

Introduction

La dépendance technologique croissante des entreprises exige que leur investissement en systèmes techniques, plus particulièrement les logiciels, soit à la hauteur de leurs aspirations. Étant donné que les logiciels constituent un outil stratégique pour ces entreprises, il devient alors impératif que leur fonctionnement soit conforme à leurs exigences. Par conséquent, les exigences logicielles gagnent en importance, en tant que moyen pour ces entreprises, de connaître à l'avance quelles solutions obtenir. Il s'agit essentiellement, avant de mettre en place de futurs logiciels, d'interpréter et de comprendre les objectifs, les besoins et les préférences, voire les croyances de ces entreprises (clients) et des utilisateurs.

Le développement de logiciels implique au moins autant de communication que d'informatique, mais les programmes d'enseignement et les activités de projet mettent souvent l'accent sur l'informatique plutôt que sur l'aspect communication. Parfois, on ne parvient pas à comprendre que l'élicitation des exigences est principalement un défi d'interaction humaine non encore automatisable (Wiegers et Beatty 2013). Souvent, des problèmes inhérents à la difficulté d'interprétation et de compréhension des besoins, et de leur traduction en exigences bien spécifiées, sont derrière l'annulation de projets logiciels.

L'analyse des exigences est une tâche critique dans le développement de logiciels, qui nécessite d'étudier d'abord le domaine du problème, en vue de développer une meilleure compréhension des vrais objectifs, besoins et préférences des clients et des utilisateurs. Malheureusement, de nombreux développeurs de logiciels, parfois expérimentés, ont encore du mal à comprendre, spécifier et gérer les exigences logicielles. En outre, il y a de nombreux utilisateurs qui, soit

ne veulent pas exprimer leurs vrais besoins (problème de rétention), soit ne peuvent pas le faire (problème de communication), ou bien encore ils ne savent pas le faire (problème d'incompétence).

Bien que les exigences aient été partiellement négligées dans l'ère précédente du génie logiciel, elles ont reçu une plus grande attention au cours des dernières décennies. En effet, l'ingénierie des exigences logicielles est devenue un domaine de connaissance à part entière avec ses propres acteurs et activités mais aussi avec ses propres concepts, techniques, outils et méthodes. Elle peut être définie comme un ensemble coordonné d'activités permettant d'explorer, évaluer, documenter, consolider, réviser et adapter les objectifs, les capacités, les qualités, les contraintes et les hypothèses que le futur système (system-to-be) devrait respecter en fonction des problèmes soulevés par le système existant (system-as-is) et les opportunités offertes par les nouvelles technologies (Van Lamsweerde 2009). Son importance est due au rôle joué par les bonnes exigences dans le succès de tout logiciel. Cependant, quoique la recherche en matière d'ingénierie des exigences soit actuellement très active, elle a encore un long chemin à parcourir avant d'atteindre la maturité de l'état de l'art en génie logiciel. Ce dernier a une très longue histoire comparativement au premier. La tendance dominante dans la recherche sur le génie logiciel a été d'abstraire les constructions de programmation jusqu'au niveau des exigences, plutôt que de propager les abstractions des exigences jusqu'au niveau de la programmation (Mylopoulos 1999).

Nous faisons un premier pas vers le changement de cette situation en ingénierie des exigences par un diagnostic de ses principales causes. En effet, nous considérons que les catégorisations actuelles des exigences, qui constituent l'ossature de l'ingénierie des exigences, souffrent énormément d'ambiguïté et de manque de cohérence. D'après Broy (Broy 2018), jusqu'à présent, une catégorisation bien définie des exigences constitue toujours l'un des défis insuffisamment résolus de l'ingénierie des exigences.

Nous pensons que cette insuffisance est due essentiellement à l'absence d'un cadre référentiel, qui permet de définir les éléments du logiciel et de son environnement auxquels se réfèrent les exigences. En effet, étant donné la complexité énorme des logiciels modernes, nous pensons qu'il est impossible de bien identifier et catégoriser leurs exigences sans une compréhension approfondie des éléments suivants :

-
- le logiciel et ses environnements de développement et d'utilisation :
 - logiciel : composé des fonctions, données, interfaces utilisateurs et techniques ;
 - environnement nécessaire de développement : composé du support humain (développeurs), des deux supports : logiciel et matériel, du support procédural à son développement (les activités de développement) et du contexte cognitif de son développement (les méthodes, les techniques, les artefacts et les outils) ;
 - environnement nécessaire d'utilisation : composé du support humain (utilisateurs finaux), des deux supports : logiciel et matériel et du support procédural à son utilisation (les activités d'utilisation du système englobant) et du contexte cognitif de son utilisation (les concepts, les buts et les règles) ;
 - les facteurs qui affectent le logiciel et ses environnements de développement et d'utilisation :
 - facteurs de qualité tels que l'utilisabilité, la fiabilité, la sécurité et la portabilité ;
 - facteurs de conformité aux différentes contraintes technologiques, économiques, culturelles.

Nous proposons ainsi une nouvelle approche systémique de catégorisation et de modélisation des exigences, baptisée SMART (Systemic Approach of caTegorizing and Modeling Requirements), qui prend en compte les éléments précédents en vue d'offrir ce qui suit :

- au niveau conceptuel :
 - un nouveau cadre référentiel des exigences R2F (Referential Requirement Framework), qui permet de définir avec précision tous les aspects relatifs au logiciel et à ses deux environnements de développement et d'utilisation, qui constituent les sujets d'exigences ainsi que les facteurs de qualité et de conformité qui les affectent. Ce cadre constitue une base solide et nécessaire pour tous les éléments qui suivent. Notre hypothèse étant que ces autres éléments ne peuvent être établis d'une manière systématique sans un référentiel logique et commun ;
 - une nouvelle définition des exigences qui est basée sur R2F ;

- une nouvelle catégorisation hybride des exigences, combinant plusieurs taxinomies et échelles, qui est basée sur R2F ;

- au niveau opérationnel :

- un nouveau modèle de métadonnées des exigences, basé sur le langage des métadonnées des exigences RML (Chikh 2017), qui permet de qualifier les exigences selon plusieurs catégories prédéfinies de métadonnées ;

- un nouveau modèle multivue des exigences qui permet de représenter toutes les catégories possibles des exigences.

Ces deux modèles sont basés sur R2F et sur la catégorisation hybride des exigences du niveau conceptuel.

Cet ouvrage nous a permis de capitaliser sur notre expérience hybride de longue haleine en vue de la partager avec nos lecteurs. Il représente ainsi l'aboutissement d'une expérience de pas moins de vingt ans, d'une part dans le monde académique, à travers l'enseignement en master et doctorat (cours de génie logiciel et de systèmes d'information et plus particulièrement d'ingénierie des exigences logicielles), la supervision de nombreuses thèses de doctorat et de projets de recherche et la publication de plusieurs articles de recherche sur des thématiques liées à l'ingénierie des exigences ; et d'autre part dans le monde industriel, à travers des études et des projets de développement de logiciels et de systèmes d'information, au profit de bureaux d'étude nationaux et internationaux, allant de l'analyse et la conception à l'implémentation et aux tests.

Cet ouvrage est destiné à des lecteurs aux profils et besoins variés : (1) les chercheurs et les étudiants en graduation et en post-graduation en génie logiciel ou dans des domaines voisins tels que les systèmes d'information ; BPM ; BPR, pour approfondir leurs connaissances théoriques et pratiques en matière d'exigences ; (2) les développeurs tels que les analystes ; les concepteurs ; les programmeurs ; et les testeurs, pour s'en servir comme un guide dans leurs projets de logiciels ; (3) les clients et les utilisateurs finaux, pour mieux comprendre et se faire comprendre par les développeurs ; (4) tout autre lecteur passionné, aussi bien novice qu'expérimenté, en vue de satisfaire sa curiosité scientifique en matière d'exigences logicielles. Enfin, l'auteur suppose un prérequis minimal pour la lecture de cet ouvrage, en recommandant que le lecteur type ait très probablement suivi un cours de base sur le génie logiciel et/ou les systèmes d'information, ou bien travaillé dans un projet réel sur les exigences.

Cet ouvrage est composé de deux grandes parties. Après cette introduction, une première partie concernant la revue de la littérature, et qui comprend les quatre chapitres suivants : le premier chapitre présente une étude de cas ; le deuxième chapitre définit le concept d'exigence et d'autres concepts sous-jacents ; le troisième chapitre présente les échelles des niveaux d'exigences ; et le quatrième chapitre présente les taxinomies d'exigences. La deuxième partie quant à elle, est consacrée à la nouvelle approche SMART, et elle comprend les quatre chapitres suivants : le cinquième chapitre qui est consacré à la conceptualisation de cette approche par la proposition d'un nouveau cadre référentiel des exigences baptisé R2F, et à la nouvelle catégorisation hybride qui en découle ; le sixième chapitre qui est consacré à l'opérationnalisation de cette approche par la proposition d'un modèle de métadonnées des exigences, et d'un modèle multivue des exigences ; le septième chapitre qui présente l'utilisation de cette approche à travers différents scénarios ; et le huitième chapitre qui présente l'évaluation de cette approche par sa comparaison avec les approches existantes. Enfin, la conclusion rappelle les apports essentiels de SMART et ouvre quelques perspectives.

Quoiqu'une lecture type de cet ouvrage soit censée suivre l'ordre des huit chapitres, le lecteur doit savoir qu'il est libre de le parcourir, intégralement ou partiellement, dans n'importe quel ordre, selon ses besoins et ses préférences.