



Mesurer les Exigences

Bonnes Pratiques

Olivier PINETTE – Denise CATTAN

STATUT : V1.6 – 2010/03/16 - VALIDE



1 Introduction

Ce document fait parti d'un ensemble de documents sur les « Bonnes pratiques » de développement de logiciels et systèmes aidées par les indicateurs.

Pourquoi la mesure des exigences ?

Parce qu'il est reconnu que la majorité des défauts (entre 40 et 60%) a comme origine les exigences.

Les exigences représentent donc un levier d'amélioration important.

Commençons par quelques rappels...

Dans l'ingénierie système/logiciel, une **exigence** est généralement la description de ce que le système/logiciel livré doit être capable de faire (exigence fonctionnelle). A cela, il faut aussi ajouter les « exigences non fonctionnelles », portant sur la manière dont le système/logiciel doit exécuter ses fonctions ou encore les « exigences de performance », les « exigences de qualité de service », etc.

A ces exigences « produit » s'ajoutent également les exigences « métier » (qui décrivent le « quoi ») et les exigences processus (qui décrivent le « comment », c'est à dire les contraintes).

L'ensemble des exigences définit les caractéristiques exigées/désirées pour le système/logiciel.

La gestion des exigences consiste à gérer les exigences hiérarchisées d'un projet, à détecter les incohérences entre elles et à assurer leur traçabilité.

La problématique...

MOA = Maître d'ouvrage (ou maîtrise d'ouvrage). C'est l'initiateur du projet. Il est donc le représentant des utilisateurs finaux à qui l'ouvrage est destiné.

MOE = Maître d'œuvre (ou maîtrise d'œuvre). C'est l'entité retenue pour réaliser l'ouvrage

Le MOA est client du MOE à qui il passe commande d'un produit nécessaire à son activité.

Généralement le MOA recueille l'expression des besoins de l'utilisateur et rédige des spécifications fonctionnelles et un cahier des charges à destination du MOE.

Le MOE fournit ce produit ; soit il le réalise lui-même, soit il passe commande à un ou plusieurs fournisseurs qui élaborent le produit sous sa direction.

Il peut également parfois y avoir des intermédiaires entre MOE et MOA.

Nous connaissons tous le jeu du « téléphone arabe » où le but est de transmettre un message oralement du premier au dernier joueur de la chaîne (sans se faire entendre des autres) en conservant l'intégralité du message. Et nous savons tous combien c'est difficile !

Je ne résiste pas à la tentation de vous fournir une illustration humoristique (variante de la fameuse balançoire sous un arbre que nous avons tous aperçue ici et là).

Illustration de ce qui arrive lorsque cette chaîne ne fonctionne pas.

Autre exemple, répondez à la question : « Qu'est-ce qui se déplace ventre à terre, laisse souvent des traces de son passage, surveille l'horizon avec des antennes télescopiques et reste bien protégé sous sa carapace ? »... Réponse page 6.

Exprimé comme cela, nous pourrions être fatalistes et dire « mission impossible »... d'autant que, dans le cas de développement système/logiciel, ce n'est pas une phrase, mais un ensemble de phrases (exigences) qu'il va falloir transmettre...

Heureusement les règles « du jeu » sont différentes ! Si dans le jeu du « téléphone arabe » il y a la règle interdisant de répéter la phrase à son voisin si ce dernier n'a pas bien compris, dans le cas du développement système/logiciel, il y a la (ou plutôt les) règle(s) inverse(s). En effet, la gestion des exigences inclut une règle permettant de détecter les incohérences et une autre, permettant de gérer leur traçabilité.



2 Bonnes pratiques

Les activités de test font indubitablement partie du processus de développement. Il semble tout naturel de tester le système/logiciel au regard des exigences. Cependant, si les développements sont basés sur des exigences imprécises ou incorrectes, même si les développements sont irréprochables, ils ne seront pas satisfaisants (du point de vue de l'utilisateur). Nous savons aujourd'hui que c'est la première cause d'échec des projets. Il faut donc tester non seulement les développements, mais aussi les exigences elles-mêmes.

Bonne pratique n°1 : Il faut valider les exigences

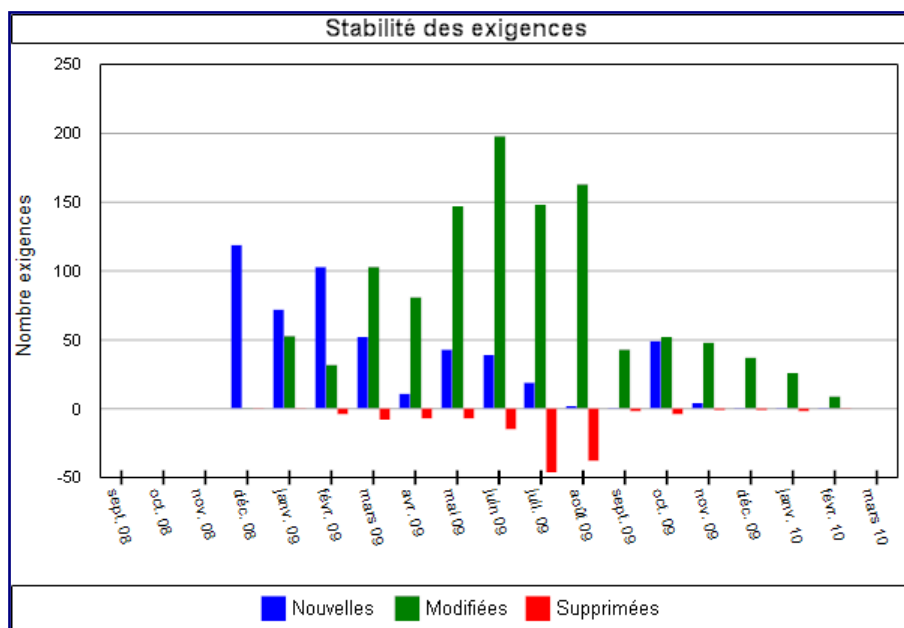
Corolaire : il faut vérifier le statut des exigences

Cependant, même si l'objectif est de faire en sorte que l'équipe de développement ait (dès le premier jour) des spécifications détaillées complètes et conformes au besoin client, il est difficile d'y parvenir. Malheureusement, les exigences sont par nature capricieuses, imprévisibles et même parfois invisibles... bref, elles changent dans le temps.

Bonne pratiques n°2 : Il faut vérifier la stabilité/volatilité des exigences

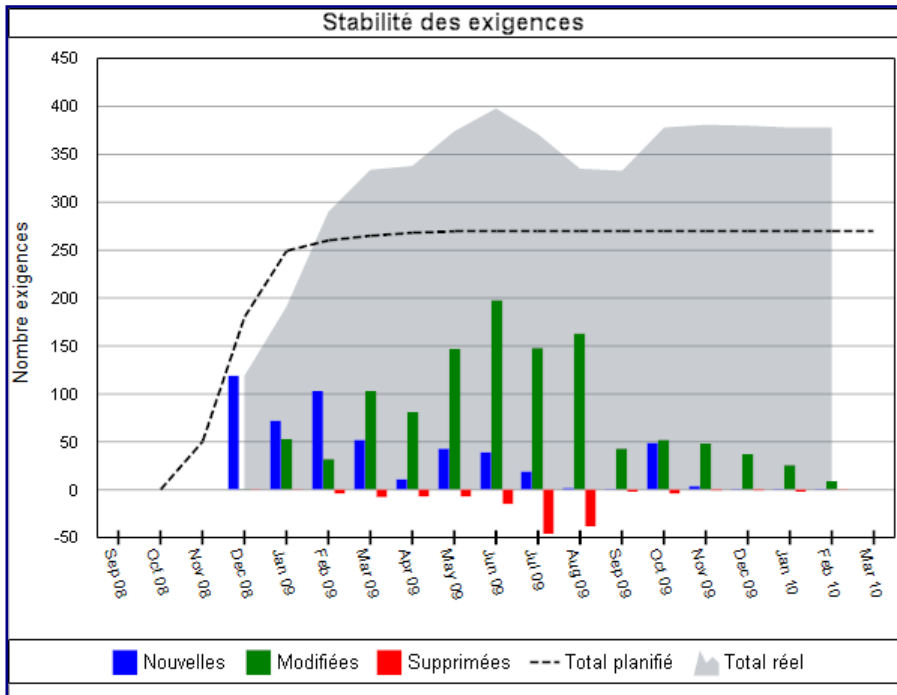
3 Les indicateurs de mesure des exigences

Indicateurs sur la Stabilité / volatilité des exigences



Cet indicateur permet d'examiner la stabilité (ou volatilité) des exigences, en dénombrant le nombre d'exigences ajoutées, modifiées et supprimées.

Cet indicateur commence à nous apprendre comment évoluent nos exigences dans le temps ... mais essayons de pousser l'analyse plus loin...

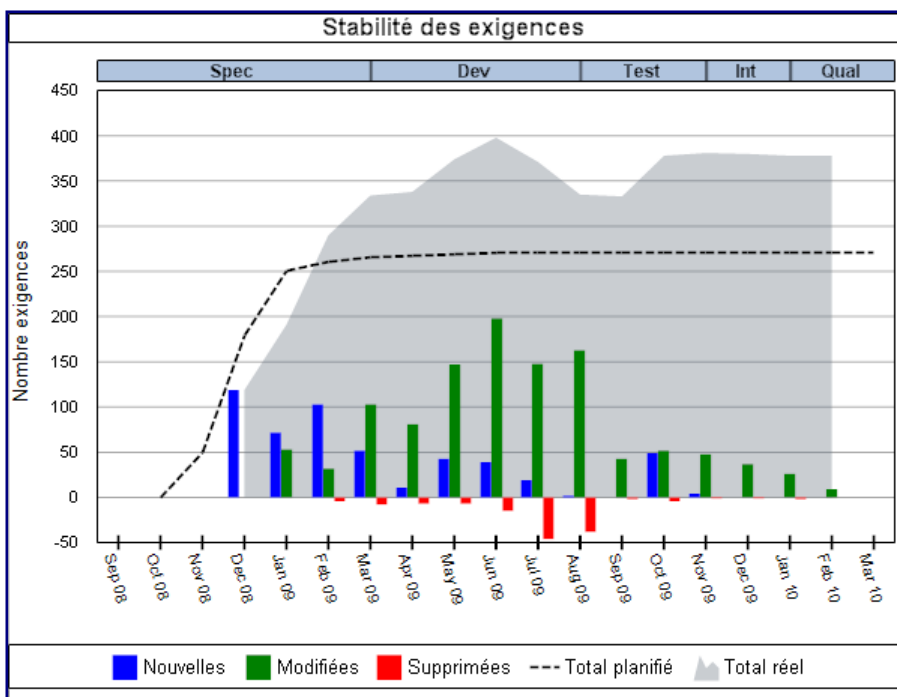


Une vision globale du statut des exigences peut être obtenue sur le même graphique en ajoutant 2 courbes :

- la somme totale des exigences (Total réel) de manière à avoir une idée de la croissance.
- la prévision de variation des exigences telle qu'elle a été estimée (Total planifié).

Dans notre exemple, la seule analyse que nous pouvons porter est que le

nombre d'exigences actuelles est bien supérieur au nombre d'exigences estimé. Une action doit donc être menée pour vérifier l'impact sur l'effort de développement et les éventuels surcoûts et/ou retards engendrés. Tout changement ou addition d'exigences, après les revues d'exigences, conduit inévitablement à un ajustement de l'effort, de la planification et/ou de budget.



Sur notre graphe, ajoutons maintenant les phases du projet.

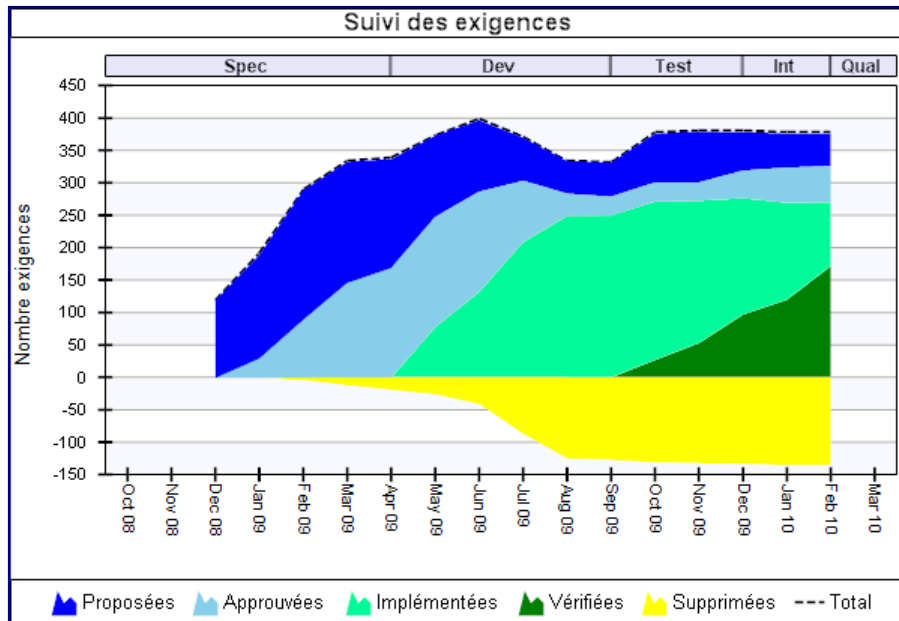
En effet, si la volatilité des exigences n'est pas un réel problème en phase de spécification, il en va tout autrement lorsque les développements ont débuté.

Dans notre exemple, de nombreux changements interviennent après la phase de spécification... il est donc certain qu'il y aura du

« rework » ! Une nouvelle raison pour réajuster l'effort.

Il serait également intéressant d'analyser les zones affectées par ces changements (composant, interface, performance, fonction clef, ...) et/ou les causes (exigences manquantes, imprécises, incomplète, ...) de manière à prévenir un tel phénomène à l'avenir.

Indicateurs de suivi des exigences



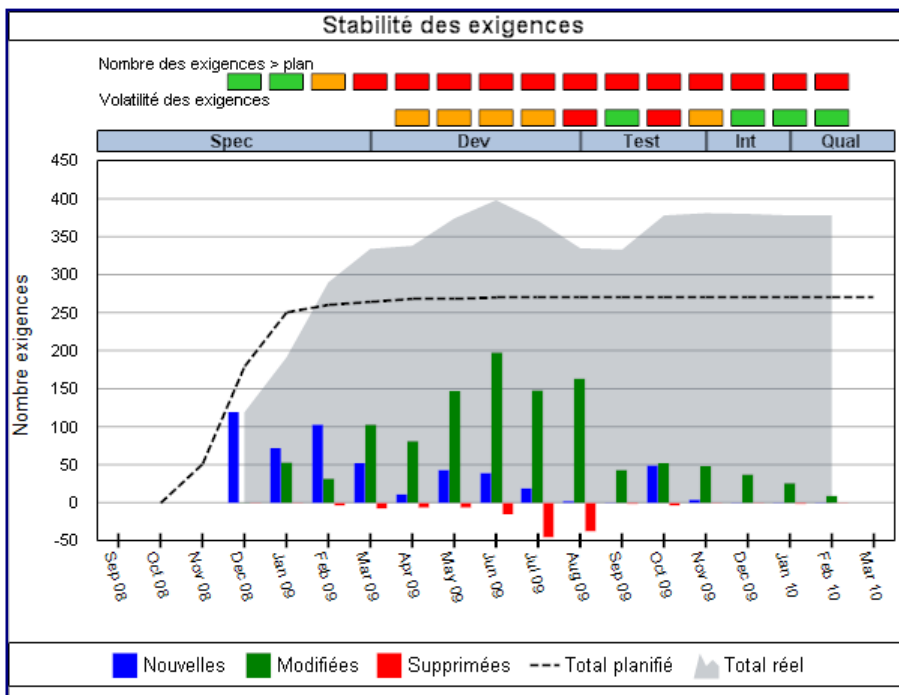
Cet indicateur nous montre les exigences selon leur statut (Proposée, Approuvée, Implémentée, Vérifiée, Supprimée)

Là aussi, nous avons placé les phases du projet.

Ce graphe confirme que les exigences n'étaient pas approuvées au démarrage des développements. Le risque était donc élevé et le « rework » aurait pu être

encore plus important. Nous voyons d'ailleurs que l'implémentation des exigences stagne, ce qui laisse préjuger d'un retard important.

Dans nos exemples, nous aurions pu également augmenter la lisibilité des indicateurs en ajoutant des zones d'alarme. Par exemple, ici, sur l'indicateur stabilité des exigences :



Une alarme se déclenche lorsque le *Total réel* dépasse le *Total planifié*.

Vert si réel < plan + 5%

Orange entre 5 et 15%

Rouge si > 15%

Et une autre (à partir de la phase de développement) sur la volatilité des exigences (% exigences modifiées / exigences total)

Vert si volatilité < 0,75%

Orange entre 0,75 et 10%

Rouge si > 10%

4 Conclusions

Nous n'avons développé ici qu'un tout petit aspect de la gestion des exigences, uniquement du point de vue de la mesure avec ces 2 indicateurs, que l'on peut considérer comme des « incontournables ».

Remarquez que tout cela n'est que du bon sens, comme c'est généralement le cas pour la mesure, mais le faites-vous ?

Dès lors que vous gérez vos exigences, et que vous avez mis en place un système de mesures, vous serez capable de créer automatiquement ces indicateurs.

5 Détente

Réponse à la question de la page 3 : Non, ce n'est pas le tank, c'est l'escargot !

Connaissez-vous l'appellation « téléphone arabe » dans d'autre langue ?

En **Anglais** : *Chinese whispers / Telephone Game / Russian Scandal*
(« chuchotements chinois / jeu du téléphone / scandale russe »)

En **Allemand** : *Buschtrommel / Stille Post*
(« tambour de la brousse / poste muette »)

En **Espagnol** : *el teléfono estropeado / el teléfono dañado / el teléfono descompuesto*
(« téléphone cassé »)

En **Italien** : *telefono senza fili*
(« téléphone sans fil »)

Et en **Arabe** me direz-vous : **التليفون المكسور**
(« téléphone cassé »)

6 Spirula en bref

Depuis près de 10 ans, Spirula propose des solutions pour mieux estimer et piloter les projets de développement de logiciels et systèmes.

Leader sur son marché, l'offre Spirula – expertise, outils, formation – permet de mieux Comprendre le passé, Piloter le présent et Prévoir l'avenir des projets d'ingénierie logicielle et système.

Nous aidons nos clients à définir les processus de développement les plus efficaces, implémenter des tableaux de bords pour le suivi des projets et augmenter la fiabilité des estimations des coûts, effort et délais des projets.

Nos consultants sont experts dans le pilotage de projet et les estimations et conduisent l'implémentation des bonnes pratiques, comme le CMMI, dont Spirula est un des co-auteurs.

Parmi nos clients, nous comptons des PME/PMI ayant une forte activité de développement de logiciels et de systèmes ainsi que des grands comptes internationaux tel qu'Alstom, BAe, Continental, Philips, Renault, Thales, ...