modélisation

6 Opérationnalisation

Après avoir déterminé ce qu'on allait mesurer lors de la phase précédente, la troisième phase vise à déterminer comment le mesurer. Elle est assez simple sur le principe puisqu'elle se base sur la connaissance de la structure des informations au sein de l'entreprise, mais peut avoir un certain coût en fonction de la complexité des modèles MoCQA à opérationnaliser.

Il s'agit donc ici de déterminer, pour chaque attribut, la façon d'en faire l'acquisition (interrogation d'une ressource humaine ou matérielle, souvent une base de données) et/ou de le calculer (manuellement ou avec automatisation) au sein de l'environnement où se déroule l'opérationnalisation. On y affine également les modèles d'évaluation si besoin.

Le plan de mesure se présentera sous la forme jugée la plus appropriée en fonction des pratiques internes de l'entreprise (document, espace documentaire,...).

6.1 Méthodes et fonctions

Chaque attribut a une méthode ou une fonction liée, selon qu'il est un attribut de base ou un attribut dérivé. Chaque méthode et fonction doit être décrite au sein d'un plan de mesure.

Si le cycle de vie des entités mesurables a été défini au format MoCQA (voir section 5.6), il serait intéressant de définir les méthodes et fonctions dans un espace transversal, ce qui permettra leur réutilisation.

Pour chaque méthode, on signalera la façon de mesurer l'attribut : il s'agira tantôt d'une requête sur une base de données, tantôt d'aller interroger une personne spécifique, tantôt de consulter un outil particulier,...

Exemple : pour l'attribut *NbrIncidents* du modèle *Survenance des incidents*, on sait qu'on doit interroger la base de données de l'outil *TrackIT!* où les incidents sont enregistrés :

```
SELECT COUNT(*)
  FROM tasks
 WHERE lookup1 = CodeProduit
        AND (clsddate IS NULL OR clsddate BETWEEN DATEADD(YEAR, -1, DateEvaluation) AND DateEvaluation)
```

CodeProduit étant le code permettant d'identifier le produit évalué dans l'outil et *DateEvaluation* étant la date à laquelle l'évaluation a lieu.

Pour la consultation d'une base de données, il n'est pas obligatoire de représenter chaque requête : il s'agit d'un choix de l'entreprise. En effet, on pourrait se contenter d'une description succincte des champs importants, la personne effectuant l'évaluation étant supposée alors connaître la syntaxe permettant d'obtenir les informations demandées.

Pour chaque fonction, on signalera l'algorithme de calcul, à nouveau de façon détaillée ou en langage naturel.

Exemple : pour calculer l'attribut *Age*, on doit faire la différence entre la date de l'évaluation et la date de création du produit :

```
calcAge(e:Evaluation, p:Produit) : integer
{
    date d = getCréation(p)
    integer r = e.date - d
    return r
}
```

6.2 Modèles d'évaluation

Sauf si la description qui en est faite dans le modèle MoCQA est jugée suffisante, il est intéressant de décrire les modèles d'évaluation dans le plan de mesure.

A nouveau, le degré de détails est fonction d'un choix à faire au sein de l'entreprise, l'essentiel étant que la personne effectuant l'évaluation sache comment, à partir des attributs en entrée, déterminer l'indicateur.

Exemple : l'indicateur de *Survenance des incidents* est ici très simple :

```
evalSurvenanceDesIncidents(e:Evaluation, p:Produit) : decimal
{
    integer n = getNbrIncidents(e, p)
    integer a = calcAge(e, p)
    decimal i = n / MIN(a, 365) * 365 / 12
    return i
}
```

On rappelle qu'un modèle d'évaluation pourra être plus complexe que l'exemple proposé ci-dessus en fonction :

- du nombre d'attributs en entrée,
- du nombre d'objectifs de qualité en entrée et
- du nombre d'indicateurs de qualité en sortie.

6.3 Le mot de la fin

A la fin de cette phase d'opérationnalisation, on dispose d'un plan de mesure indiquant comment mesurer (le « comment »), comme l'indique la figure ci-dessous. On peut ainsi passer à l'étape d'évaluation, qui va permettre d'enfin avoir des résultats à exploiter.

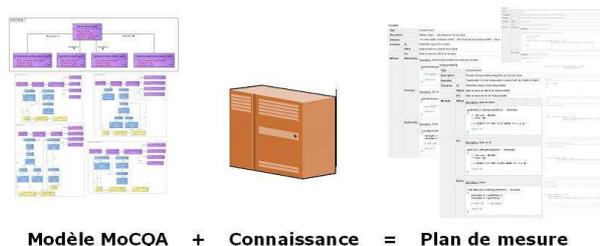


Figure 13 : Représentation de la phase d'opérationnalisation

7 Evaluation

Une fois que l'on a déterminé ce qu'on allait mesurer (phase de modélisation) et comment le mesurer (phase d'opérationnalisation), on peut réaliser l'évaluation elle-même.

Il s'agit ici d'appliquer le plan de mesure sur base du modèle de qualité afin d'acquérir et de calculer les métriques adéquates, ainsi que d'évaluer les indicateurs de qualité.

7.1 Application du plan de mesure

On applique le plan de mesure sur base du modèle de qualité afin d'obtenir les résultats : c'est ici qu'on exécute les requêtes en base de données, qu'on applique les procédures, qu'on fait tourner les algorithmes, etc. identifiés lors de la phase d'opérationnalisation.

Les résultats doivent être stockés dans une base de données (au sens large du terme) : il peut s'agir d'un système de gestion de base de données, d'un simple fichier Excel ou XML, d'un environnement de type *Business Intelligence*, etc. Ce choix dépend de la culture de l'entreprise, de la portée de l'évaluation et aussi des outils disponibles au sein de l'entreprise.

Concernant la nature des résultats à stocker, à nouveau, un choix doit s'opérer : soit on sauvegarde l'ensemble des valeurs d'attributs afin de pouvoir retrouver l'historique d'une valeur facilement, soit on ne sauvegarde que les valeurs d'indicateurs.

7.2 Opportunité d'automatisation

Bien que ce ne soit pas une obligation, il est intéressant d'étudier l'opportunité d'automatiser certaines méthodes, fonctions et/ou modèles d'évaluation. Cette automatisation permettrait évidemment un gain de temps au niveau de l'évaluation et aussi de mettre en place des scripts automatiques effectuant les évaluations à intervalles de temps réguliers.

7.3 Présentation des résultats

La présentation des résultats est de grande importance : il est plus qu'utile d'appliquer les codes couleur définis avec les règles d'interprétation aux résultats afin d'identifier les indicateurs plus sensibles.

Définir des statistiques (minimum, maximum, moyenne,...) sur les indicateurs peut s'avérer intéressant pour la hiérarchie si on évalue plusieurs produits en même temps.

Lorsqu'on communique les résultats, ils doivent être accompagnés de la méthode utilisée pour les obtenir (modèles de qualité et plans de mesure), ainsi que des règles d'interprétations identifiées. Les valeurs des attributs peuvent également être données, afin d'aider à la compréhension de certaines valeurs d'indicateurs.

7.4 Accès aux résultats

On l'a vu précédemment, le risque de mauvaise interprétation des indicateurs est présent et nécessite une bonne communication sur l'utilisation de ceux-ci.

Il semble donc important de limiter l'accès aux résultats d'évaluation aux personnes concernées par ceux-ci.

En effet, une crainte constatée auprès des ressources impliquées dans un produit qui a été évalué est la comparaison avec leurs collègues. Cela dépend de la politique de l'entreprise où est effectué le déploiement mais on peut décider que seules certaines ressources ont accès à l'entièreté des indicateurs (typiquement : la hiérarchie) et qu'un responsable de produit a accès uniquement aux indicateurs concernant son produit, ce qui évitera d'éveiller la curiosité et de comparer ces résultats à ceux des autres équipes.

7.5 Le mot de la fin

A la fin de cette phase, on dispose alors d'un tableau reprenant, pour chaque produit logiciel évalué, les valeurs liées à chaque indicateur de qualité, comme l'indique la figure ci-dessous. Si on s'arrêtait ici, on disposerait tout simplement de chiffres, éventuellement de couleurs différentes : tout l'intérêt réside dans la dernière phase, à savoir la phase d'exploitation.

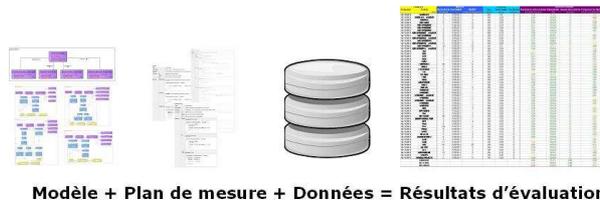


Figure 14 : Représentation de la phase d'évaluation

8 Exploitation

La dernière phase du processus de gestion des indicateurs de qualité vise à exploiter les résultats issus de la phase d'évaluation.

C'est ici qu'on étudiera les résultats de l'évaluation, ce qui permettra de faire des constatations :

- sur le modèle de qualité issu de la phase de modélisation et éventuellement identifier des changements à y apporter,
- sur le plan de mesure issu de la phase d'opérationnalisation et éventuellement identifier des changements à y apporter,
- sur les objectifs de qualité à atteindre, ce qui peut donner lieu à un déclassement de certains indicateurs jugés inutiles et/ou à l'identification de nouveaux objectifs à modéliser,
- sur la qualité des produits logiciels évalués et éventuellement identifier des actions correctives à apporter sur les produits.

8.1 Qui exploite les résultats d'évaluation ?

A nouveau, cela dépend de la culture et de la taille de l'entreprise. On recommande de faire exploiter les résultats par les ressources humaines ayant les rôles suivants :

- la ou les personnes ayant réalisé l'évaluation,
- la ou les personnes responsables du produit évalué ou responsables de la liste des produits si plusieurs et
- la ou les personnes ayant le pouvoir de décision (typiquement, la hiérarchie).

On ajoutera à ce noyau commun toute personne jugée utile à l'exploitation des résultats ciblés.

Exemple : l'environnement où est déployé l'exemple qui sert de fil rouge au présent guide étant un organisme de grande taille, un comité de gestion des indicateurs a été mis en place. Ce comité est composé du chef de service, du responsable de la qualité, du responsable des applications métiers et du responsable des outils transversaux. En fonction de la nature des résultats à évaluer, d'autres ressources sont invitées à se joindre au comité lors des phases d'exploitation.

8.2 Procédure d'exploitation

Afin d'exploiter les résultats d'une évaluation, une réunion doit être planifiée. On invite à cette réunion les ressources humaines identifiées dans la section précédente.

La question se pose alors de la pertinence d'envoyer les résultats avant la réunion ou pas. Les deux options ont toutes deux des avantages et inconvénients :

- si on envoie les résultats avant la réunion, cela permet aux ressources de les étudier et de préparer la réunion. D'un autre côté, si ces mêmes ressources ne sont pas bien conscientes des interprétations possibles des indicateurs, il se peut qu'elles arrivent à la réunion avec des aprioris négatifs ;
- si on n'envoie pas les résultats, on évite ce risque mais on doit alors gérer une réunion qui n'a pas été préparée par les intervenants.

Il est recommandé de choisir la seconde option lors des premières itérations d'exploitation des indicateurs de qualité au sein de l'entreprise, et d'éventuellement se tourner vers la première option lorsque l'on est confiant dans le fait que les ressources ont compris le concept d'interprétation des indicateurs.

En guise de préparation à la réunion, il est utile que l'organisateur (généralement le qualicien ayant effectué l'évaluation) établisse déjà une liste de constatations sur les indicateurs. On commencera la réunion par la présentation des résultats, suivie d'un tour de table permettant à chacun d'exprimer ses constatations, pour compléter éventuellement avec la liste de constatations préparée.

Pour chaque constatation, une action (ou un ensemble d'actions) sera définie : chaque action sera assignée à une ressource particulière et sera ajoutée dans un plan d'actions. Ce plan d'actions devrait être idéalement parcouru au début de la réunion d'exploitation suivante pour faire l'état des actions effectuées ou posant problème.

Les constatations peuvent être de natures différentes :

- soit elles portent sur la démarche d'évaluation elle-même, auquel cas on identifiera des actions liées à la modélisation, l'opérationnalisation ou le domaine organisationnel ;
- soit elles portent sur un produit particulier pour lequel on identifie des actions, généralement correctrices, à accomplir.

8.3 Importance du plan d'actions

Le plan d'actions est très important pour exploiter toute la force des indicateurs de qualité. En effet, il se peut que les premières itérations identifient des problèmes déjà connus au sein de l'entreprise, la résolution de ces problèmes étant régulièrement repoussée à une date ultérieure, sans assignation à une personne particulière.

Si, lors de l'exploitation des indicateurs, on s'oblige à remplir un plan d'actions, on s'oblige en même temps à résoudre les problèmes identifiés en y assignant une ressource, en lui fixant des échéances et en en assurant le suivi.

Exemple : lors de la première exploitation au sein de l'organisme servant d'exemple, le premier constat a été qu'on ne disposait pas d'une liste de produits uniformisée : chaque système (ou outil) utilisait ses propres codes pour désigner les produits, ce qui nuisait à l'agrégation des résultats avec les indicateurs. On se retrouvait parfois avec des produits en double, des produits absents ou carrément des produits obsolètes.

Ce problème avait déjà été soulevé mais régulièrement reporté : la première exploitation a permis d'assigner une personne à une action « *uniformiser la liste des produits* », qui a mené à une solution au bout de quelques temps.

8.4 Le mot de la fin

L'ensemble de ces constatations se transforme ainsi en un plan d'actions à réaliser, comme le montre la figure ci-dessous, ceci dans un souci d'amélioration continue.

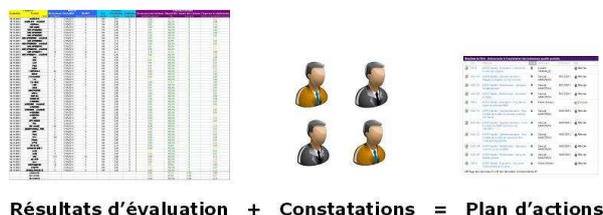


Figure 15 : Représentation de la phase d'exploitation

9 Communication

Bien qu'il ne s'agisse pas d'une phase de l'approche MoCQA, l'expérience a montré que l'aspect communication est primordial à prendre en compte lors d'un déploiement de ce type.

Beaucoup de points importants ont déjà été traités dans les chapitres précédents. On ne les répétera pas mais ils sont résumés ici :

- « Un indicateur peut être interprété de multiples façons et ne se suffit pas à lui-même pour donner un aperçu correct de la qualité » (voir section 3.1.1) ;
- Importance de comprendre la finalité des indicateurs (voir section 3.1.4) ;
- Besoin d'un support hiérarchique (voir section 3.3) ;
- Sensibilisation des intervenants à la démarche (voir section 4.1) ;
- La manière d'obtenir les indicateurs doit être expliquée (voir section 7.3) ;
- L'accès aux résultats d'évaluation doit être restreint (voir section 7.4) ;
- Les ressources humaines exploitant les résultats d'évaluation doivent être soigneusement choisies (voir section 8.1).

Un risque existe lorsqu'on démarre un déploiement de manière progressive, en ne tenant au courant que quelques intervenants de la nature de la démarche : au bout d'un certain temps, et bien que cela soit à nuancer en fonction de la culture de l'entreprise, des bruits de couloir peuvent faire leur apparition et déformer l'objectif du déploiement. Il est essentiel de rester attentif à ce type de risque et de soigner la communication, même informelle, concernant le domaine.

A l'inverse, on recommande de ne pas communiquer « en masse » tant qu'on n'a pas réalisé quelques itérations car cela pourrait nuire à l'objectif du déploiement : en effet, le fait de pouvoir présenter des indicateurs existants, avec les interprétations possibles liées à ceux-ci, sera probablement mieux perçu qu'une présentation théorique amenant son lot de doutes et d'incertitudes au sein des ressources.

10 Conclusion

L'objectif de ce guide est d'expliquer aux professionnels de l'informatique, désireux de mettre en place des indicateurs de qualité sur leurs produits logiciels, ce qu'est et ce que propose l'approche MoCQA en y joignant des exemples concrets d'utilisation et des bonnes pratiques de déploiement.

Essentiellement issues de l'expérience de l'auteur, les bonnes pratiques reprises dans ce document ne sont certainement pas exhaustives : espérons que le guide évolue dans le temps au fur et à mesure des expérimentations réalisées dans différents environnements professionnels.

De son côté, Samuel Hanoteau continue sa collaboration avec les FUNDP, ce qui amènera sans doute à une évolution régulière du présent guide de déploiement. Toutes remarques, suggestions ou retours d'expérience peuvent d'ailleurs être soumises, dans un souci d'amélioration continue.

A cette fin, les coordonnées des personnes de contact sont reprises ci-dessous. Si le présent guide a généré une envie de déploiement de l'approche MoCQA au sein d'une entreprise, nul doute que les FUNDP apporteront le support nécessaire à cet effet.

Naji Habra
(FUNDP)

naji.habra@fundp.ac.be

Benoît Vanderose
(FUNDP)

benoit.vanderose@fundp.ac.be

Samuel Hanoteau
(particulier)

s.hanoteau@vonotar.com

Annexe 2 : Livrables du déploiement

La présente annexe reprend une partie des livrables réalisés lors du déploiement. Les livrables non présentés le sont pour cause de confidentialité.

A2.1 Objectifs de qualité

Ci-dessous un extrait des objectifs de qualité du Service Desk identifiés lors de la première itération.

Informations transversales

Identifiant	DESK
Date de l'interview	19/08/2011
Intervenant	Pierre Girboux
Fonction	Responsable du Service Desk
Type d'intervenant	Générique
Validation	Validé (22/08/2011)
Remarques additionnelles	Non

Objectifs qualité

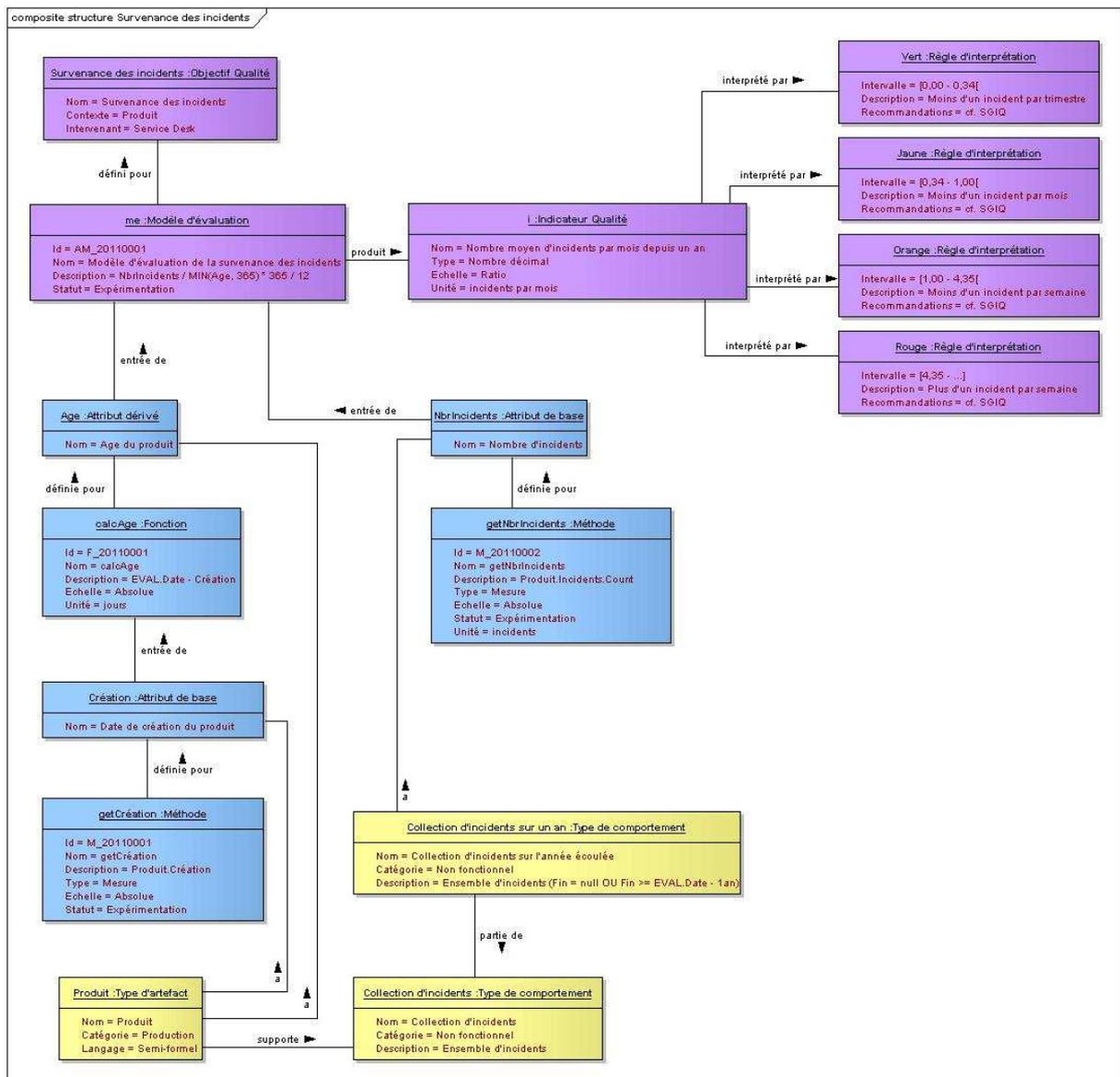
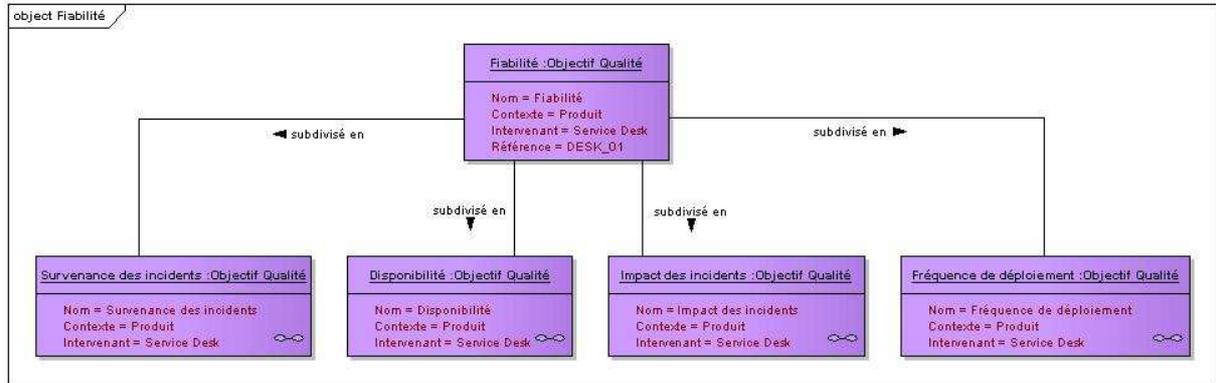
Fiabilité

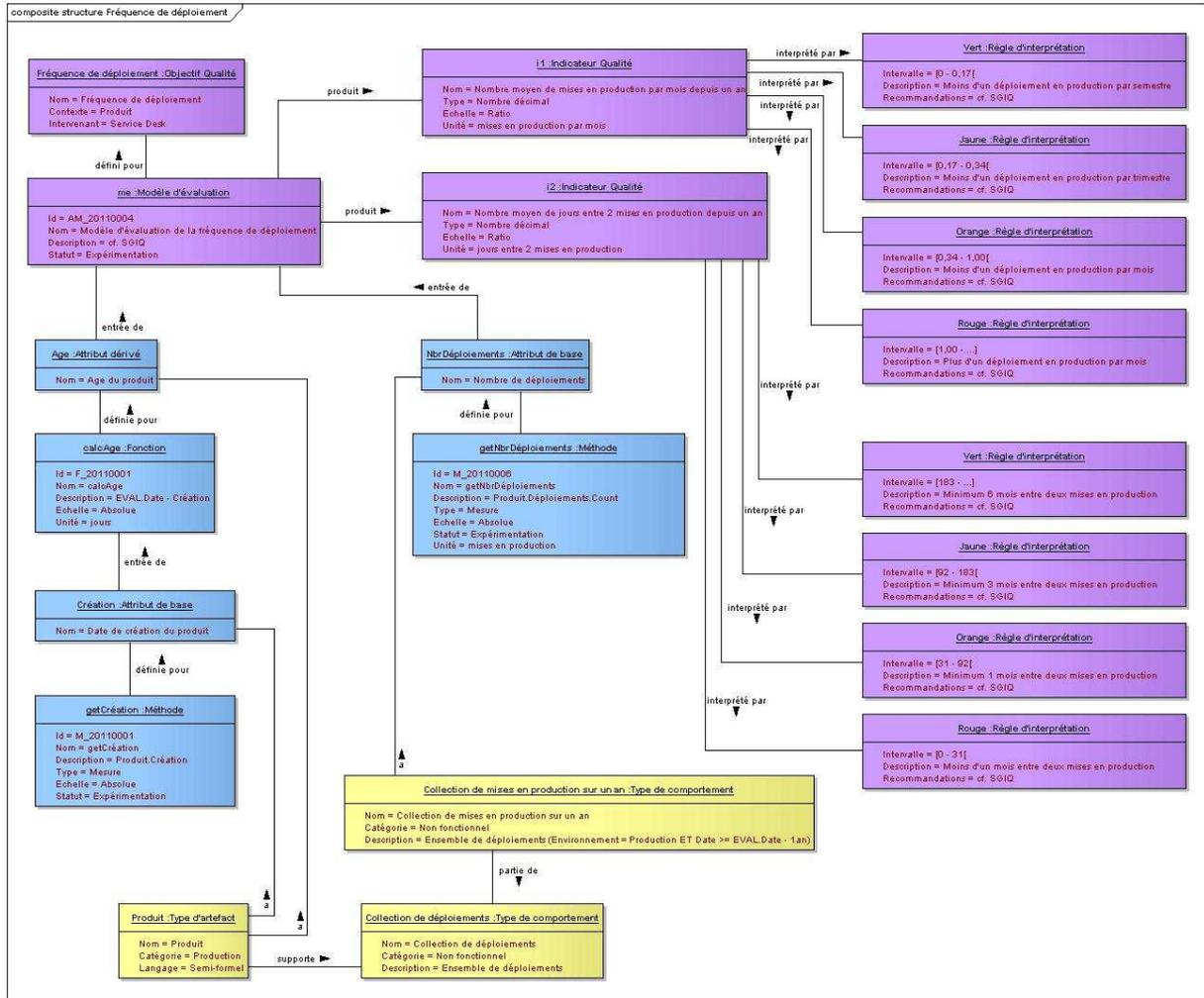
Identifiant	DESK_01
Priorité	1
Description	<p>Les solutions logicielles livrées doivent être fiables et stables. Voici les sous-objectifs identifiés pour cet objectif:</p> <ul style="list-style-type: none"> • Survenance des incidents (un nombre élevé d'incidents peut indiquer un manque de fiabilité) • Disponibilité (une application doit être disponible pour ses utilisateurs au maximum de temps) • Impact des incidents (un incident doit au minimum impacter la production) • Fréquence de déploiement (un produit déployé fréquemment en production n'est pas rassurant)
Standards adoptés	Non
Parties concernées	<ul style="list-style-type: none"> • Livrables
Mesures existantes	<ul style="list-style-type: none"> • Nombre d'incidents liés à une application. <ul style="list-style-type: none"> • Qui: via l'outil Track-It • Niveau de confiance: élevé • Evaluation: à la demande
Entités mesurables	<ul style="list-style-type: none"> • Taux de disponibilité moyen de l'application (pourcentage) • Temps d'indisponibilité moyen après incident (nombre entier, unité: minute) • Nombre de déploiements moyen (nombre entier, unité: déploiement/mois)

A2.2 Modèles MoCQA

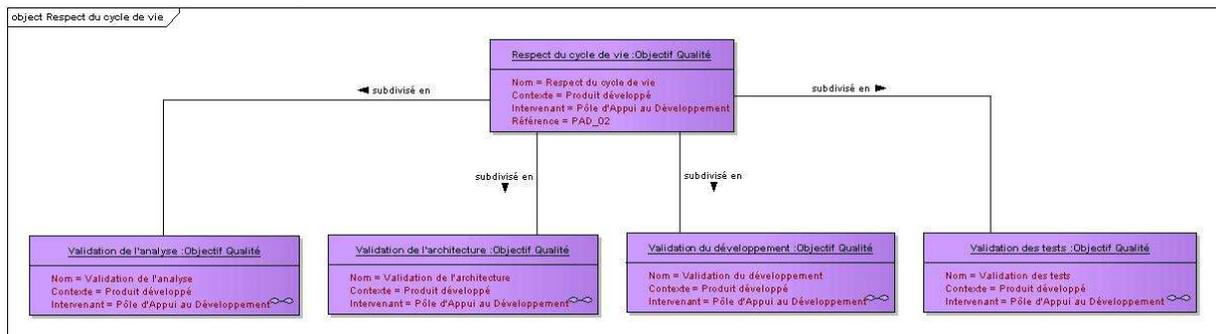
Ci-dessous l'ensemble des modèles MoCQA réalisés pour les objectifs « Fiabilité » et « Respect du cycle de vie ».

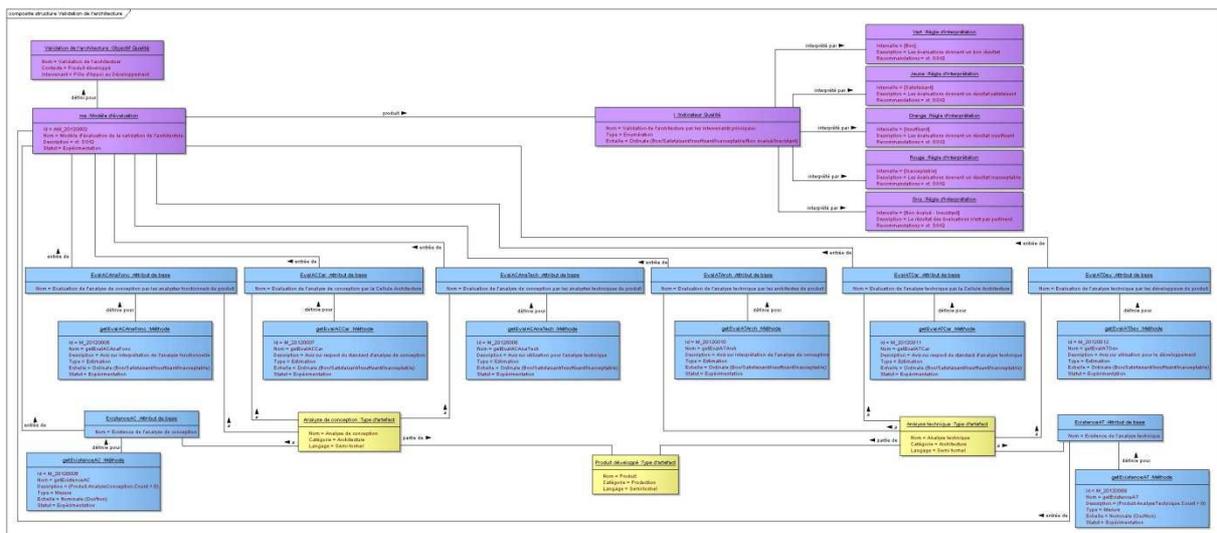
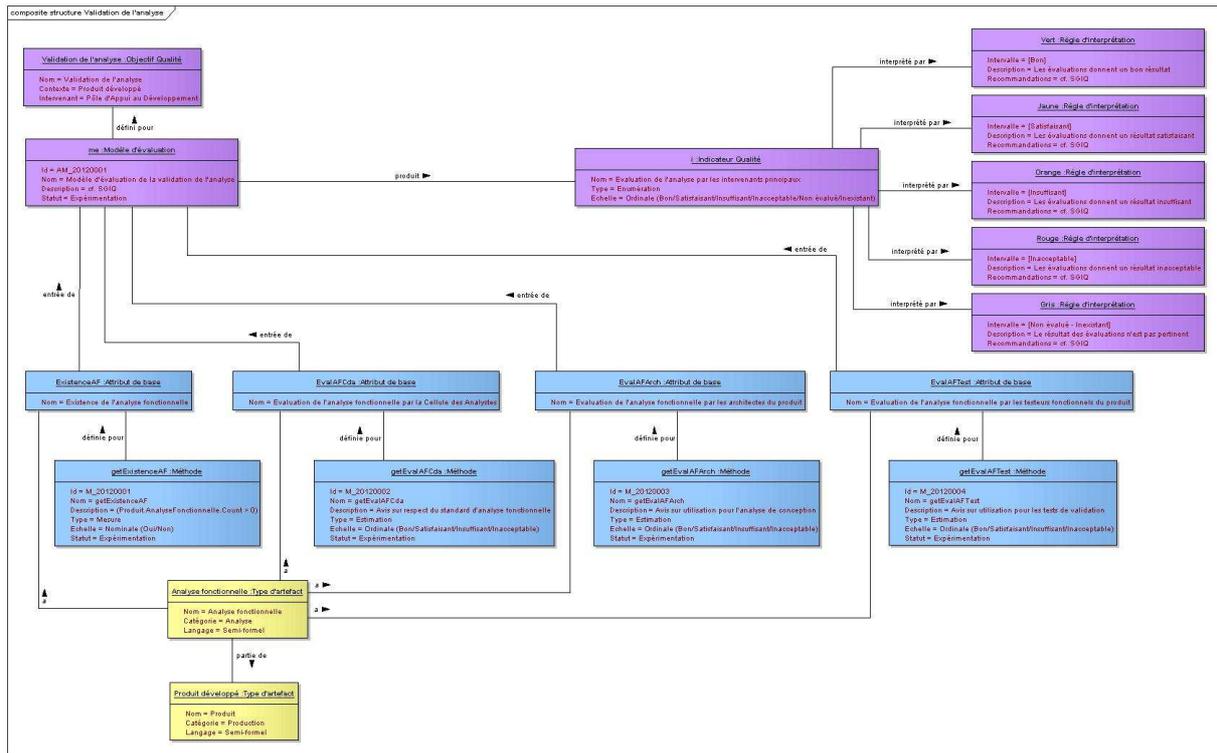
Modèle « Fiabilité »

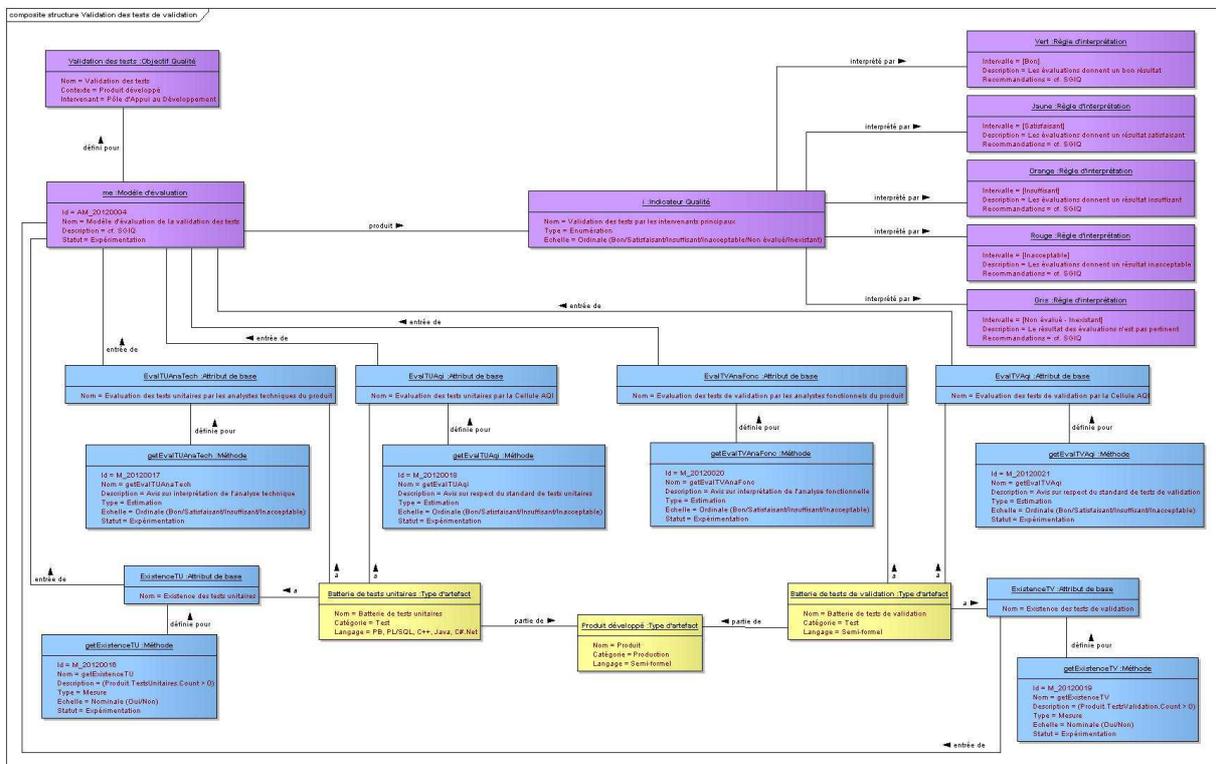
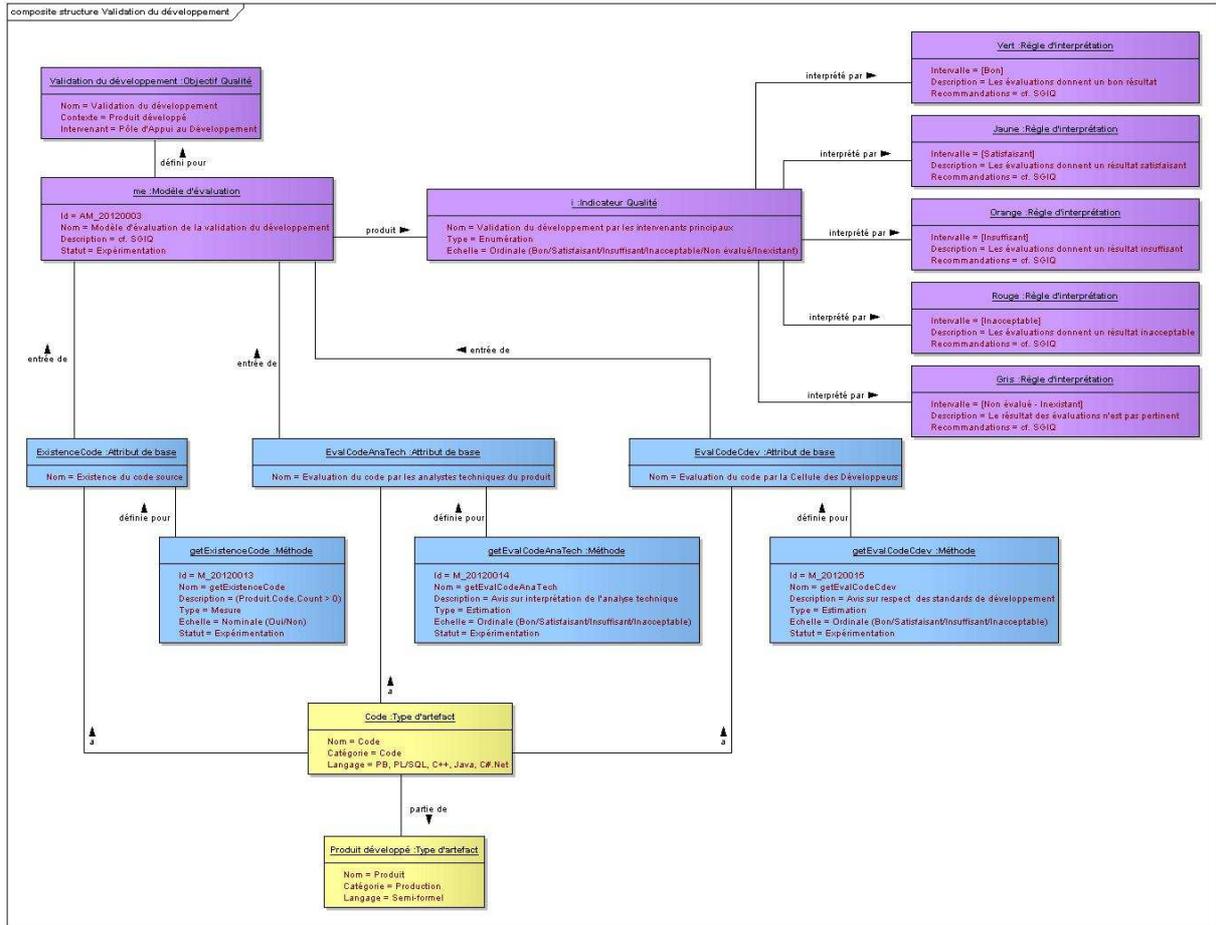




Modèle « Respect du cycle de vie »







A2.3 Plans de mesure et règles d'interprétation

Ci-dessous l'ensemble des plans de mesure, accompagnés des règles d'interprétation, réalisés pour les modèles « Fiabilité » et « Respect du cycle de vie » ainsi que le plan de mesures transversal des différents attributs liés aux entités mesurables.

Plan de mesure « Fiabilité »

Modèle d'évaluation

Informations

Id	AM_20110001
Nom	Survenance des Incidents
Statut	Experimentation

Détails d'évaluation

```
evalSurvenanceDesIncidents(e:évaluation, p:produit) : (1:decimal)
{
  Integer n = getNbrIncidents(e, p)
  Integer a = CalculAge(e, p)

  // l = Nombre moyen d'Incidents par mois depuis un an
  l = n / MIN(a, 365) * 365 / 12
}
```

Règles d'interprétation

Nombre moyen d'incidents par mois depuis un an

Intervalle

Code couleur	Intervalle de valeurs	Description
Vert	0,00 <= l < 0,34	moins d'un incident par trimestre
Jaune	0,34 <= l < 1,00	moins d'un incident par mois
Orange	1,00 <= l < 4,35	moins d'un incident par semaine
Rouge	4,35 <= l	plus d'un incident par semaine

Recommandations

Interprétations possibles d'une valeur verte	Action
Peu ou pas d'incidents concernant ce produit.	Continuer comme ça !
Peu ou pas d'incidents sont remontés au Service Desk.	Sensibiliser les utilisateurs du produit.
Peu ou pas d'incidents sont encodés par le Service Desk.	Demander au Service Desk d'encoder les incidents.
Le produit est obsolète et n'est plus utilisé.	Le faire supprimer de la liste des produits.
Les valeurs mesurées et/ou les calculs d'évaluation sont erronés.	Décrire l'erreur à la cellule AQL.
Autre ?	Contactez la cellule AQL.
Interprétations possibles d'une valeur jaune	Action
Quelques incidents concernant ce produit.	En fonction de la criticité du produit, mener des actions correctives.
Quelques incidents sont remontés au Service Desk mais pas tous.	Sensibiliser les utilisateurs du produit.
Quelques incidents sont encodés par le Service Desk mais pas tous.	Demander au Service Desk d'encoder les incidents.
Les valeurs mesurées et/ou les calculs d'évaluation sont erronés.	Décrire l'erreur à la cellule AQL.
Autre ?	Contactez la cellule AQL.
Interprétations possibles d'une valeur orange	Action
Beaucoup d'incidents concernant ce produit.	Mener des actions correctives.
Beaucoup d'incidents sont remontés par erreur au Service Desk.	Sensibiliser et/ou former les utilisateurs du produit.
Les valeurs mesurées et/ou les calculs d'évaluation sont erronés.	Décrire l'erreur à la cellule AQL.
Autre ?	Contactez la cellule AQL.
Interprétations possibles d'une valeur rouge	Action
Trop d'incidents concernant ce produit.	Mener des actions correctives prioritaires.
Trop d'incidents sont remontés par erreur au Service Desk.	Sensibiliser et/ou former les utilisateurs du produit.
Les valeurs mesurées et/ou les calculs d'évaluation sont erronés.	Décrire l'erreur à la cellule AQL.
Autre ?	Contactez la cellule AQL.

Modèle d'évaluation

Informations

Id	AM_20110002
Nom	Disponibilité
Statut	Expérimentation

Détails d'évaluation

```

evalDisponibilité(w:évaluation, p:produit) : (i:decimal)
{
  decimal d = CalculDuréeIndisp(w, p)
  Integer a = CalculAge(w, p)
  // i = Taux de disponibilité depuis un an
  i = d / MSN(a, 305)
}

```

Règles d'interprétation

Taux de disponibilité depuis un an

Intervalle

Code couleur	Intervalle de valeurs	Description
Vert	100% >= i > 99,8%	Maximum 1 heure d'indisponibilité par mois
Jaune	99,8% >= i > 98,9%	Maximum 8 heures d'indisponibilité par mois
Orange	98,9% >= i > 96,7%	Maximum 24 heures d'indisponibilité par mois
Rouge	96,7% >= i	Plus de 24 heures d'indisponibilité par mois

Recommandations

Interprétations possibles d'une valeur verte	Action
Le produit comporte peu ou pas d'anomalies.	Continuer comme ça !
Le produit est très tolérant aux pannes.	Continuer comme ça !
Peu ou pas d'indisponibilités sont encodées par le Service Desk.	Demander au Service Desk d'encoder les indisponibilités.
Le produit est obsolète et n'est plus utilisé.	Le faire supprimer de la liste des produits.
Les valeurs mesurées et/ou les calculs d'évaluation sont erronés.	Décrire l'erreur à la cellule AQI.
Autre ?	Contactez la cellule AQI.
Interprétations possibles d'une valeur jaune	Action
Le produit comporte quelques anomalies.	En fonction de la criticité du produit, mener des actions correctives.
Le produit est relativement tolérant aux pannes.	En fonction de la criticité du produit, refactoriser le produit pour le rendre tolérant aux pannes.
Quelques indisponibilités sont encodées par le Service Desk mais pas toutes.	Demander au Service Desk d'encoder les indisponibilités.
Les valeurs mesurées et/ou les calculs d'évaluation sont erronés.	Décrire l'erreur à la cellule AQI.
Autre ?	Contactez la cellule AQI.
Interprétations possibles d'une valeur orange	Action
Le produit comporte beaucoup d'anomalies.	Mener des actions correctives.
Le produit est peu tolérant aux pannes.	Refactoriser le produit pour le rendre tolérant aux pannes.
Les valeurs mesurées et/ou les calculs d'évaluation sont erronés.	Décrire l'erreur à la cellule AQI.
Autre ?	Contactez la cellule AQI.
Interprétations possibles d'une valeur rouge	Action
Le produit comporte trop d'anomalies.	Mener des actions correctives prioritaires.
Le produit n'est pas assez tolérant aux pannes.	Refactoriser le produit pour le rendre tolérant aux pannes.
Une panne a eu lieu un vendredi en fin de journée et n'a pu être corrigée que le lundi en début de journée.	Consulter d'autres indicateurs et éventuellement améliorer la tolérance aux pannes.
Les valeurs mesurées et/ou les calculs d'évaluation sont erronés.	Décrire l'erreur à la cellule AQI.
Autre ?	Contactez la cellule AQI.

Modèle d'évaluation

Informations

Id	AM_20110003
Nom	Impact des Incidents
Statut	Expérimentation

Détails d'évaluation

```

wallImpactDesIncidents(e:evaluation, p:produit) : [!:decimal]
{
  Integer n = getNbIncidents(e, p)

  decimal t = CalculTotalIndisp(e, p)

  // i = Temps d'indisponibilité moyen après Incident
  i = t / n
}

```

Règles d'interprétation

Temps d'indisponibilité moyen après incident.

Intervalle

Code couleur	Intervalle de valeurs	Description
Vert	0 <= i < 15	Un incident provoque moins de 15 minutes d'indisponibilité en moyenne.
Jaune	15 <= i < 60	Un incident provoque moins d'une heure d'indisponibilité en moyenne.
Orange	60 <= i < 480	Un incident provoque moins de 8 heures d'indisponibilité en moyenne.
Rouge	480 <= i	Un incident provoque plus de 8 heures d'indisponibilité en moyenne.

Recommandations

Interprétations possibles d'une valeur verte	Action
Le produit comporte peu ou pas d'anomalie et/ou est très tolérant aux pannes.	Continuer comme ça !
La réactivité de l'équipe informatique du produit est très élevée.	Améliorer la tolérance aux pannes tout en respectant les procédures de résolution des incidents.
Les indisponibilités ne sont pas liées aux incidents.	Demander au Service Desk de lier les indisponibilités aux incidents les provoquant.
Le produit est obsolète et n'est plus utilisé.	Le faire supprimer de la liste des produits.
Les valeurs mesurées et/ou les calculs d'évaluation sont erronés.	Décrire l'erreur à la cellule AQI.
Autre ?	Contactez la cellule AQI.
Interprétations possibles d'une valeur jaune	Action
Le produit est relativement tolérant aux pannes.	En fonction de la criticité du produit, refactoriser le produit pour le rendre tolérant aux pannes.
La réactivité de l'équipe informatique du produit est élevée.	Améliorer la tolérance aux pannes tout en respectant les procédures de résolution des incidents.
Les indisponibilités ne sont pas toujours liées aux incidents.	Demander au Service Desk de lier les indisponibilités aux incidents les provoquant.
Les valeurs mesurées et/ou les calculs d'évaluation sont erronés.	Décrire l'erreur à la cellule AQI.
Autre ?	Contactez la cellule AQI.
Interprétations possibles d'une valeur orange	Action
Le produit est peu tolérant aux pannes.	Refactoriser le produit pour le rendre tolérant aux pannes.
Les valeurs mesurées et/ou les calculs d'évaluation sont erronés.	Décrire l'erreur à la cellule AQI.
Autre ?	Contactez la cellule AQI.
Interprétations possibles d'une valeur rouge	Action
Le produit n'est pas assez tolérant aux pannes.	Refactoriser le produit pour le rendre tolérant aux pannes.
Une panne a eu lieu un vendredi en fin de journée et n'a pu être corrigée que le lundi en début de journée.	Consulter d'autres indicateurs et éventuellement améliorer la tolérance aux pannes.
Les valeurs mesurées et/ou les calculs d'évaluation sont erronés.	Décrire l'erreur à la cellule AQI.
Autre ?	Contactez la cellule AQI.

Modèle d'évaluation

Informations

Id	AM_20110004
Nom	Fréquence de déploiement
Statut	Experimentation

Détails d'évaluation

```

evalFréquenceDeDéploiement(e:évaluation, p:produit) : (I1:decimal, I2:decimal)
{
  integer n = getNbRNP(e, p)
  integer a = CalculAge(e, p)
  // I1 = Nombre moyen de mises en production par jour depuis un an
  I1 = n / MIN(a, 365) * 365 / 12
  // I2 = Nombre moyen de jours entre 2 mises en production depuis un an
  if (r1 = 0)
    r2 = 365
  else
    r2 = 1 / (r1 * 12 / 365)
}

```

Règles d'interprétation

Nombre moyen de mises en production par jour depuis un an

Intervalle

Code couleur	Intervalle de valeurs	Description
Vert	0,00 <= i < 0,17	Moins d'un déploiement en production par semestre.
Jaune	0,17 <= i < 0,34	Moins d'un déploiement en production par trimestre.
Orange	0,34 <= i < 1,00	Moins d'un déploiement en production par mois.
Rouge	1,00 <= i	Plus d'un déploiement en production par mois.

Recommandations

Interprétations possibles d'une valeur verte	Action
Le produit est stable.	Continuer comme ça !
La fréquence des déploiements est planifiée et priorisée.	Continuer comme ça !
Les déploiements sont peu ou pas encodés par le Service Desk.	Demander au Service Desk d'encoder les déploiements.
Le produit est obsolète et n'est plus utilisé.	Le faire supprimer de la liste des produits.
Les valeurs mesurées et/ou les calculs d'évaluation sont erronés.	Décrire l'erreur à la cellule AQI.
Autre ?	Contactez la cellule AQI.
Interprétations possibles d'une valeur jaune	Action
Le produit est moyennement stable.	En fonction de la criticité du produit, mener des actions correctives.
La fréquence des déploiements est moyennement planifiée et priorisée.	En fonction de la criticité du produit, améliorer la planification et la priorisation des déploiements.
Les déploiements ne sont pas toujours encodés par le Service Desk.	Demander au Service Desk d'encoder les déploiements.
Les valeurs mesurées et/ou les calculs d'évaluation sont erronés.	Décrire l'erreur à la cellule AQI.
Autre ?	Contactez la cellule AQI.
Interprétations possibles d'une valeur orange	Action
Le produit est peu stable.	Mener des actions correctives.
Le produit est jeune.	Améliorer la planification et la priorisation des déploiements.
Les utilisateurs mettent la pression pour déployer les modifications au plus vite.	Sensibiliser les utilisateurs au coût et aux risques des déploiements.
Les valeurs mesurées et/ou les calculs d'évaluation sont erronés.	Décrire l'erreur à la cellule AQI.
Autre ?	Contactez la cellule AQI.
Interprétations possibles d'une valeur rouge	Action
Le produit n'est pas stable.	Mener des actions correctives prioritaires.
Le produit est très jeune.	Améliorer la planification et la priorisation des déploiements.
Les utilisateurs mettent la pression pour déployer les modifications au plus vite.	Sensibiliser les utilisateurs au coût et aux risques des déploiements, avec le soutien de la hiérarchie.
Les valeurs mesurées et/ou les calculs d'évaluation sont erronés.	Décrire l'erreur à la cellule AQI.
Autre ?	Contactez la cellule AQI.

Nombre moyen de jours entre 2 mises en production depuis un an

Intervalles

Code couleur	Intervalle de valeurs	Description
Vert	183 <=	Maximum un déploiement en production par semestre.
Jaune	092 <= < 183	Maximum un déploiement en production par trimestre.
Orange	031 <= < 092	Maximum un déploiement en production par mois.
Rouge	000 <= < 031	Plus d'un déploiement en production par mois.

Recommandations

Interprétations possibles d'une valeur verte	Action
Le produit est stable.	Continuer comme ça !
La fréquence des déploiements est planifiée et priorisée.	Continuer comme ça !
Les déploiements sont peu ou pas encodés par le Service Desk.	Demander au Service Desk d'encoder les déploiements.
Le produit est obsolète et n'est plus utilisé.	Le faire supprimer de la liste des produits.
Les valeurs mesurées et/ou les calculs d'évaluation sont erronés.	Décrire l'erreur à la cellule AQI.
Autre ?	Contactez la cellule AQI.
Interprétations possibles d'une valeur jaune	Action
Le produit est moyennement stable.	En fonction de la criticité du produit, mener des actions correctives.
La fréquence des déploiements est moyennement planifiée et priorisée.	En fonction de la criticité du produit, améliorer la planification et la priorisation des déploiements.
Les déploiements ne sont pas toujours encodés par le Service Desk.	Demander au Service Desk d'encoder les déploiements.
Les valeurs mesurées et/ou les calculs d'évaluation sont erronés.	Décrire l'erreur à la cellule AQI.
Autre ?	Contactez la cellule AQI.
Interprétations possibles d'une valeur orange	Action
Le produit est peu stable.	Mener des actions correctives.
Le produit est jeune.	Améliorer la planification et la priorisation des déploiements.
Les utilisateurs mettent la pression pour déployer les modifications au plus vite.	Sensibiliser les utilisateurs au coût et aux risques des déploiements.
Les valeurs mesurées et/ou les calculs d'évaluation sont erronés.	Décrire l'erreur à la cellule AQI.
Autre ?	Contactez la cellule AQI.
Interprétations possibles d'une valeur rouge	Action
Le produit n'est pas stable.	Mener des actions correctives prioritaires.
Le produit est très jeune.	Améliorer la planification et la priorisation des déploiements.
Les utilisateurs mettent la pression pour déployer les modifications au plus vite.	Sensibiliser les utilisateurs au coût et aux risques des déploiements, avec le soutien de la hiérarchie.
Les valeurs mesurées et/ou les calculs d'évaluation sont erronés.	Décrire l'erreur à la cellule AQI.
Autre ?	Contactez la cellule AQI.

Plan de mesure « Respect du cycle de vie »

Modèle d'évaluation

Informations

Id	AM_2012001
Nom	Validation de l'analyse
Statut	Expérimentation

Détails d'évaluation

```
enumValid = {0=Inexistent, 1=Non évalué, 2=Inacceptable, 3=Insuffisant, 4=Satisfaisant, 5=Bon}
enumVal = {0=Aucun, 1=Inacceptable, 2=Insuffisant, 3=Satisfaisant, 4=Bon}

evalValidationAnalyse(e:évaluation, p:produit) : (i:enumValid)
{
    boolean ex = getExistenceAF(e, p)

    if (ex = false) {
        i = Inexistent
    }
    else {
        enumVal cds = getValAFcds(e, p)
        enumVal arch = getValAFarch(e, p)
        enumVal test = getValAFtest(e, p)

        Integer x = cds + arch + test

        if (x == 0) { i = Non évalué }
        if (x == 1) { i = Inacceptable }
        if (x == 2) { i = Insuffisant }
        if (x == 3) { i = Satisfaisant }
        if (x == 4) { i = Bon }
    }
}
```

Règles d'interprétation

Evaluation de l'analyse par les intervenants principaux

Intervalle

Code couleur	Intervalle de valeurs	Description
Vert	4 = Bon	Les évaluations donnent un bon résultat
Jaune	3 = Satisfaisant	Les évaluations donnent un résultat satisfaisant
Orange	2 = Insuffisant	Les évaluations donnent un résultat insuffisant
Rouge	1 = Inacceptable	Les évaluations donnent un résultat inacceptable
Gris	0 = Non évalué ou 1 = Inexistent	Le résultat des évaluations n'est pas pertinent

Recommandations

Interprétations possibles d'une valeur verte	Action
L'évaluation par les intervenants principaux est bonne.	Continuer comme ça !
Les valeurs mesurées et/ou les calculs d'évaluation sont erronés.	Décrire l'erreur à la cellule AQI.
Autre ?	Contactez la cellule AQI.
Interprétations possibles d'une valeur jaune	Action
L'évaluation par les intervenants principaux est satisfaisante.	Contactez les intervenants pour identifier éventuellement les problèmes.
Les valeurs mesurées et/ou les calculs d'évaluation sont erronés.	Décrire l'erreur à la cellule AQI.
Autre ?	Contactez la cellule AQI.
Interprétations possibles d'une valeur orange	Action
L'évaluation par les intervenants principaux est insuffisante.	Contactez les intervenants pour identifier les problèmes.
Les valeurs mesurées et/ou les calculs d'évaluation sont erronés.	Décrire l'erreur à la cellule AQI.
Autre ?	Contactez la cellule AQI.
Interprétations possibles d'une valeur rouge	Action
L'évaluation par les intervenants principaux est inacceptable.	Contactez d'urgence les intervenants pour identifier les problèmes.
Les valeurs mesurées et/ou les calculs d'évaluation sont erronés.	Décrire l'erreur à la cellule AQI.
Autre ?	Contactez la cellule AQI.
Interprétations possibles d'une valeur grise	Action
La phase de réalisation de l'analyse n'a pas encore eu lieu.	Garder cette phase en tête et la réaliser au moment opportun.
L'évaluation par les intervenants n'a pas eu lieu.	Demander une évaluation par les intervenants.
L'analyse fonctionnelle n'existe pas.	Réaliser une analyse fonctionnelle.
Les valeurs mesurées et/ou les calculs d'évaluation sont erronés.	Décrire l'erreur à la cellule AQI.
Autre ?	Contactez la cellule AQI.

Modèle d'évaluation

Informations

Id	AM_20120002
Nom	Validation de l'architecture
Statut	Expérimentation

Détails d'évaluation

```

enumval = {0=Inexistant, 1=Non évalué, 2=Inacceptable, 3=Insuffisant, 4=Satisfaisant, 5=Bon}
enumval = {0=aucun, 1=Inacceptable, 2=Insuffisant, 3=Satisfaisant, 4=Bon}

evalValidationArchitecture(w:evaluation, p:produit) : (I:enumval)
{
  boolean ex = getExistenceAC(w, p);
  if (ex = false) {
    I = Inexistant
  }
  else {
    enumval ans = getvalACAns(w, p)
    enumval car = getvalACCar(w, p)
    enumval tech = getvalACGnaTech(w, p)

    Integer x = ans + car + tech
    enumval ac

    if (x = 0) { ac = Non évalué }
    if (x <= 3) { ac = Inacceptable }
    if (x <= 6) { ac = Insuffisant }
    if (x <= 9) { ac = Satisfaisant }
    if (x >= 10) { ac = Bon }
  }

  ex = getExistenceAT(w, p)

  if (ex = false) {
    if (ac > 2) { I = ac - 1 }
    if (ac = 2) { I = Inacceptable }
    I = Inexistant
  }
  else {
    enumval arch = getvalATArch(w, p)
    enumval car = getvalATCar(w, p)
    enumval dev = getvalATDev(w, p)

    Integer x = arch + car + dev
    enumval at

    if (x = 0) { at = Non évalué }
    if (x <= 3) { at = Inacceptable }
    if (x <= 6) { at = Insuffisant }
    if (x <= 9) { at = Satisfaisant }
    if (x >= 10) { at = Bon }

    I = (ac + at) / 2
  }
}

```

Règles d'interprétation

Evaluation de l'architecture par les intervenants principaux

Intervalle

Code couleur	Intervalle de valeurs	Description
Vert	I = Bon	Les évaluations donnent un bon résultat
Jaune	I = Satisfaisant	Les évaluations donnent un résultat satisfaisant
Orange	I = Insuffisant	Les évaluations donnent un résultat insuffisant
Rouge	I = Inacceptable	Les évaluations donnent un résultat inacceptable
Gris	I = Non évalué ou I = Inexistant	Le résultat des évaluations n'est pas pertinent

Recommandations

Interprétations possibles d'une valeur verte	Action
L'évaluation par les intervenants principaux est bonne.	Continuer comme ça !
Les valeurs mesurées et/ou les calculs d'évaluation sont erronés.	Décrire l'erreur à la cellule AQI.
Autre ?	Contactez la cellule AQI.
Interprétations possibles d'une valeur jaune	Action
L'évaluation par les intervenants principaux est satisfaisante.	Contactez les intervenants pour identifier éventuellement les problèmes.
Les valeurs mesurées et/ou les calculs d'évaluation sont erronés.	Décrire l'erreur à la cellule AQI.
Autre ?	Contactez la cellule AQI.
Interprétations possibles d'une valeur orange	Action
L'évaluation par les intervenants principaux est insuffisante.	Contactez les intervenants pour identifier les problèmes.
Les valeurs mesurées et/ou les calculs d'évaluation sont erronés.	Décrire l'erreur à la cellule AQI.
Autre ?	Contactez la cellule AQI.
Interprétations possibles d'une valeur rouge	Action
L'évaluation par les intervenants principaux est inacceptable.	Contactez d'urgence les intervenants pour identifier les problèmes.
L'analyse de conception n'existe pas alors que l'analyse technique existe.	Réaliser une analyse de conception.
Les valeurs mesurées et/ou les calculs d'évaluation sont erronés.	Décrire l'erreur à la cellule AQI.
Autre ?	Contactez la cellule AQI.
Interprétations possibles d'une valeur grise	Action
La phase de réalisation de l'architecture n'a pas encore eu lieu.	Garder cette phase en tête et la réaliser au moment opportun.
L'évaluation par les intervenants n'a pas eu lieu.	Demander une évaluation par les intervenants.
L'analyse de conception n'existe pas.	Réaliser une analyse de conception.
L'analyse technique n'existe pas.	Réaliser une analyse technique.
Les valeurs mesurées et/ou les calculs d'évaluation sont erronés.	Décrire l'erreur à la cellule AQI.
Autre ?	Contactez la cellule AQI.

Modèle d'évaluation

Informations

Id	AM_2012003
Nom	Validation du développement
Statut	Experimentation

Détails d'évaluation

```

enumValid = {0=Inexistent, 1=Non évalué, 2=Inacceptable, 3=Insuffisant, 4=Satisfaisant, 5=Bon}
enumVal = {0=aucun, 1=Inacceptable, 2=Insuffisant, 3=Satisfaisant, 4=Bon}

evalValidationDevelopment(s=évaluation, p=produit) : I(enumValid)
{
  boolean ex = getExistenceCode(s, p)

  if (ex = false) {
    i = Inexistent
  }
  else {
    enumVal tech = getValCodeAnaTech(s, p)
    enumVal dev = getValCodeCdev(s, p)

    Integer x = tech + dev

    if (x = 0) { i = Non évalué }
    if (x <= 2) { i = Inacceptable }
    if (x <= 3) { i = Insuffisant }
    if (x <= 4) { i = Satisfaisant }
    if (x >= 5) { i = Bon }
  }
}

```

Règles d'interprétation

Évaluation du développement par les intervenants principaux

Intervalle

Code couleur	Intervalle de valeurs	Description
Vert	4 = Bon	Les évaluations donnent un bon résultat
Jaune	3 = Satisfaisant	Les évaluations donnent un résultat satisfaisant
Orange	2 = Insuffisant	Les évaluations donnent un résultat insuffisant
Rouge	1 = Inacceptable	Les évaluations donnent un résultat inacceptable
Gris	0 = Non évalué ou 1 = Inexistent	Le résultat des évaluations n'est pas pertinent

Recommandations

Interprétations possibles d'une valeur verte	Action
L'évaluation par les intervenants principaux est bonne.	Continuer comme ça !
Les valeurs mesurées et/ou les calculs d'évaluation sont erronés.	Décrire l'erreur à la cellule AQI.
Autre ?	Contactez la cellule AQI.
Interprétations possibles d'une valeur jaune	Action
L'évaluation par les intervenants principaux est satisfaisante.	Contactez les intervenants pour identifier éventuellement les problèmes.
Les valeurs mesurées et/ou les calculs d'évaluation sont erronés.	Décrire l'erreur à la cellule AQI.
Autre ?	Contactez la cellule AQI.
Interprétations possibles d'une valeur orange	Action
L'évaluation par les intervenants principaux est insuffisante.	Contactez les intervenants pour identifier les problèmes.
Les valeurs mesurées et/ou les calculs d'évaluation sont erronés.	Décrire l'erreur à la cellule AQI.
Autre ?	Contactez la cellule AQI.
Interprétations possibles d'une valeur rouge	Action
L'évaluation par les intervenants principaux est inacceptable.	Contactez d'urgence les intervenants pour identifier les problèmes.
Les valeurs mesurées et/ou les calculs d'évaluation sont erronés.	Décrire l'erreur à la cellule AQI.
Autre ?	Contactez la cellule AQI.
Interprétations possibles d'une valeur grise	Action
La phase de réalisation du développement n'a pas encore eu lieu.	Garder cette phase en tête et la réaliser au moment opportun.
L'évaluation par les intervenants n'a pas eu lieu.	Demander une évaluation par les intervenants.
Le code n'existe pas.	Réaliser le code.
Les valeurs mesurées et/ou les calculs d'évaluation sont erronés.	Décrire l'erreur à la cellule AQI.
Autre ?	Contactez la cellule AQI.

Modèle d'évaluation

Informations

Id	AM_20120004
Nom	Validation des tests
Statut	Expérimentation

Détails d'évaluation

```

enumValid = {0=Inexistant, 1=Non évalué, 2=Inacceptable, 3=Insuffisant, 4=Satisfaisant, 5=Bon}
enumVal = {0=Aucune, 1=Inacceptable, 2=Insuffisant, 3=Satisfaisant, 4=Bon}

wvalValidationTests(w:evaluation, p:produit) : (i:wenumValid)
{
    boolean ex = getExistenceTu(w, p)
    if (ex = false) {
        i = Inexistant
    }
    else {
        enumVal tech = getValTUestech(w, p)
        enumVal sql = getValTUsql(w, p)

        Integer x = tech + sql
        enumValid tu

        if (x = 0) { tu = Non évalué }
        if (x <= 2) { tu = Inacceptable }
        if (x <= 4) { tu = Insuffisant }
        if (x <= 6) { tu = Satisfaisant }
        if (x >= 7) { tu = Bon }
    }

    ex = getExistenceTV(w, p)
    if (ex = false) {
        if (tu > 2) { i = tu - 1 }
        if (tu = 2) { i = Inacceptable }
        i = Inexistant
    }
    else {
        enumVal ans = getValTVansFonc(w, p)
        enumVal sql = getValTVsql(w, p)

        Integer x = ans + sql
        enumValid tv

        if (x = 0) { tv = Non évalué }
        if (x <= 2) { tv = Inacceptable }
        if (x <= 4) { tv = Insuffisant }
        if (x <= 6) { tv = Satisfaisant }
        if (x >= 7) { tv = Bon }
    }

    i = (tu + tv) / 2
}
}

```

Règles d'interprétation

Evaluation des tests par les intervenants principaux

Intervalle

Code couleur	Intervalle de valeurs	Description
Vert	4 = Bon	Les évaluations donnent un bon résultat
Jaune	3 = Satisfaisant	Les évaluations donnent un résultat satisfaisant
Orange	2 = Insuffisant	Les évaluations donnent un résultat insuffisant
Rouge	1 = Inacceptable	Les évaluations donnent un résultat inacceptable
Gris	0 = Non évalué ou 5 = Inexistant	Le résultat des évaluations n'est pas pertinent

Recommandations

Interprétations possibles d'une valeur verte	Action
L'évaluation par les intervenants principaux est bonne.	Continuer comme ça !
Les valeurs mesurées et/ou les calculs d'évaluation sont erronés.	Décrire l'erreur à la cellule AQI.
Autre ?	Contactez la cellule AQI.
Interprétations possibles d'une valeur jaune	Action
L'évaluation par les intervenants principaux est satisfaisante.	Contactez les intervenants pour identifier éventuellement les problèmes.
Les valeurs mesurées et/ou les calculs d'évaluation sont erronés.	Décrire l'erreur à la cellule AQI.
Autre ?	Contactez la cellule AQI.
Interprétations possibles d'une valeur orange	Action
L'évaluation par les intervenants principaux est insuffisante.	Contactez les intervenants pour identifier les problèmes.
Les valeurs mesurées et/ou les calculs d'évaluation sont erronés.	Décrire l'erreur à la cellule AQI.
Autre ?	Contactez la cellule AQI.
Interprétations possibles d'une valeur rouge	Action
L'évaluation par les intervenants principaux est inacceptable.	Contactez d'urgence les intervenants pour identifier les problèmes.
Les valeurs mesurées et/ou les calculs d'évaluation sont erronés.	Décrire l'erreur à la cellule AQI.
Autre ?	Contactez la cellule AQI.
Interprétations possibles d'une valeur grise	Action
La phase de réalisation des tests n'a pas encore eu lieu.	Garder cette phase en tête et la réaliser au moment opportun.
L'évaluation par les intervenants n'a pas eu lieu.	Demander une évaluation par les intervenants.
Les tests unitaires n'existent pas.	Réaliser des tests unitaires.
Les tests de validation n'existent pas.	Réaliser des tests de validation.
Les valeurs mesurées et/ou les calculs d'évaluation sont erronés.	Décrire l'erreur à la cellule AQI.
Autre ?	Contactez la cellule AQI.

Plan de mesures transversal

Concepts

Evaluation

Type		Propriétaire MoCQA: ce concept est présenté uniquement pour indiquer les attributs d'évaluation utilisables dans les fonctions
Description		Evaluation d'un panel de produits pour un panel d'indicateurs
Colonnes	Date	Date de l'évaluation

Produit

Type		Artefact	
Description		Application, progiciel, service,... développé(e) par et/ou utilisé(e) au SPI	
Exemples		AirPac, Métagré, VisualProjet, Silver, Time, Jira, Service de publipostage,...	
Source		Fichier Produits.xls sous la responsabilité de Laurent Vanhalle	
Attributs de base	Code	Code unique au travers de l'ensemble des systèmes	Code OFFICIEL
	Création	Date de création du produit	Création
Attributs dérivés	Age	<p><u>Id:</u> F_20110001 <u>Description:</u> Age du produit en jours</p> <pre> calcAge(e:Evaluation, p:Produit) : integer { date d = getCréation(p) r = e.date - d return r } </pre>	
	DuréeIndisp	<p><u>Id:</u> F_20110003 <u>Description:</u> Durée totale d'indisponibilité</p> <pre> calcDuréeIndisp(e:Evaluation, p:Produit, coll_i:Incident[]) : decimal { r = 0 for each i in coll_i { r = r + i.Durée } return r } </pre>	

Incident

Type		Comportement	
Description		Plainte, bogue, ... remonté(e) au Service Desk	
Exemples		"Un calcul Airpac n'est pas correct", "Time n'est pas accessible le matin", "Silver est lent", ...	
Source		Serveur SP1200V - Table TASKS	
Attributs de base	Id	Identifiant unique d'un incident	WO_NUM
	CodeProduit	Code du produit concerné par l'incident	LOOKUP1
	Début	Date et heure d'ouverture de l'incident	OPENDATE
	Fin	Date et heure de clôture de l'incident	CLSDDATE
	Indisponibilité	Indique si l'incident provoque une indisponibilité ('Oui') ou pas ('Non')	TASKLOOKUP6
Attributs dérivés	DuréeIndisp	Id: F_20110002 Description: Durée de l'indisponibilité en heures provoquée par l'incident	
		<pre> calcDurée(i:Incident) : decimal { if (i.Indisponibilité = 'Oui') return i.Fin - i.Début else return 0 } </pre>	

Déploiement

Type		Comportement	
Description		Déploiement dans un environnement particulier	
Exemples		"Mise en production d'une nouvelle version de SIG", "Mise en test d'une nouvelle version de NPP", ...	
Source		Outil Silver - Bilan des distributions	
Attributs de base	Id	Identifiant unique d'un déploiement	N° FDD
	CodeProduit	Code du produit concerné par le déploiement	Produit
	Environnement	Environnement où le déploiement a lieu (Prod,Test,Acceptation)	Environnement
Attributs dérivés			

Analyse fonctionnelle

Type		Artéfact	
Description		Analyse fonctionnelle d'un produit développé	
Exemples		"Analyse fonctionnelle de BUD", ...	
Source		Outil Confluence - Page Application	
Attributs de base	CodeProduit	Code du produit concerné par l'analyse fonctionnelle	
	ExistenceAF	Existence d'une analyse fonctionnelle	Fichier <CodeProduit>_AF.doc
	EvalAFCda	Evaluation de l'analyse fonctionnelle par la Cellule des Analystes	Fichier <CodeProduit>_AF_EvalCda.xls
	EvalAFArch	Evaluation de l'analyse fonctionnelle par les architectes du produit	Fichier <CodeProduit>_AF_EvalArch.xls
	EvalAFTest	Evaluation de l'analyse fonctionnelle par les testeurs fonctionnels du produit	Fichier <CodeProduit>_AF_EvalTest.xls
Attributs dérivés			

Analyse de conception

Type		Artéfact	
Description		Analyse de conception d'un produit développé	
Exemples		"Analyse de conception de ELF", ...	
Source		Outil Confluence - Page Application	
Attributs de base	CodeProduit	Code du produit concerné par l'analyse de conception	
	ExistenceAC	Existence d'une analyse de conception	Fichier <CodeProduit>_AC.doc
	EvalACAnaFonc	Evaluation de l'analyse de conception par les analystes fonctionnels du produit	Fichier <CodeProduit>_AC_EvalAnaFonc.xls
	EvalACCar	Evaluation de l'analyse de conception par la Cellule Architecture	Fichier <CodeProduit>_AC_EvalCar.xls
	EvalACAnaTech	Evaluation de l'analyse de conception par les analystes techniques du produit	Fichier <CodeProduit>_AC_EvalAnaTech.xls
Attributs dérivés			

Analyse technique

Type		Artéfact	
Description		Analyse technique d'un produit développé	
Exemples		"Analyse technique de ELF", ...	
Source		Outil Confluence - Page Application	
Attributs de base	CodeProduit	Code du produit concerné par l'analyse technique	
	ExistenceAT	Existence d'une analyse technique	Fichier <CodeProduit>_AT.doc
	EvalATArch	Evaluation de l'analyse technique par les architectes du produit	Fichier <CodeProduit>_AT_EvalArch.xls
	EvalATCar	Evaluation de l'analyse technique par la Cellule Architecture	Fichier <CodeProduit>_AT_EvalCar.xls
	EvalATDev	Evaluation de l'analyse technique par les développeurs du produit	Fichier <CodeProduit>_AT_EvalDev.xls
Attributs dérivés			

Code

Type		Artéfact	
Description		Code d'un produit développé	
Exemples		"Code PowerBuilder de BUD", "Code Java de Gesdos", ...	
Source		Outil CVS - Branche (a) Outil Confluence - Page Application (b)	
Attributs de base	CodeProduit	Code du produit concerné par le code	(a) <Branche>
	ExistenceCode	Existence du code	(a) <CodeProduit>*.*
	EvalCodeAnaTech	Evaluation du code par les analystes techniques du produit	(b) Fichier <CodeProduit>_Code_EvalAnaTech.xls
	EvalCodeCdev	Evaluation du code par la Cellule des Développeurs	(b) Fichier <CodeProduit>_Code_EvalCdev.xls
Attributs dérivés			

Tests unitaires

Type		Artéfact	
Description		Tests unitaires d'un produit développé	
Exemples		"Tests unitaires de BUD", ...	
Source		Outil CVS - Branche (a) Outil Confluence - Page Application (b)	
Attributs de base	CodeProduit	Code du produit concerné par les tests unitaires	(a) <Branche>
	ExistenceTU	Existence de tests unitaires	(a) <CodeProduit>\TU*.*
	EvalTUAnaTech	Evaluation des tests unitaires par les analystes techniques du produit	(b) Fichier <CodeProduit>_TU_EvalAnaTech.xls
	EvalTUAqi	Evaluation des tests unitaires par la Cellule AQI	(b) Fichier <CodeProduit>_TU_EvalAqi.xls
Attributs dérivés			

Tests de validation

Type		Artéfact	
Description		Tests de validation d'un produit développé	
Exemples		"Tests de validation fonctionnelle de BUD", "Tests de validation système de ELF", ...	
Source		Outil Confluence - Page Application	
Attributs de base	CodeProduit	Code du produit concerné par les tests unitaires	<CodeProduit>
	ExistenceTV	Existence de tests unitaires	Fichier <CodeProduit>_TV.doc
	EvalTVAnaFonc	Evaluation des tests de validation par les analystes fonctionnels du produit	Fichier <CodeProduit>_TV_EvalAnaFonc.xls
	EvalTVAqi	Evaluation des tests de validation par la Cellule AQI	Fichier <CodeProduit>_TV_EvalAqi.xls
Attributs dérivés			

A2.4 Résultats d'évaluation

Ci-dessous les résultats d'évaluation pour les indicateurs « Fiabilité ». A noter que le nom des produits a été modifié pour des raisons de confidentialité.

Evolution	FIABILITE Produit	Mesures			Fonctions			Survenance des incidents (incidents par jour)	Disponibilité (%)	Impact des incidents (minutes)	Indicateurs Qualité (mises en production par jour)	Fréquence de déploiement (jours entre 2 mises en production)
		NbrIncidents (incidents)	Date1sMEP	NbrMEP (mises en production)	Age (jours)	DureeIndsp (jours)	TotalIndsp (minutes)					
16/11/2011	Produit 1	3	31/12/2004	3	2511	0,00	0	0,02	100,0%	0	0,02	61
16/11/2011	Produit 2	0	31/12/2004	7	2511	0,00	0	0,00	100,0%	0	0,04	26
16/11/2011	Produit 3	2	23/03/2010	4	603	0,00	0	0,01	100,0%	0	0,02	46
16/11/2011	Produit 4	0	09/09/1999	0	4451	0,00	0	0,00	100,0%	0	0,00	366
16/11/2011	Produit 5	1	09/09/1999	0	4451	0,00	0	0,01	100,0%	0	0,00	366
16/11/2011	Produit 6	2	09/09/1999	0	4451	0,00	0	0,01	100,0%	0	0,00	366
16/11/2011	Produit 7	0	09/09/1999	1	4451	0,00	0	0,00	100,0%	0	0,01	183
16/11/2011	Produit 8	0	09/09/1999	2	4451	0,00	0	0,00	100,0%	0	0,01	92
16/11/2011	Produit 9	0	09/09/1999	1	4451	0,00	0	0,00	100,0%	0	0,01	183
16/11/2011	Produit 10	0	09/09/1999	1	4451	0,00	0	0,00	100,0%	0	0,01	183
16/11/2011	Produit 11	2	09/09/1999	2	4451	0,00	0	0,01	100,0%	0	0,01	92
16/11/2011	Produit 12	0	09/09/1999	2	4451	0,00	0	0,00	100,0%	0	0,01	92
16/11/2011	Produit 13	34	09/09/1999	10	4451	0,00	0	0,19	100,0%	0	0,05	18
16/11/2011	Produit 14	0	09/09/1999	4	4451	0,00	0	0,00	100,0%	0	0,02	46
16/11/2011	Produit 15	7	09/09/1999	0	4451	0,00	0	0,01	100,0%	0	0,00	366
16/11/2011	Produit 16	2	01/06/2005	1	2359	0,00	0	0,01	100,0%	0	0,01	183
16/11/2011	Produit 17	1	31/12/2006	1	1781	0,00	0	0,01	100,0%	0	0,01	183
16/11/2011	Produit 18	1	30/12/2009	1	686	0,00	0	0,01	100,0%	0	0,01	183
16/11/2011	Produit 19	1	18/04/2006	4	2038	0,00	0	0,01	100,0%	0	0,02	46
16/11/2011	Produit 20	40	08/04/2005	6	2413	0,00	0	0,22	100,0%	0	0,03	31
16/11/2011	Produit 21	3	31/12/2004	0	2511	0,00	0	0,02	100,0%	0	0,00	366
16/11/2011	Produit 22	0	09/09/1999	0	4451	0,00	0	0,00	100,0%	0	0,00	366
16/11/2011	Produit 23	4	09/09/1999	0	4451	0,00	0	0,02	100,0%	0	0,00	366
16/11/2011	Produit 24	0	09/09/1999	0	4451	0,00	0	0,00	100,0%	0	0,00	366
16/11/2011	Produit 25	2	14/09/2009	0	793	0,00	0	0,01	100,0%	0	0,00	366
16/11/2011	Produit 26	2	10/05/2005	1	2381	0,00	0	0,01	100,0%	0	0,01	183
16/11/2011	Produit 27	0	09/09/1999	1	4451	0,00	0	0,00	100,0%	0	0,01	183
16/11/2011	Produit 28	0	09/09/1999	1	4451	0,00	0	0,00	100,0%	0	0,01	183
16/11/2011	Produit 29	2	31/12/2004	0	2511	0,00	0	0,01	100,0%	0	0,00	366
16/11/2011	Produit 30	0	09/09/1999	7	4451	0,00	0	0,00	100,0%	0	0,04	26
16/11/2011	Produit 31	0	09/09/1999	1	4451	0,00	0	0,00	100,0%	0	0,01	183
16/11/2011	Produit 32	0	09/09/1999	1	4451	0,00	0	0,00	100,0%	0	0,01	183
16/11/2011	Produit 33	0	09/09/1999	2	4451	0,00	0	0,00	100,0%	0	0,01	92
16/11/2011	Produit 34	0	09/09/1999	1	4451	0,00	0	0,00	100,0%	0	0,01	183
16/11/2011	Produit 35	0	09/09/1999	1	4451	0,00	0	0,00	100,0%	0	0,01	183
16/11/2011	Produit 36	0	09/09/1999	2	4451	0,00	0	0,00	100,0%	0	0,01	92
16/11/2011	Produit 37	0	09/09/1999	3	4451	0,00	0	0,00	100,0%	0	0,02	61
16/11/2011	Produit 38	0	09/09/1999	1	4451	0,00	0	0,00	100,0%	0	0,01	183
16/11/2011	Produit 39	0	09/09/1999	2	4451	0,00	0	0,00	100,0%	0	0,01	92
16/11/2011	Produit 40	46	14/03/2005	3	2438	0,00	0	0,25	100,0%	0	0,02	61
16/11/2011	Produit 41	0	09/09/1999	1	4451	0,00	0	0,00	100,0%	0	0,01	183
16/11/2011	Produit 42	3	07/12/2005	4	2170	0,00	0	0,02	100,0%	0	0,02	46
16/11/2011	Produit 43	0	09/09/1999	1	4451	0,00	0	0,00	100,0%	0	0,01	183
16/11/2011	Produit 44	166	09/09/1999	0	4451	0,00	0	0,91	100,0%	0	0,00	366
16/11/2011	Produit 45	0	10/05/2005	1	2381	0,00	0	0,00	100,0%	0	0,01	183
16/11/2011	Produit 46	0	28/03/2007	3	1694	0,00	0	0,00	100,0%	0	0,02	61
16/11/2011	Produit 47	0	09/09/1999	0	4451	0,00	0	0,00	100,0%	0	0,00	366
16/11/2011	Produit 48	0	29/05/2009	2	901	0,00	0	0,00	100,0%	0	0,01	92
16/11/2011	Produit 49	5	31/12/2004	1	2511	0,00	0	0,03	100,0%	0	0,01	183
16/11/2011	Produit 50	17	09/09/1999	1	4451	0,00	0	0,09	100,0%	0	0,01	183
16/11/2011	Produit 51	101	09/09/1999	20	4451	0,00	0	0,55	100,0%	0	0,11	9
16/11/2011	Produit 52	0	09/09/1999	3	4451	0,00	0	0,00	100,0%	0	0,02	61
16/11/2011	Produit 53	0	31/12/2004	3	2511	0,00	0	0,00	100,0%	0	0,02	61
16/11/2011	Produit 54	13	09/09/1999	0	4451	0,00	0	0,07	100,0%	0	0,00	366
16/11/2011	Produit 55	7	09/09/1999	1	4451	0,00	0	0,04	100,0%	0	0,01	183
16/11/2011	Produit 56	0	29/05/2009	5	901	0,00	0	0,00	100,0%	0	0,03	37
	MINIMUM							0,00	100,0%	0,00	0,00	9
	MOYENNE							0,05	100,0%	0,00	0,01	174
	MAXIMUM							0,91	100,0%	0,00	0,11	366

A2.5 Plans d'actions

Id	Evaluation	Création	Modification	Indicateur	Constatation	Ticket JIRA
1	16/11/2011	16/12/2011	16/12/2011	Fiabilité	Dans la collection d'incidents, certains incidents ne sont liés à aucun produit.	AQI-118
2	16/11/2011	16/12/2011	16/12/2011	Fiabilité	On ne dispose des données d'incidents que depuis le 17/05/2011.	AQI-119
3	16/11/2011	16/12/2011	16/12/2011	Fiabilité	La date de première mise en production des applications est inconnue.	SPI-3
4	16/11/2011	16/12/2011	16/12/2011	Fiabilité	La date de première mise en production des applications est inconnue.	AQI-120
5	16/11/2011	16/12/2011	16/12/2011	Fiabilité	On ne dispose pas d'un lien éventuel entre un incident et une indisponibilité.	SPI-4
6	16/11/2011	16/12/2011	16/12/2011	Fiabilité	On ne dispose pas d'un lien éventuel entre un incident et une indisponibilité.	AQI-121
7	16/11/2011	16/12/2011	16/12/2011	Fiabilité	L'identifiant d'un même produit dans les collections d'incidents, d'indisponibilités et de mises en production n'est pas toujours identique. De même, les produits ne remontant ni incident, ni indisponibilité, ni mise en production n'apparaissent pas dans la liste.	SPI-5
8	16/11/2011	16/12/2011	16/12/2011	Fiabilité	Chaque formule reprenant MAX(Age,365) devrait être MIN(Age,365).	AQI-122
9	16/11/2011	16/12/2011	16/12/2011	Fiabilité	Aucune règle d'interprétation n'a été définie.	AQI-123
10	16/11/2011	16/12/2011	16/12/2011	Fiabilité	Les durées tiennent compte des heures non prestées (week-end, congés, nuits,...).	AQI-124
11	16/11/2011	16/12/2011	16/12/2011	Fiabilité	Un indice de "fiabilité" globale, évalué sur base des 4 sous-indicateurs, serait un plus.	AQI-125
12	16/11/2011	16/12/2011	16/12/2011	Fiabilité	L'indicateur "Disponibilité" est exprimé de 0 à 1 alors que l'unité est le %.	AQI-126
13	16/11/2011	16/12/2011	16/12/2011	Fiabilité	L'indicateur "Fréquence de déploiement" n'est pas très significatif.	AQI-127
14	16/11/2011	16/12/2011	16/12/2011	Fiabilité	En règle générale, les échelles utilisées pour les indicateurs paraissent étranges et peu significatives.	AQI-128

Résultats du filtre : Actions suite à l'exploitation des indicateurs qualité produits

T	Clé	Résumé	Pr +	Attribution	Echéance	Etat
	SPI-5	[IQP] Fiabilité - Evaluation - Uniformiser la liste des produits		Laurent VANHALLE		Affectée
	AQI-118	[IQP] Fiabilité - Opérationnalisation - Adapter la requête sur les incidents		Samuel HANOTEAU	30/12/2011	Affectée
	AQI-127	[IQP] Fiabilité - Modélisation - Indicateur de déploiement		Samuel HANOTEAU	30/12/2011	Affectée
	AQI-128	[IQP] Fiabilité - Modélisation - Nouvelles échelles		Samuel HANOTEAU	30/12/2011	Affectée
	SPI-3	[IQP] Fiabilité - Evaluation - Encoder les dates de première MEP		Pierre Girboux		En cours
	AQI-119	[IQP] Fiabilité - Opérationnalisation - Tenir compte de la date du nouveau système d'incidents		Samuel HANOTEAU	16/01/2012	Affectée
	AQI-120	[IQP] Fiabilité - Opérationnalisation - Fixer les dates de MEP inconnues au 1/01/2011		Samuel HANOTEAU	16/01/2012	Affectée
	AQI-121	[IQP] Fiabilité - Opérationnalisation - Tenir compte de la mise en service du lien incident/indisponibilité		Samuel HANOTEAU	16/01/2012	Affectée
	AQI-124	[IQP] Fiabilité - Modélisation - Indicateur sans les jours ouvrables		Samuel HANOTEAU	16/01/2012	Affectée
	AQI-125	[IQP] Fiabilité - Modélisation - Indice de fiabilité globale		Samuel HANOTEAU	16/01/2012	Affectée
	SPI-4	[IQP] Fiabilité - Evaluation - Lien entre indisponibilité et incident		Pierre Girboux		Affectée

Affichage des demandes 1 à 11 des demandes correspondantes11.

Annexe 3 : Coûts du projet

On présente ici les coûts du projet de mise en place des indicateurs de qualité, prestations internes au SPI et individuelles de l'étudiant confondues.

Activité	Coût ¹³
Recherche de documentation	12 J.H
Prise de connaissance de l'approche MoCQA	14 J.H
Cycle de développement au format MoCQA	3 J.H
Acquisition	3 J.H
Fiabilité	
Modélisation	3 J.H
Opérationnalisation	3 J.H
Evaluation	1 J.H
Exploitation	1 J.H
Cycle de vie	
Modélisation	3 J.H
Plan de mesure	2 J.H
Gestion de projet	9 J.H
Réunions	3 J.H
Communications	5 J.H
Guide de déploiement	3 J.H
Rédaction du mémoire	11 J.H
TOTAL :	76 J.H

¹³ 1 J.H = 7,6 Heures productives

Annexe 4 : Comptes rendus de réunions

On reprend ici l'ensemble des comptes rendus des réunions entre l'étudiant et ses (co-)promoteurs.



QUALITE_PRODUIITS Compte rendu de réunion

Classification
Sensible

Version
1.0

Date
18/07/2011

Page
1 sur 2



Informations

Sujet : Réunion d'avancement
Date : 18/07/2011
Auteur : S. Hanoteau
Participants : N. Habra ; B. Vanderose ; S. Hanoteau
Excusés :
Distribution : Participants ; F. Dermine

1 Avant propos

Les parties du présent compte rendu concernant directement les FUNDP sont en couleur **bleue**, puisque ne concernant pas la DGARNE. Le compte rendu publié sur l'espace projet public de la DGARNE ne contiendra pas ces parties.

2 Ordre du jour

- Mise en contact entre B. Vanderose et S. Hanoteau ;
- Assistance pour le début des activités ;
- Proposition de sujet de mémoire.

3 Contenu

- Proposition de sujet de mémoire MIHD-60 :
 - S. Hanoteau propose un document dans le but d'officialiser le mémoire.
 - Le mot « expérimentation » devrait plutôt être remplacé par « déploiement ».
 - La mention de l'environnement hétérogène est à réfléchir : est-ce vraiment l'objectif du mémoire ?
 - Pour le reste, notamment les objectifs, chacun marque son accord.
 - S. Hanoteau modifiera le document en fonction des remarques et l'enverra à nouveau pour validation.
- Etat du framework MoCQA :
 - Le framework MoCQA a été principalement testé en environnement universitaire jusque maintenant. Son application à un environnement professionnel constituerait un feedback très profitable pour les FUNDP.
 - Etant à l'état de développement, le framework ne propose pas encore un guide d'utilisation formalisé. B. Vanderose est donc la source d'information privilégiée pour le déploiement du framework.
 - S. Hanoteau s'engage à collecter ses remarques et suggestions d'amélioration au fur et à mesure du déploiement du framework et à transmettre ces informations.
 - Point de vue outillage, B. Vanderose finalise un outil (fichier XSD) de réalisation de modèle MoCQA. Cela devrait être terminé fin juillet.
 - Un technical report existe mais n'est pas à jour par rapport à la dernière version du framework. B. Vanderose l'enverra à S. Hanoteau, qui prendra garde aux différences éventuelles. Ce technical report donne plus d'informations sur le méta-modèle MoCQA.
- Projet de déploiement :
 - Le framework est conçu pour être flexible : un modèle doit donc être conçu pour chaque projet : à noter qu'on appelle ici « projet » l'ensemble des livrables gravitant autour d'une solution logicielle (application, analyse, documentation,...). Il va de soi qu'un modèle existant peut être repris et adapté à un nouveau projet.
 - La 1^{ère} phase (initialisation) est généralement rapide : il s'agit ici d'identifier les intervenants du projet (ceux qui ont un intérêt dans le retour qualité du projet : hiérarchie,





QUALITE_PRODUIITS
Compte rendu de réunion

Classification
Sensible

Version
1.0

Date
18/07/2011

Page
2 sur 2



- utilisateurs,...). Ensuite, on établit ce qui constitue la qualité pour chaque intervenant. Cette 1^{ère} phase peut être facilitée par l'utilisation de check-list contenant les besoins en qualité les plus courants. B. Vanderose enverra la check-list mise à jour à S. Hanoteau pour fin de cette semaine.
- La 2^{nde} phase (design) consiste à établir un modèle qualité répondant aux éléments qualité identifiés dans la 1^{ère} phase. Il s'agit ici d'identifier les métriques existantes et les métriques à mettre en place.
 - Théoriquement, une première version de l'implémentation de ces deux phases devrait être faite pour début août. Les phases suivantes seront alors envisagées.
 - Baser les premières métriques sur le respect des méthodes et standards pourrait être une solution intéressante.
 - Modes de communication :
 - La communication par mail reste privilégiée : B. Vanderose est disponible pour répondre.
 - **Des réunions d'avancement régulières seront organisées, en fonction de l'avancement des travaux.**
 - Vis-à-vis de la DGARNE, S. Hanoteau compte faire une présentation de ce qui va être concrètement réalisé en septembre à sa hiérarchie.
 - Divers :
 - **Des exemples de déploiement concrets existent sur des projets d'étudiant. Dans un premier temps, ils ne seront pas communiqués à S. Hanoteau afin de ne pas orienter son propre déploiement.**
 - Le framework est ouvert pour appliquer, à moyen terme, une norme ISO comme la 9126 ou les 250xx (SQuaRE).
 - A priori, le méta-modèle est suffisamment flexible pour mettre en place des métriques hors logiciels, comme des métriques sur la gestion des chefs de projet.
 - Cours du jour « Qualité des produits et des processus » :
 - **Ce cours peut être choisi par S. Hanoteau comme option pour son année de Master (2 options étant obligatoires). Il serait très intéressant puisque abordant en partie le framework MoCQA.**
 - **S'agissant d'un cours du jour uniquement, S. Hanoteau craint que le travail de groupe avec les étudiants du cours du jour soit complexe à organiser. N. Habra signale qu'une personnalisation est faisable pour un travail seul.**
 - **S. Hanoteau doit se renseigner auprès de son employeur et de son client pour voir si cela est faisable.**

4 **Prochaine réunion**

A planifier la première quinzaine d'août ou début septembre en fonction de l'avancement des travaux.





QUALITE_PRODITS
Compte rendu de réunion

Classification
Sensible

Version
1.0

Date
04/10/2011

Page
1 sur 2



Informations

Sujet : Réunion d'avancement
Date : 04/10/2011
Auteur : S. Hanoteau
Participants : N. Habra ; B. Vanderose ; S. Hanoteau
Excusés :
Distribution : Participants ; F. Dermine

1 Avant propos

Les parties du présent compte rendu concernant directement les FUNDP sont en couleur [bleue](#), puisque ne concernant pas la DGARNE. Le compte rendu publié sur l'espace projet public de la DGARNE ne contiendra pas ces parties.

2 Ordre du jour

- Etat d'avancement ;
- Présentation de la nouvelle version du méta-modèle MoCQA ;
- Réponse aux questions.

3 Contenu

- Etat d'avancement :
 - Les exigences qualité ont été récoltées auprès de 5 intervenants génériques : Responsable du Pôle Développement, Responsable du Pôle d'Appui au Développement, Responsable du Service Desk, Coordinateur de la Cellule d'Architecture, Coordinateur de la Cellule d'Assurance Qualité Informatique.
 - Un premier objectif « Fiabilité » (demande du Service Desk) a été sélectionné pour la première itération. Samuel a fait une première tentative de modèle MoCQA sur cet objectif, ce qui a généré quelques questions, à la fois du côté de Samuel et du côté de Benoît.
 - Afin de mieux s'approprier le méta-modèle, Samuel a redessiné celui-ci dans Enterprise Architect en ajoutant les cardinalités non présentes et le sens des relations.
 - Samuel a écrit une description de chaque classe du méta-modèle avec une explication pour chaque attribut et chaque relation. Cela a généré des questions de compréhension.
 - [En même temps, il s'agit d'une première proposition de traduction en français des concepts.](#)
- Méthode d'instanciation du méta-modèle : Benoît propose d'instancier le méta-modèle en 4 étapes. D'abord, définir le « Quality Issue », ensuite décrire l'existant avec la partie « project-related », ensuite définir les mesures avec la partie « measurement-related » et enfin compléter la partie « quality-related ». Samuel expérimentera cette méthode lors de sa prochaine instanciation.
- Samuel va adapter la description du méta-modèle en fonction des réponses données, avec éventuellement des exemples à chaque fois.
- Benoît propose une instance de l'objectif « Fiabilité » au nouveau format XOCQAM. Les spécifications de celui-ci devraient être disponibles prochainement (fin de semaine prochaine espérée).
- Concernant les types d'artefact (transformation horizontale/verticale endogène/exogène), Benoît va envoyer un article sur le sujet à Samuel.
- [Dans le cadre du mémoire, l'objectif ne doit pas se limiter à améliorer le méta-modèle MoCQA : la méthode et les impacts d'utilisation de celui-ci dans l'environnement professionnel seront étudiés.](#)





QUALITE_PRODUIITS
Compte rendu de réunion

Classification
Sensible

Version
1.0

Date
04/10/2011

Page
2 sur 2



4 **Prochaine réunion**

Date : mardi 11 octobre 2011

Heure : 17h-18h

Lieu : Salle académique

Ordre du jour : (dans la mesure du possible en une heure)

- Etat d'avancement ;
- Validation de la description adaptée du méta-modèle ;
- Explication de la 3^{ème} phase pour l'établissement du plan de mesure ;
- Avis sur nouvelle version de l'objectif « Fiabilité » ;
- Validation du formalisme des exigences qualité ;
- [Livrables attendus pour le mémoire.](#)





QUALITE_PRODUIITS
Compte rendu de réunion



Classification
Sensible

Version
1.0

Date
11/10/2011

Page
1 sur 2

Informations

Sujet : Réunion d'avancement
Date : 11/10/2011
Auteur : S. Hanoteau
Participants : B. Vanderose ; S. Hanoteau
Excusés : N. Habra
Distribution : Participants ; N. Habra ; F. Dermine

1 Ordre du jour

- Etat d'avancement ;
- Méta-modèle MoCQA ;
- Objectifs « Fiabilité » ;
- Phase d'opérationnalisation.

2 Contenu

- Etat d'avancement :
 - La description détaillée en français du méta-modèle a été modifiée mais Samuel souhaite faire encore quelques mises à jour avant validation.
 - L'objectif « Fiabilité » a été renommé en « Rapport incidents » afin d'en faire un sous-objectif de « Fiabilité ». Samuel a suivi la proposition de Benoît qui consistait à définir dans l'ordre l'objectif qualité, la partie produit, la partie mesure et enfin revenir sur la partie qualité. Il confirme qu'il s'agit d'une bonne méthode.
 - Un second sous-objectif de « Fiabilité » a été modélisé : « Disponibilité ». Deux autres sous-objectifs sont à venir.
- Méta-modèle MoCQA :
 - Samuel a supprimé de sa version du méta-modèle les classes « Scale » et « Unit » car ne comportant qu'un seul champ : il les a remplacées par des champs dans les classes associées. Benoît préfère laisser ces concepts sous forme de classes car il s'agit de concepts incontournables en matière de modélisation qualité.
 - Une question se pose par rapport à la façon de représenter les champs optionnels en UML ([0..1] ou autre ?) : Benoît et Samuel vérifieront.
 - Les concepts « Scale » et « Unit » ont également été ajoutés à la classe « Fonction » : Benoît confirme que c'est correct.
 - Samuel avait un problème de compréhension par rapport à la classe « Behaviour ». En fait, il s'avère que le concept « Incident » proposé par Benoît dans sa modélisation représente une collection d'incidents, ce qui solutionne la problématique de somme rencontrée par Samuel.
 - Par la même occasion, le besoin d'un lien supplémentaire permettant d'associer un « Behaviour » avec un autre apparaît, ceci afin de modéliser des concepts de type « Un incident est une partie d'une collection d'incidents ». Le méta-modèle sera mis à jour en fonction.
 - Benoît propose un exemple concret d'une dérivation.
 - Le statut « Proposition » devient maintenant « Expérimentation ».
 - Benoît va envoyer une version stable des spécifications XOCQAM à Samuel prochainement.
- Objectifs « Fiabilité » :
 - Il serait bon de mieux représenter le fait que l'objectif « Fiabilité » est composé de 4 sous-objectifs (voir outil EA ou utilisation de lignes pointillées).
 - Le champ « Scale » est obligatoire, y compris pour le Quality Indicator.
 - Pour l'objectif « Disponibilité », un attribut dérivé intermédiaire « Durée d'indisponibilité » doit être évalué juste avant la durée totale.





QUALITE_PRODUIITS
Compte rendu de réunion

Classification
Sensible

Version
1.0

Date
11/10/2011

Page
2 sur 2



- L'échelle d'une somme est toujours de type absolu.
- L'artefact « Application » (au sens exécutable) n'est finalement pas pertinent pour la « Fiabilité » : il faudrait le remplacer par un artefact « Produit » (voir concept de ligne de produit) qui concernerait l'application au sens large du terme (toutes versions confondues).
- Phase d'opérationnalisation :
 - La sensibilisation de la hiérarchie et des intervenants concernés par les évaluations est importante, le terme « évaluation » étant souvent mal perçu. On évalue des produits et pas des ressources humaines. Les indicateurs qualité doivent être idéalement des outils à destination de chacun. Samuel est tout à fait d'accord avec cette approche et communique déjà en ce sens : ces points seront néanmoins rappelés lors de la présentation à la hiérarchie D443.
 - La première partie de cette 3^{ème} phase est de définir comment mesurer (outils, ...), notamment en créant les documents dont l'ID est mentionné dans le modèle.
 - La seconde partie est de définir comment stocker les mesures. A cette fin, Samuel créera d'abord manuellement une base de données « Fiabilité » en consignant les différentes étapes à suivre à partir du modèle. Ensuite, un outil devrait être développé pour créer automatiquement les scripts de création de base de données.

3 Prochaine réunion

Date : mardi 25 octobre 2011

Heure : 17h-18h

Lieu : Salle académique

Ordre du jour : (dans la mesure du possible en une heure)

- Etat d'avancement ;
- Validation de la description adaptée du méta-modèle ;
- Avis sur nouvelle version de l'objectif « Fiabilité » ;
- Validation du formalisme des exigences qualité.





QUALITE_PRODUIITS

Compte rendu de réunion



Classification
Sensible

Version
1.0

Date
25/10/2011

Page
1 sur 2

Service puk

Informations

Sujet : Réunion d'avancement
Date : 25/10/2011
Auteur : S. Hanoteau
Participants : B. Vanderose ; S. Hanoteau
Excusés : N. Habra
Distribution : Participants ; N. Habra ; F. Dermine

1 Avant propos

Les parties du présent compte rendu concernant directement les FUNDP sont en couleur [bleue](#), puisque ne concernant pas la DGARNE. Le compte rendu publié sur l'espace projet public de la DGARNE ne contiendra pas ces parties.

2 Ordre du jour

- Etat d'avancement ;
- Modèle complet « Fiabilité » ;
- Phase d'opérationnalisation ;
- Divers.

3 Contenu

- Etat d'avancement :
 - Samuel a terminé le modèle « Fiabilité » avec ses quatre sous-objectifs.
 - Samuel a également formalisé la méthode à suivre pour la 2^{ème} phase (modélisation).
 - Il a également formalisé la méthode de la 1^{ère} phase mais celle-ci sera revue au début de la seconde itération. En attendant, Samuel va déjà envoyer ses pistes d'amélioration à Benoît pour info.
- Modèle complet « Fiabilité » :
 - Ce modèle est maintenant composé de quatre sous-objectifs : « Survenance des incidents », « Disponibilité », « Impact des incidents » et « Fréquence de déploiement ».
 - L'instanciation d'un comportement « Evaluation » n'a pas sa place dans le modèle MoCQA : les données externes, telles que la date d'évaluation, doivent plutôt être introduites via les fonctions/méthodes.
 - Concernant la gestion du temps pour les incidents, Samuel propose de modéliser sous forme d'un comportement « Incidents récents », qui représente la collection des incidents datant de moins d'un an. Ainsi, on n'a plus à se soucier de cette gestion par la suite. La qualité première d'un modèle étant sa compréhension, cette modélisation est considérée comme adéquate.
 - Samuel va faire les quelques adaptations nécessaires et envoyer le modèle pour révision.
- Phase d'opérationnalisation :
 - Samuel va pouvoir opérationnaliser le modèle Fiabilité. Il va définir une structure de données manuellement et éventuellement utiliser le formalisme XOCQAM si envoi des spécifications par Benoît. Sinon, une description semi-formelle suffira pour ne pas bloquer l'avancement.
 - Le sujet des règles d'interprétation intéresse particulièrement Benoît : un lien (via une modélisation séparée) pourrait être identifié entre les attributs évalués et les entités pour déterminer les causes d'éventuels mauvais résultats. De son côté, Samuel compte assortir les règles d'interprétation de recommandations aidant les ressources à définir les causes.





QUALITE_PRODUIITS
Compte rendu de réunion

Classification
Sensible

Version
1.0

Date
25/10/2011

Page
2 sur 2



- Samuel va probablement définir le cycle de vie des produits de son Service Informatique au format MoCQA (en utilisant les concepts orientés Produit uniquement), ceci étant intéressant car les indicateurs mis en place dans un premier temps sont génériques.
- Divers :
 - La question « philosophique » de l'échelle d'une date demeure.
 - Concernant la possibilité de développer l'outil de création des scripts de base de données en C#.Net, Benoît ne préfère pas, étant donné qu'un développement en Java permettrait une intégration plus simple. Benoît va réfléchir à la possibilité de le faire lui-même, ou de partager la tâche avec Samuel, celui-ci ne connaissant pas le développement d'interfaces graphiques en Java (et le développement étant hors sujet du mémoire).
 - Le méta-modèle en version française étant stabilisé, Samuel souhaite une validation de celui-ci : il enverra une copie pour révision.
 - Samuel va profiter des prochains jours pour également avancer dans la rédaction du mémoire : il va en même temps proposer une liste de livrables attendus découlant des objectifs. Ces livrables seront soumis à la validation de Naji Habra.

4 Prochaine réunion

Date : jeudi 10 novembre 2011

Heure : 17h-18h

Lieu : Salle académique

Ordre du jour : (dans la mesure du possible en une heure)

- Etat d'avancement ;
- Remarques éventuelles sur le modèle « Fiabilité » ;
- Validation de la méthode « Modélisation » ;
- Validation du méta-modèle en version française ;
- Discussion sur les livrables attendus du mémoire.





QUALITE_PRODUIITS
Compte rendu de réunion

Classification
Sensible

Version
1.0

Date
10/11/2011

Page
1 sur 2



Informations

Sujet : Réunion d'avancement
Date : 10/11/2011
Auteur : S. Hanoteau
Participants : N. Habra ; B. Vanderose ; S. Hanoteau
Excusés :
Distribution : Participants ; F. Dermine

1 Avant propos

Les parties du présent compte rendu concernant directement les FUNDP sont en couleur [bleue](#), puisque ne concernant pas la DGARNE. Le compte rendu publié sur l'espace projet public de la DGARNE ne contiendra pas ces parties.

2 Ordre du jour

- Livrables attendus du mémoire ;
- Organisation.

3 Contenu

- Livrables attendus :
 - Samuel propose une liste structurée séparant le contenu du mémoire et les livrables du déploiement.
 - A l'avenir, il serait mieux de travailler avec la table des matières.
 - Un état de l'art des autres modèles qualité serait un plus.
 - A l'avenir, on devrait plutôt parler de l'approche MoCQA qui intègre la méthodologie, le méta-modèle et les outils.
 - Finalement, la grosse valeur ajoutée du mémoire sera de « vulgariser » les concepts universitaires en formalisant un guide d'utilisation de l'approche MoCQA. Celui-ci devrait permettre à toute entreprise de partir de rien et de s'approprier l'approche MoCQA sans devoir passer par les étapes de compréhension par lesquelles Samuel est passé. Un nouveau standard interne de gestion des indicateurs qualité devra être réalisé pour la D443 : celui-ci servira de base pour le guide complet et inversement.
 - Les futurs sujets de réflexion seront les bienvenus pour d'éventuels futurs mémoires ou recherches réalisées au sein des FUNDP et/ou de la D443.
 - La validation elle-même de l'approche est plutôt de la responsabilité de la thèse de Benoît. Néanmoins, une contribution à la validation sera faite via une section du mémoire.
 - Il est important d'avoir une vision critique à la fois de l'approche en général et sur sa propre interprétation.
 - Le retour de la hiérarchie D443 sur l'approche sera également une valeur ajoutée du mémoire.
 - Le coût du déploiement est également une information intéressante.
- Organisation :
 - Les révisions du mémoire pourront se faire régulièrement en fonction de l'avancement des travaux. Naji se concentrera plus au niveau de la table des matières et Benoît sur le contenu.
 - L'échéance de remise du mémoire aux FUNDP étant le 31/05/2012, la fin des travaux de déploiement eux-mêmes est fixée au 30/04/2012 pour laisser à Samuel le temps de finaliser d'éventuels derniers livrables et surtout le mémoire lui-même. La fin des travaux liés au mémoire ne marque bien sûr pas la fin des travaux à la D443.



DIRECTION GÉNÉRALE OPÉRATIONNELLE
DE L'AGRICULTURE, DES RESSOURCES NATURELLES ET DE L'ENVIRONNEMENT



QUALITE_PRODUIITS
Compte rendu de réunion

Classification
Sensible

Version
1.0

Date
10/11/2011

Page
2 sur 2



4 **Prochaine réunion**

Date : jeudi 8 décembre 2011

Heure : 17h-18h

Lieu : Salle académique

Ordre du jour : (dans la mesure du possible en une heure)

- Etat d'avancement.





QUALITE_PRODUITS
Compte rendu de réunion



Classification
Sensible

Version
1.0

Date
08/12/2011

Page
1 sur 2

Service puk

Informations

Sujet : Réunion d'avancement
Date : 08/12/2011
Auteur : S. Hanoteau
Participants : N. Habra ; B. Vanderose ; S. Hanoteau
Excusés :
Distribution : Participants ; F. Dermine

1 Avant propos

Les parties du présent compte rendu concernant directement les FUNDP sont en couleur [bleue](#), puisque ne concernant pas la DGARNE. Le compte rendu publié sur l'espace projet public de la DGARNE ne contiendra pas ces parties.

2 Ordre du jour

- Etat d'avancement.

3 Contenu

- Afin de produire une première idée de livrable, Samuel a reçu un export des attributs de base modélisés pour la « Fiabilité ». Ces mesures ont été intégrées dans un simple fichier Excel et complétées avec le calcul des attributs dérivés et des indicateurs qualité. Cela a donné naissance à un premier tableau de bord, accompagné de plusieurs constatations.
- Bien que Samuel aie déjà quelques idées de réponses aux constatations, il ne veut pas avoir la prétention de les soumettre : une réunion sera provoquée avec le responsable du Service Desk (à l'origine de la demande), le responsable du Pôle Développement (gardien des applications) et le chef du Service Informatique (preneur de décisions). Chacun contribuera ainsi à apporter des solutions aux constatations émises.
- Parce que l'approche MoCQA n'est pas encore connue, et donc pour éviter une mauvaise interprétation des résultats, il est préférable de ne pas transmettre ceux-ci à l'avance aux intervenants.
- Un feedback de cette réunion sera donné aux FUNDP lors de la prochaine réunion d'avancement. L'un des objectifs est également de voir à quel niveau le modèle MoCQA a une valeur ajoutée dans l'aspect communication.
- Samuel va maintenant formaliser le plan de mesure qui a permis d'atteindre le résultat.
- Une base de données sera créée pour accueillir les données et donc remplacer le fichier Excel. A ce sujet, la question se pose de savoir ce qu'on conserve en DB : soit uniquement les attributs de base qui permettront de recalculer les attributs dérivés et les indicateurs qualité à chaque interrogation, soit toutes les métriques. Cette dernière solution semble plus adéquate à Benoît et Samuel pour des raisons de performance et de simplicité pour les statistiques. Benoît va néanmoins présenter le contexte à ses collègues spécialistes en base de données pour obtenir leur avis sur la question.
- Samuel va également définir les règles d'interprétation des indicateurs qualité.
- Enfin, le cycle de vie logiciel D443 sera modélisé au format MoCQA sous forme d'artefacts, de comportements et de dérivations. Cette représentation devrait permettre également de mettre au jour certains manques éventuels.
- Dans les prochaines 2 semaines, Benoît devrait finaliser l'outil de création de modèles MoCQA : celui-ci sera fourni à Samuel.
- **Le plan du mémoire soumis récemment par Samuel est validé.**





QUALITE_PRODUIITS
Compte rendu de réunion

Classification
Sensible

Version
1.0

Date
08/12/2011

Page
2 sur 2



4 Prochaine réunion

Date : mardi 20 décembre 2011

Heure : 17h-18h

Lieu : FUNDP - Salle académique

Ordre du jour : (dans la mesure du possible en une heure)

- Etat d'avancement,
- Feedback sur la première évaluation.





QUALITE_PRODUIITS
Compte rendu de réunion

Classification
Sensible

Version
1.0

Date
20/12/2011

Page
1 sur 2



Informations

Sujet : Réunion d'avancement
Date : 20/12/2011
Auteur : S. Hanoteau
Participants : B. Vanderose ; S. Hanoteau
Excusés : N. Habra
Distribution : Participants ; Excusés ; F. Dermine

1 Avant propos

Les parties du présent compte rendu concernant directement les FUNDP sont en couleur [bleue](#), puisque ne concernant pas la DGARNE. Le compte rendu publié sur l'espace projet public de la DGARNE ne contiendra pas ces parties.

2 Ordre du jour

- Etat d'avancement.

3 Contenu

- **Approche MoCQA**
 - Samuel présente une proposition de structure pour l'enregistrement des informations :
 - Un *registre des objectifs qualité* reprenant chaque objectif (le contenu doit être fortement amélioré car la première version a été faite alors que Samuel ne maîtrisait pas du tout l'approche MoCQA).
 - Un *registre des indicateurs qualité* (reprenant la description de chaque indicateur ainsi que son modèle d'évaluation).
 - Un *registre des évaluations* (reprenant chaque évaluation et chaque exploitation).
 - La *structure des produits* D443, reprenant chaque artefact du cycle de vie logiciel de la D443 avec la description des méthodes et fonctions associées. Pour ce point, plutôt que modéliser l'entièreté du cycle de vie en une fois comme prévu initialement, Samuel prend plutôt une approche itérative, le but étant d'ajouter les artefacts manquants dès qu'ils sont utilisés.
- **Indicateurs « Fiabilité »**
 - L'exploitation de la première évaluation des indicateurs Fiabilité a donné lieu à une réunion où participaient le responsable du help-desk (car demandeur de ces indicateurs), le responsable du développement (car responsable de la liste officielle des applications), le chef de service (car responsable des décisions et de l'arbitrage inter-pôles) et le coordinateur de la cellule d'assurance qualité (=Samuel, responsable des métriques).
 - Cette première réunion a donné naissance à un tableau de suivi des constatations. Ce tableau est composé des colonnes suivantes :
 - Id (identifiant unique)
 - Evaluation (date de l'évaluation ayant généré la constatation)
 - Création (date de création)
 - Modification (date de dernière modification)
 - Indicateur (indicateur concerné)
 - Constatation (description)
 - Action (description de l'action décidée)
 - Phase MoCQA (phase de la méthodologie MoCQA concernée par l'action)
 - Attribution (ressource responsable de l'action)
 - Sévérité (« bloquant », « critique », « majeure », « mineure »)





QUALITE_PRODUIITS
Compte rendu de réunion



Classification
Sensible

Version
1.0

Date
20/12/2011

Page
2 sur 2

Service pvt

- Statut (« à faire », « en cours », « en attente », « à valider », « terminé »)
- Remarques (remarques éventuelles)
- Le tableau contient 14 constatations, incluant 1 critique, 7 majeures et 6 mineures : le fait que la première itération produise ces constatations est déjà une victoire pour Samuel et Benoît. Même si la majorité de ces problèmes étaient connus, ceux-ci sont désormais identifiés et formalisés avec un suivi à faire.
- A noter que pour le suivi des actions, un workflow pourrait être prochainement défini dans l'outil de tracking Jira mais Samuel fait le choix de ne pas s'embarrasser de cela à court terme.
- Samuel signale que le coût ne semble pas énorme, la majorité du temps passé étant pour comprendre et mettre en place les concepts : la production elle-même ne constitue qu'une petite partie. Samuel essaiera de quantifier cela pour la prochaine réunion.
- Samuel doit encore achever de formaliser les méthodes et fonctions, ainsi que définir des règles d'interprétation.
- **Indicateurs « Cycle de vie »**
 - Samuel va pouvoir commencer une nouvelle itération sur le cycle de vie.
- **Mémoire**
 - Samuel est en train de préparer une première version du document : il la soumettra dès que possible.

4 **Prochaine réunion**

Date : mardi 24 janvier 2012 (si peu d'avancement, possibilité de report au 31 janvier 2012)

Heure : 17h-18h

Lieu : FUNDP - Salle académique

Ordre du jour : (dans la mesure du possible en une heure)

- Etat d'avancement,
- Lancement de la seconde itération.





QUALITE_PRODUITS
Compte rendu de réunion

Classification
Sensible

Version
1.0

Date
13/02/2012

Page
1 sur 2



Informations

Sujet : Réunion d'avancement
Date : 13/02/2012
Auteur : S. Hanoteau
Participants : B. Vanderose ; S. Hanoteau
Excusés : N. Habra
Distribution : Participants ; Excusés ; F. Dermine

1 Avant propos

Les parties du présent compte rendu concernant directement les FUNDP sont en couleur [bleue](#), puisque ne concernant pas la DGRNE. Le compte rendu publié sur l'espace projet public de la DGRNE ne contiendra pas ces parties.

2 Ordre du jour

- Etat d'avancement.

3 Contenu

- **Indicateurs « Fiabilité »**
 - Samuel a rédigé un plan de mesure avec, pour chaque modèle d'évaluation, les détails de l'évaluation sous forme de pseudo-code et des règles d'interprétation.
 - Les détails sous forme de pseudo-code sont une bonne idée, puisque permettant de bien comprendre l'évaluation.
 - Les règles d'interprétation sous forme de codes couleur dont les tranches de valeurs sont déterminées en fonction d'une description plus parlante (exemple : « maximum un incident par trimestre » donne une tranche « $0,00 \leq i < 0,02$ ») sont bienvenues.
 - Samuel propose plusieurs interprétations possibles d'une valeur basse et d'une valeur élevée. Ceci est correct mais il manque un mapping entre les codes couleur et les tableaux d'interprétations. Idéalement, si on a 4 tranches, on doit avoir 4 tableaux. La difficulté est de déterminer les tableaux intermédiaires : peut-être affiner en fonction de la criticité des produits. Samuel va refaire une proposition.
 - Concernant le sous-indicateur « Disponibilité », l'une des actions émergeant de l'exploitation de celui-ci est de créer un autre indicateur reprenant la disponibilité pendant le temps de travail uniquement (contrairement à une disponibilité 24h/24). Benoît signale qu'il est mieux de créer un second indicateur lié au même modèle d'évaluation : un doute subsiste car le méta-modèle ne le permet peut-être pas. Benoît va vérifier et donner un feedback au plus tôt à Samuel à ce sujet.
 - S'il est confirmé qu'on peut créer plusieurs indicateurs pour un même modèle d'évaluation, un problème potentiel existe concernant la limite à ne pas dépasser puisque dans le pire des cas, on pourrait alors avoir un seul modèle d'évaluation accédant à tous les attributs possibles et générant tous les indicateurs possibles. La limite doit donc consister à grouper uniquement des indicateurs ayant la même sémantique (ici : disponibilité).
- **Indicateurs « Cycle de vie »**
 - Samuel a réalisé le modèle de l'indicateur « Cycle de vie » ainsi qu'un sous-indicateur « Validation de l'analyse fonctionnelle » : les autres sous-indicateurs seront quasi du même style.
 - Dans l'ensemble, il s'agit d'une bonne modélisation. Seule remarque : la présence de la valeur « inexistant » dans l'échelle de validation d'une analyse enlève le caractère ordinal de l'échelle. La bonne pratique ici est de ne pas spécifier cette valeur : dans le modèle





QUALITE_PRODUTS
Compte rendu de réunion

Classification
Sensible

Version
1.0

Date
13/02/2012

Page
2 sur 2



d'évaluation, on traitera le cas de l'inexistence d'une validation mais pas au niveau des attributs.

- **Mémoire**

- Samuel a proposé une nouvelle version reprenant une structuration mise à jour avec, pour chaque section, une description succincte du futur contenu, ainsi que le chapitre 3 complet concernant l'environnement.
- Benoît a relu la structure et est d'accord avec celle-ci : Naji Habra doit encore la valider.
- Benoît n'a pas encore eu le temps de lire le chapitre sur l'environnement mais le fera et donnera un feedback à Samuel.
- Samuel se pose la question de la pertinence de la section « Prérequis à la lecture » puisque théoriquement, un mémoire est sensé être compréhensible par ses pairs. La section ne dérange pas et peut rappeler que le lecteur devra avoir des notions d'UML et peut-être de BPMN, tout en rappelant les références de l'OMG concernant ces notations.
- Samuel se demande également s'il est utile de préciser les sites intranet SPW dans les références, alors que ceux-ci ne sont pas consultables par le public. Ils peuvent effectivement être enlevés de la liste.
- Dans le feedback de déploiement, il serait intéressant de discuter des outils permettant de gérer le plan de mesure ou les actions liées à l'exploitation des indicateurs (typiquement, l'outil Jira est adéquat au SPW pour gérer les actions).
- Un nouveau chapitre pourrait être créé, plus « philosophique » : le fait que les indicateurs permettent d'atténuer la notion de « contrôle » qualité, jamais bien perçu par les ressources humaines. L'objectivité des indicateurs est mieux perçue que la subjectivité du ressenti par rapport à un produit, même si souvent, le ressenti est confirmé par les indicateurs.

4 **Prochaine réunion**

Date : à définir par mail

Heure : à définir par mail

Lieu : à définir par mail

Ordre du jour : (dans la mesure du possible en une heure)

- Mise à jour des indicateurs « Fiabilité »
- Mise à jour des indicateurs « Cycle de vie »
- **Avancement du mémoire**





QUALITE_PRODUIITS
Compte rendu de réunion

Classification
Sensible

Version
1.0

Date
28/02/2012

Page
1 sur 2



Informations

Sujet : Réunion d'avancement
Date : 28/02/2012
Auteur : S. Hanoteau
Participants : B. Vanderose ; S. Hanoteau
Excusés : N. Habra
Distribution : Participants ; Excusés ; F. Dermine

1 Avant propos

Les parties du présent compte rendu concernant directement les FUNDP sont en couleur [bleue](#), puisque ne concernant pas la DGARNE. Le compte rendu publié sur l'espace projet public de la DGARNE ne contiendra pas ces parties.

2 Ordre du jour

- Etat d'avancement.

3 Contenu

- **Indicateurs « Fiabilité »**
 - Samuel a fait un test de deux indicateurs pour un même modèle d'évaluation dans le modèle « Fréquence de déploiement ». Benoît valide que c'est bien la bonne modélisation.
- **Indicateurs « Cycle de vie »**
 - Samuel a réalisé le modèle de l'indicateur « Cycle de vie » ainsi qu'un sous-indicateur « Validation de l'analyse fonctionnelle » : les autres sous-indicateurs seront quasi du même style.
 - Dans l'ensemble, il s'agit d'une bonne modélisation. Seule remarque : la présence de la valeur « inexistant » dans l'échelle de validation d'une analyse enlève le caractère ordinal de l'échelle. La bonne pratique ici est de ne pas spécifier cette valeur : dans le modèle d'évaluation, on traitera le cas de l'inexistence d'une validation mais pas au niveau des attributs.
- **Méthodologie**
 - Samuel a révisé la nouvelle version du méta-modèle envoyé et relève certains points :
 - Bien qu'au début d'une modélisation, un *MeasurableEntityType* n'ait pas forcément au minimum 1 *Attribute*, dans la réalité c'est le cas.
 - Un *AssessmentModel* n'a pas au minimum 1 *Attribute* en entrée : il peut en avoir 0 s'il prend un *QualityIssue* en entrée.
 - Un *Attribute* n'est pas systématiquement une entrée d'un *AssessmentModel* puisqu'il peut être une entrée d'une fonction. Mais c'est effectivement soit l'un, soit l'autre (contrainte).
 - La cardinalité de sous-division d'un *BehaviourType* est 0..1 et non 1.
 - Une question se pose sur la pertinence de montrer le tableau des indicateurs aux utilisateurs.
 - Cela dépend essentiellement de l'environnement et de la politique de l'organisation mais il est clair que si les chiffres sont publiés, une sensibilisation de l'interprétation de ceux-ci est essentielle.
 - Certains indicateurs peuvent être fournis s'il s'agit d'indicateurs définis pour l'acceptation (où l'utilisateur est le Stakeholder).
 - Samuel va prochainement faire une présentation aux responsables de Pôles et aux coordinateurs du PAD.





QUALITE_PRODUIITS
Compte rendu de réunion

Classification
Sensible

Version
1.0

Date
28/02/2012

Page
2 sur 2



- Les slides ont fait l'objet de quelques remarques par Naji Habra : ils ont été adaptés en ce sens.
 - Pas de remarque de Benoît.
- Samuel va désormais attaquer les procédures BPMN de définition de la méthodologie. Benoît est très intéressé par recevoir les premiers jets au fur et à mesure qu'ils arrivent.
- **Mémoire**
 - Samuel a soumis une nouvelle version du mémoire.
 - Naji Habra est ok pour la structure et a peu de remarques sur le contenu.
 - Benoît n'a pas encore eu le temps de réviser le document.
 - Défense du mémoire.
 - Samuel s'interroge si, lors de la défense, l'approche MoCQA doit être expliquée : oui mais rapidement car cela ne constitue pas le sujet de mémoire en lui-même.
 - Samuel s'inquiète d'une éventuelle redondance entre le « guide de déploiement MoCQA » qu'il réalise dans le cadre de son mémoire et les travaux actuels de Benoît : pas de souci à ce niveau-là car les travaux de Benoît sont essentiellement théoriques alors que le guide met l'accent sur la pratique.
 - Samuel compte soumettre prochainement une nouvelle version avec les chapitres « Modèles Qualité » et « Approche MoCQA ».

4 Prochaine réunion

Date : mardi 13 mars 2012

Heure : 17h00

Lieu : FUNDP - Salle académique

Ordre du jour :

- Etat d'avancement.





QUALITE_PRODUIITS

Compte rendu de réunion

Classification
Sensible

Version
1.0

Date
03/04/2012

Page
1 sur 2



Informations

Sujet : Réunion d'avancement
Date : 03/04/2012
Auteur : S. Hanoteau
Participants : B. Vanderose ; S. Hanoteau
Excusés : N. Habra
Distribution : Participants ; Excusés

1 Avant propos

Les parties du présent compte rendu concernant directement les FUNDP sont en couleur [bleue](#), puisque ne concernant pas la DGARNE. Le compte rendu publié sur l'espace projet public de la DGARNE ne contiendra pas ces parties.

2 Ordre du jour

- Etat d'avancement.

3 Contenu

- **Mémoire**
 - Samuel a soumis une nouvelle version du mémoire incluant le chapitre 1 sur les modèles qualité.
 - Aucune révision n'a encore pu être faite.
 - Samuel a terminé les chapitres 2 (MoCQA) et 4 (déploiement) mais ne les a pas publiés car certains schémas manquants : il va les intégrer et soumettre une nouvelle version prochainement.
 - A cause du début non prévu des examens au 31/05, Samuel planifie d'avoir sa première version complète du mémoire pour fin avril afin de le soumettre pour révision à sa liste de réviseurs. Il pourrait ainsi remettre le mémoire officiellement pour mi-mai.
 - Planning à venir :
 - 22/04 au plus tard : Samuel soumet une version complète (sauf annexe 1) à Naji Habra et Benoît Vanderose pour avis.
 - 27/04 au plus tard : Naji Habra et Benoît Vanderose remettent leur remarques à Samuel.
 - 30/04 au plus tard : Samuel soumet une nouvelle version tenant compte des remarques et accompagnée de l'annexe 1 (guide de déploiement) à Naji Habra, Benoît Vanderose et sa liste de réviseurs (composée de collègues et/ou amis personnels).
 - 11/05 au plus tard : les réviseurs remettent leurs dernières remarques à Samuel.
 - 16/05 au plus tard : Samuel soumet une version officielle du mémoire aux FUNDP tenant compte des remarques.
 - Défense du mémoire.
 - Samuel aimerait connaître au plus vite la date de défense de son mémoire : à cette fin, un jury doit être composé.





QUALITE_PRODUIITS
Compte rendu de réunion

Classification
Sensible

Version
1.0

Date
03/04/2012

Page
2 sur 2



4 Prochaine réunion

Date : vendredi 27 avril 2012

Heure : 17h00

Lieu : FUNDP - Salle académique

Ordre du jour :

- Discussion sur les remarques faites sur la version complète (sauf annexe 1) du mémoire.

