

Exigences d'utilisabilité pour les outils d'ingénierie des exigences

Hélène Gaspard-Boulinc, Stéphane Conversy

▶ **To cite this version:**

Hélène Gaspard-Boulinc, Stéphane Conversy. Exigences d'utilisabilité pour les outils d'ingénierie des exigences. IHM 2014, 26th International Conference of "Association Francophone d'Interaction Homme-Machine", Oct 2014, Villeneuve d'Ascq, France. 10.1145/2670444.2670458 . hal-01086959

HAL Id: hal-01086959

https://enac.hal.science/hal-01086959

Submitted on 10 Mar 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Exigences d'utilisabilité pour les outils d'ingénierie des exigences

Hélène Gaspard-Boulinc
Université de Toulouse - ENAC
31055, Toulouse, France
helene.gaspard-boulinc@enac.fr

Stéphane Conversy
Université de Toulouse- ENAC
31055, Toulouse, France
stephane.conversy@enac.fr

RESUME
Les outils d'ingénierie des exigences sont nécessaires pour plusieurs raisons : ils permettent de gérer un nombre croissant d'informations, de maintenir la traçabilité entre exigences, solution et tests, d'évaluer l'impact d'un changement d'exigences sur la solution et les tests.

Cet article propose une analyse des pratiques en ingénierie système dans l'industrie aéronautique, centrée sur l'ingénierie des exigences. A partir de la réalisation d'enquêtes contextuelles, nous avons identifié des scénarios représentatifs de l'activité des ingénieurs système sur la rédaction et la gestion des exigences. L'analyse de ces scénarios nous a permis d'identifier des exigences en termes d'utilisabilité des outils, plus précises que « user-friendliness », qui alimenteront la conception de nouveaux outils d'ingénierie des exigences.

Mots Clés
Ingénierie système; exigence; utilisabilité.

ACM Classification Keywords
H.5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous. Voir http://www.acm.org/about/class/1998/ pour la liste complète des catégories ACM.

INTRODUCTION

La complexité des systèmes conçus par les êtes humains a fait émerger le besoin d'un cadre améliorant la communication et la coopération parmi les parties qui créent et gèrent les composants d'un système. Par exemple, le projet ATLAS de missile balistique intercontinental dans les années 50 aux Etats-Unis [9] a impliqué 18 000 scientifiques, ingénieurs, experts techniques dans les universités et les industries. 70 000 personnes, de l'administration jusqu'à l'usine, réparties dans 22 entreprises, ont participé activement au projet. Ces chiffres montrent le nombre important d'acteurs mais également leur hétérogénéité, et donc la nécessité d'un cadre de coordination entre ces acteurs.

La réponse est l'ingénierie système, définie désormais par des normes [11][12] : c'est une approche collaborative et interdisciplinaire dont l'objectif est de définir, concevoir, réaliser et maintenir des systèmes complexes, en cherchant à élaborer une solution optimisée entre performance, coût et délai.

La notion d'exigence – requirement en anglais- est un élément central de l'ingénierie système. Une exigence est « un énoncé qui traduit ou exprime un besoin et ses contraintes et conditions associées » [13]. Une exigence bien exprimée doit être vérifiable et techniquement réalisable, et doit avoir une seule interprétation possible. Ainsi, si l'exigence est exprimée en langage naturel, il est recommandé d'adopter des règles de syntaxe telles que [Condition] [Sujet] [Verbe] [Complément] [Contrainte], où le sujet est le système d'intérêt (**Figure 1**). Une exigence possède des attributs tels qu'un identifiant unique, une priorité, la partie prenante intéressée par l'exigence, la justification. Elle possède aussi des attributs de gestion (date de création, date de dernière modification, auteur, version du système implémentant l'exigence).

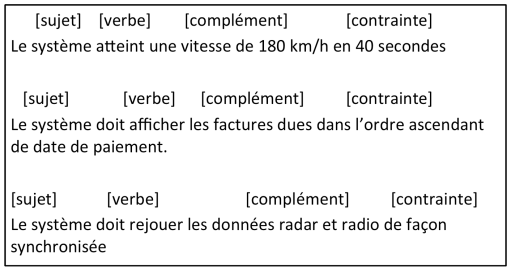


Figure 1: exemples d'exigences

Un référentiel (baseline) d'exigences est un ensemble d'exigences validées par les parties prenantes à une date donnée. Le référentiel guide les autres activités du processus d'ingénierie système et représente la définition du problème à résoudre. Il évolue dans le temps, du fait de l'évolutivité des exigences. Il doit lui aussi respecter des bonnes propriétés définies dans [13].

L'ingénierie des exigences (Requirement Engineering) désigne les activités d'ingénierie système qui visent à établir et maintenir les exigences à remplir par le système. Elle nécessite l'utilisation d'outils pour plusieurs raisons: les outils permettent d'organiser et

gérer un nombre important d'exigences, de maintenir la traçabilité entre exigences-solution-tests, d'évaluer l'impact d'un changement d'exigences sur la solution et les tests.

L'utilisabilité [16] d'un outil est primordiale dans l'efficacité de l'utilisateur : plus précisément, l'utilisabilité est « le degré selon lequel un produit peut être utilisé, par des utilisateurs identifiés, pour atteindre des buts définis avec efficacité (précision et degré d'achèvement des objectifs), efficience (rapport entre les ressources dépensées et l'efficacité), et satisfaction (absence d'inconfort, attitudes positives dans l'utilisation du produit), dans un contexte d'utilisation spécifié ».

L'objectif de cet article est de préciser les exigences d'utilisabilité des outils d'ingénierie des exigences. Pour cela, nous décrivons d'abord les activités prescrites de l'ingénieur système. Nous proposons ensuite un état de l'art sur les travaux relatifs à la définition et l'évaluation de l'utilisabilité des outils d'ingénierie des exigences. Pour compléter ces travaux, nous adoptons une approche de conception participative, avec la réalisation d'enquêtes contextuelles dans l'industrie aéronautique. L'analyse de ces enquêtes nous a permis d'identifier des scénarios d'usage, avec les forces et les faiblesses des outils utilisés. Nous en déduisons des exigences d'utilisabilité sur les outils d'ingénierie des exigences, justifiées par nos scénarios.

ACTIVITES PRESCRITES DE L'INGENIEUR SYSTEME

La norme IEEE 1220 [12] propose un processus global d'ingénierie système divisé en 8 sous-processus centrés sur les activités techniques d'ingénierie (Figure 2) : analyse des exigences, validation des exigences, analyse fonctionnelle, vérification fonctionnelle, analyse système (à droite), synthèse, vérification du design, contrôle.

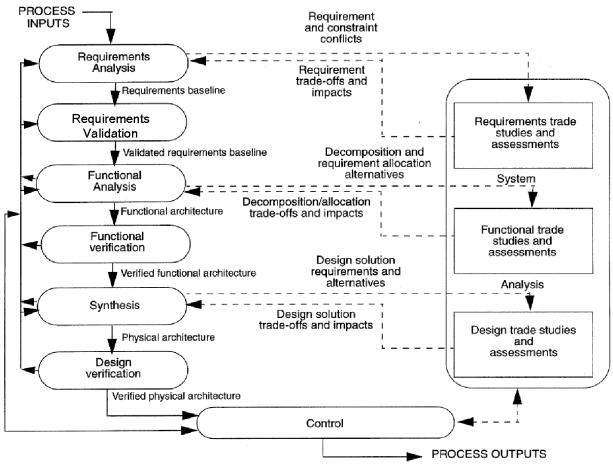
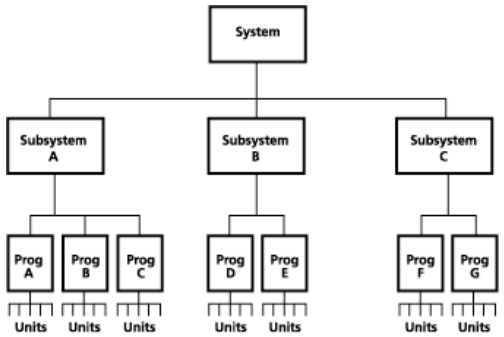


Figure 2: les processus d'ingénierie système selon la norme IEEE 1220-2005

Les processus sont reliés entre eux par des entrées-sorties. Les deux processus sur les exigences (Requirements Analysis et Requirements Validation) produisent un référentiel vérifié d'exigences. Ce référentiel d'exigences représente la spécification du système à concevoir, en termes de fonctions, performances et interfaces avec l'environnement extérieur. Le référentiel d'exigences est en entrée des processus de conception fonctionnelle, qui produisent une architecture fonctionnelle vérifiée. Cette dernière est en entrée de l'analyse système qui va produire en sortie une architecture physique, c'est-à-dire une décomposition du système par composant, avec l'allocation des exigences par composant. Le processus de contrôle prend en entrée toutes les sorties des processus pour assurer leur documentation et sa mise à jour de façon ordonnée. Les flèches à gauche à double sens, ainsi que les flèches en pointillés, montrent la possibilité de pouvoir revenir sur le processus précédent de façon itérative tant qu'on identifie des problèmes (« variance and conflict » dans la norme).

Plus on avance dans la conception du système, plus les exigences sont affinées en exigences détaillées, pour atteindre une spécification précise des composants et des interactions entre composants auxquels les exigences sont allouées. Il en résulte une multiplication du nombre des exigences et une complexification des liens entre les différents éléments d'ingénierie (exigences, exigences détaillées, composants, interfaces entre composants, tests). Pour illustrer cela, nous pouvons reprendre l'exemple simple proposé par [24], d'un système composé de 3 sous-systèmes, avec 6 exigences (Figure 3a), où les exigences sont allouées par sous-systèmes (Figure 3b) et affiner en 21 exigences (Figure 3c).



SYSTEM REQUIREMENTS	SUBSYSTEM A	SUBSYSTEM B	SUBSYSTEM C
SYS 001	X	X	
SYS 002	X		X
SYS 003		X	
SYS 004	X	X	X
SYS 005			X
SYS 006	X	X	

SYSTEM REQUIREMENT	SYSTEM A REQUIREMENT	SYSTEM B REQUIREMENT	SYSTEM C REQUIREMENT
SYS 001	SSA 001 SSA 002	SSB 001	–
SYS 002	SSA 003 SSA 004 SSA 005	–	–
SYS 003	–	SSB 002 SSB 003	–
SYS 004	SSA 006 SSA 007	SSB 004 SSB 005 SSB 006	SSC 003
SYS 005	–	–	SSC 004 SSC 005
SYS 006	–	SSB 007 SSB 008	–

Figure 3a: décomposition d'un système en 3 sous-systèmes, figure 3b: allocation des exigences selon les sous-systèmes, figure 3c : exigences affinées par sous-système.

Mis à part les entrées/sorties qui lient les processus et l'aspect itératif entre processus, les normes ne précisent pas de cycle de vie d'un projet de conception système : c'est aux industriels de choisir un cycle de vie parmi les modèles existants tels que le cycle en cascade ou le cycle en spirale, que [24] liste de façon exhaustive.

Cependant, les industriels affichent généralement l'adoption d'un cycle en V [2] [20], qui favorise une séquence chronologique des processus dans le temps et une répartition du travail entre les différentes parties prenantes (client, entreprise ingénierie système, fournisseurs), comme illustré dans la Figure 4, adaptée d'une proposition faite par [22].

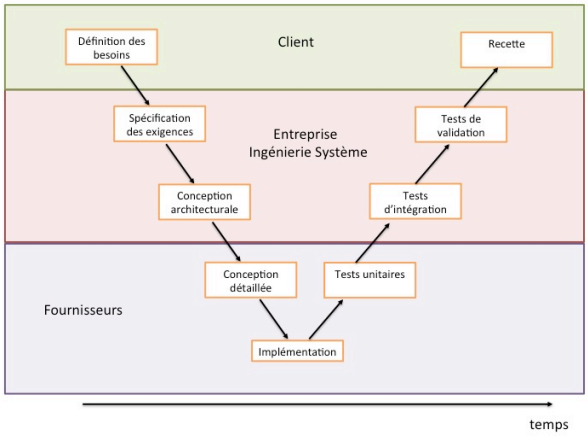


Figure 4: répartition du travail entre parties prenantes selon le cycle en V

D'autre part, les systèmes conçus par les industriels étant de plus en plus complexes, les ingénieurs élaborent et maintiennent un nombre croissant d'exigences et de liens entre exigences-composants-interfaces. Pour les assister dans ces activités, ils utilisent des outils de gestion des exigences, nombreux sur le marché : la société savante internationale en ingénierie système, l'INCOSE, a par exemple répertorié 34 outils différents [10]. On peut citer les plus connus tels que DOORS édité par IBM, ou CORE édité par Vitech.

L'utilisabilité d'un outil est primordiale dans l'efficacité de l'utilisateur dans ses tâches. Une erreur dans l'ingénierie d'un système peut avoir des conséquences sur la sécurité des opérations du système. De plus, à l'heure où la compétition entre industriels met une forte pression sur les coûts et le temps de développement d'un système, le choix d'un outil efficace pour la réalisation des tâches d'ingénierie des exigences devient stratégique.

ETAT DE L'ART SUR L'UTILISABILITE DES OUTILS D'INGENIERIE DES EXIGENCES

Nous proposons un état de l'art sur les travaux relatifs à la définition et l'évaluation de l'utilisabilité des outils d'ingénierie de exigences. Cet état de l'art est structuré selon trois axes : le cadre d'analyse des outils (framework), la méthode d'analyse utilisée, les résultats de l'analyse en termes d'exigences d'utilisabilité.

Le rapport technique ISO/IEC TR 24766 [14] décrit les capacités attendues des outils d'ingénierie des exigences, et propose de classer ces capacités selon huit catégories : élicitation, analyse, spécification, modélisation, vérification et validation, management, traçabilité, et « autres ». Dans la catégorie « autres », nous trouvons deux aspects qui nous intéressent plus particulièrement en IHM : l'accès collaboratif et l'interface graphique. La collaboration sur les exigences est envisagée à travers la gestion des accès aux données : gestion des rôles et des groupes utilisateurs, accès aux données en lecture ou écriture selon les rôles. Les aspects relatifs à l'interface utilisateur sont très vagues : le rapport préconise que les outils fournissent des vues multiples (sans préciser lesquelles), des entrées et des manipulations interactives graphiques des données (sans préciser), une interface web, les fonctions de copier-coller, et des capacités classiques des systèmes d'exploitation tels que le multifenêtrage et l'accès concurrent à plusieurs fonctions. Ce rapport se présente comme une aide à la comparaison et à l'évaluation des outils pour les entreprises dans un processus d'achat, mais ne propose pas lui-même d'évaluation des outils avec ce cadre. L'INCOSE a par contre utilisé ce cadre pour évaluer les outils répertoriés [10], mais en demandant aux éditeurs des outils de faire l'évaluation : chaque outil est (forcément) bien évalué par son propre éditeur.

[17] propose une évaluation statistique de 38 outils utilisés par les praticiens, en utilisant également l'ISO/IEC TR 24766 comme cadre, mais ne retient que l'aspect de l'interface web comme critère, montrant la difficulté de préciser les autres aspects. La méthode est contestable car ce sont également les éditeurs d'outils qui ont évalué leur outil sur la base d'un questionnaire.

[3] propose un *cadre d'analyse* des visualisations à partir des activités d'ingénierie des exigences et d'un cycle de vie des exigences. Les activités d'ingénierie des

exigences sont très similaires aux catégories proposées par ISO/IEC TR 24766 : élicitation, modélisation et analyse, communication et négociation, vérification et validation. Nous notons cependant deux différences : les activités de négociation et communication sont reconnues comme des activités de premier plan, alors que les activités de management des exigences n'apparaissent plus de façon explicite. Ces activités de management sont transférées dans une phase du cycle de vie des exigences. Ce cycle de vie des exigences est composé de 4 phases : mise en place, élaboration, raffinement, spécification, ces phases se succédant de façon chronologique. En matière de *méthode*, l'évaluation est réalisée par 4 chercheurs en ingénierie des exigences, en utilisant ce cadre croisé activités/phases, et porte sur des visualisations issues de 29 articles de recherche. Ils introduisent également une catégorisation des visualisations : tabulaire, relationnelle, séquentielle, hiérarchique, quantitative/métaphorique. En matière de *résultats*, dans leur effort de catégorisation, les auteurs se rendent compte du recouvrement entre activités et phases, lié notamment à la nature itérative du travail fait sur les exigences. Ils notent une prévalence des visualisations de type relationnelle et hiérarchique, sur les phases de raffinement et de spécification. C'est pour la phase d'élaboration des exigences que la plus grande variété de visualisation est constatée (séquentielle, tabulaire, hiérarchique et relationnelle). En revanche, peu de visualisations sont proposées pour les activités de vérification et validation des exigences. Les auteurs concluent en préconisant le développement de visualisation pour les tâches de vérification et validation, et l'ouverture vers des visualisations quantitatives/métaphoriques, sans formuler d'exigences plus précises. Ils préconisent également d'aller vers une meilleure compréhension des tâches à instrumenter à travers les visualisations, conscients des limites de leur cadre d'évaluation. Nous pouvons noter que l'interaction sur les visualisations n'est pas du tout prise en compte dans leur évaluation.

Au contraire, [21] identifie l'interaction sur les visualisations d'exigences comme un moyen d'exploration, de découverte de connaissances et de raisonnements efficaces. Une analyse des récents travaux sur la visualisation des exigences est proposée par les auteurs, avec également un cadre. Ce *cadre* n'est plus centré sur les activités d'ingénierie des exigences, mais sur un processus de visualisation de données par l'utilisateur : l'utilisateur sélectionne des données, les données correspondent à un « modèle » du système, l'utilisateur spécifie le « modèle », le « modèle » est transformé en visualisation, l'utilisateur interagit sur la visualisation pour comprendre et découvrir. L'interaction sur la visualisation est proposée comme de première importance pour l'utilisateur, alors que la sélection des données et l'édition du modèle sont évaluées comme secondaires. Nous retenons de ce cadre les « questions opérationnelles » identifiées sur les visualisations : des vues multiples, avec des possibilités de navigation, recherche, filtrage et annotation sur ces vues. En matière de *méthode*, l'évaluation est conduite par les deux auteurs de l'article, puis discutée avec un troisième chercheur, sur 8 outils issus de la littérature académique ayant une approche de « visual analytics » (VA) sur les exigences. Ils notent qualitativement sur une échelle de 4 l'atteinte des questions opérationnelles de leur cadre. *Les résultats* montrent que les aspects visualisation, en termes de vues multiples, navigation entre vues, recherche, filtrage, requêtes et annotation sont mal couverts par les outils proposés. La proposition de vues multiples par les outils est cependant le point le mieux traité. Les aspects liés à la découverte de connaissances, tels que la détection d'anomalies, les explications détaillées, l'aide au raisonnement, sont également mal couverts. Les auteurs ne formalisent pas les résultats sous forme d'exigences à remplir par les visualisations et les interactions, mais restent sur cette notion de « question opérationnelle ».

Dans un deuxième temps, les auteurs reportent une étude de cas sur un projet réel, impliquant 3 utilisateurs, dans laquelle ils demandent aux participants d'utiliser l'outil qu'ils ont conçu, RecVisu+, pour passer en revue et analyser les exigences du projet réel des utilisateurs. Les auteurs n'utilisent plus leur cadre d'évaluation, mais se centrent sur les tâches d'ingénierie des exigences (quelles sont les tâches qui ont besoin de VA? comment VA supportent ces tâches ? quels bénéfices peut-on attendre ?). En matière de *méthode*, l'évaluation est faite lors de 4 séances, alternant observation et discussion, par 3 utilisateurs-ingénieurs, sur l'outil RecVisu+. *Les résultats* de l'étude sont présentés sous forme de tâches d'ingénierie des exigences intéressantes à instrumenter par du visual analytics, et comment l'outil RecVisu+ fournit ou non ce support. Les auteurs ne formalisent pas ces résultats sous forme d'exigences en tant que telles pour les outils d'ingénierie des exigences, mais plutôt comme des améliorations à apporter à leur outil.

[8] affiche clairement comme objectif de préciser les exigences pour les outils de gestion des exigences (c'est son titre). Les auteurs prennent comme *cadre d'analyse* le standard ISO/IEC 9126 d'évaluation des caractéristiques qualité d'un logiciel [15]. La *méthode* utilisée est basée sur l'expérience des auteurs dans des projets de développement de système. Les nombres de projets, d'années d'expérience ne sont pas précisés. Les outils rencontrés lors de ces expériences ne sont pas précisés non plus. En termes de *résultats*, les auteurs présentent un catalogue d'exigences selon trois acteurs différents, le développeur, l'administrateur du projet, et l'administrateur de l'outil. Bien que les auteurs annoncent que les exigences sont purement fonctionnelles et ne couvrent pas des aspects généraux tels que la « user-friendliness », on peut identifier des exigences d'utilisabilité dans le catalogue. Il existe notamment une partie « vues », dans laquelle les exigences restent floues (« graphical views of the

requirements should be available », « all users must be able to customize the standard views without changing the template »).

Synthèse

Notre état de l'art montre de nombreuses tentatives de définition de cadres d'analyse des outils, qui se révèlent difficiles à utiliser par les auteurs mêmes de ces cadres. Cette difficulté d'utilisation peut s'expliquer par une similitude de ces cadres avec les activités prescrites de l'ingénierie des exigences. Une meilleure compréhension des tâches à instrumenter est d'ailleurs préconisée de façon récurrente [1] [5] [21].

Les méthodes d'analyse des outils sont à base de jugement d'expert [3] [21] [8], ou de questionnaires remplis par les éditeurs des outils [10][17]. Lorsque ces évaluations impliquent des utilisateurs, comme dans [21], les résultats de ces évaluations ne sont pas généralisées à des exigences que doivent remplir les outils, mais restent centrés sur des améliorations de l'outil évalué.

Finalement, les exigences sur l'utilisabilité des outils d'ingénierie des exigences restent mal formulées si on y applique les règles de bonne formulation d'une exigence (**Figure 1**). L'exigence la plus récurrente est de proposer des vues multiples des mêmes données (viewpoint ou views) à l'utilisateur, mais sans les préciser de façon non ambiguë. L'interactivité sur les vues est présentée par [21] comme prometteuse pour la découverte de connaissances par l'ingénieur (détection d'anomalie, explication détaillée, raisonnement par hypothèse ou scénario), mais n'est pas formalisée sous forme d'exigences précises.

ANALYSE DE L'ACTIVITE DE L'INGENIEUR SYSTEME DANS L'INDUSTRIE AERONAUTIQUE

Nous proposons de construire une meilleure compréhension des tâches d'ingénierie des exigences à instrumenter, telles qu'elles sont actuellement réalisées par les industries aéronautiques. Après avoir expliqué la méthode utilisée, nous présentons les résultats de notre analyse de l'activité en caractérisant d'abord les projets et les données produites lors de l'activité sans en révéler les contenus détaillés pour des raisons de confidentialité. Nous présentons ensuite des scénarios d'usage, en formalisant les forces et faiblesses relevées sous forme d'exigences d'utilisabilité.

Méthode

Nous avons choisi comme méthode la réalisation d'enquêtes contextuelles auprès d'ingénieurs, afin de bien identifier les problèmes réels des ingénieurs, comme suggéré par [1]. Notre objectif est d'identifier les tâches réalisées, cerner les problèmes d'utilisabilité de leurs outils, et formuler des exigences plus précises en termes d'interactions, de visualisations et de collaboration.

9 ingénieurs ont été interviewés de janvier à mars 2014, de 4 entreprises différentes, dans l'industrie aéronautique. 6 ingénieurs sont confirmés, avec au moins 10 ans d'expérience, les 3 autres ayant 4 ans d'expérience. Chaque interview a duré 2 heures, et a fait l'objet d'un enregistrement vidéo, sauf pour 3 ingénieurs pour des raisons de confidentialité. Une transcription de chaque interview a été réalisée, en y incluant des captures de vidéo pour préciser les propos. Des données ont été récupérées auprès de 6 ingénieurs, telles que des schémas d'architecture, des documents de spécification, des référentiels d'exigences, des matrices de traçabilité entre exigences et tests, des présentations, des cahiers de tests, des comptes rendus de réunion, des échanges de mails entre ingénieurs. Ces données ont été exploitées lors des transcriptions pour identifier les détails des visualisations et des interactions utilisées lors des interviews.

Caractérisation des projets et des données

Parmi les ingénieurs interviewés, nous avons 5 projets *d'évolution* de systèmes existants. Le système est déjà opérationnel, les composants du système et les interfaces entre ces composants sont connus, il s'agit de faire évoluer le système existant avec un nouveau service. 3 projets sont relatifs à la prise en compte d'évolutions de systèmes bord (IHM cockpit d'un hélicoptère, service d'anti-collision d'un avion de transport, évolution d'une suite avionique d'un turbopropulseur). 2 projets sont relatifs à la prise en compte d'évolutions de système sol (IHM et service de collaboration pour la gestion des départs, service de communication entre actuel et futur systèmes de contrôle du trafic aérien).

Cette situation est assez fréquente pour l'ingénieur système, la durée de vie des systèmes sol et bord étant d'au moins 20 ans. Pendant cette durée, la maintenance corrective et évolutive doit être assurée : la prise en compte des anomalies (ou faits techniques) suit souvent le même processus d'ingénierie système qu'une évolution. La mise en place d'une nouvelle version d'un système est coûteuse, car elle exige la production de tous les documents de preuve de conformité pour la certification.

Parmi les autres projets observés, un projet est relatif à un nouveau système de radio-téléphone pour le contrôle aérien, avec appel d'offre, discussion et choix des fournisseurs. Les 3 autres projets concernent des projets exploratoires, où des travaux de modélisation système sont mis en œuvre en préparation d'affaires futures.

Les documents produits sont des documents de spécification (System SubSystem Specification, Software Requirements Specification), où le nombre d'exigences système va de 100 à 2500 pour une nouvelle fonction. Quand on passe au niveau sous-système, une exigence système est affinée en au moins 3 exigences sous-systèmes. Quand on passe niveau logiciel, chaque exigence sous-système est encore affinée, avec un

facteur plus important s'il s'agit d'IHM. En effet, les exigences peuvent être très détaillées, avec des captures d'écran issues de prototypage à l'appui (*ex : If more than two aircraft are displayed in the ITP traffic list, the 'TOO MANY TRAFFIC' message shall be displayed in amber large font characters on the LSK 2R label line (refer to Figure 68)*). Le système FMS (Flight Management System) d'un avion est défini par 16 000 exigences.

Les scénarios présentés ci-après vont être illustrés par des extraits de ces documents. Nos exigences d'utilisabilité seront intercalées entre crochets dans les scénarios, les exigences seront détaillées dans le tableau de la Figure 8 afin de ne pas surcharger la lecture des scénarios. Les scénarios sont numérotés afin de pouvoir les référencer.

1- Conception exploratoire et partagée de l'architecture entre ingénieur système et fournisseurs des composants

Dans trois entreprises différentes, nous avons constaté que l'ingénieur système adopte une façon de travailler différente de l'activité prescrite que nous avons présentée précédemment. En effet, il ne va pas réfléchir seul à l'allocation des exigences système entre composants, puis rédiger les exigences pour chaque composant, pour enfin les transmettre aux fournisseurs des composants.

A l'inverse, il choisit de lister les problèmes de conception que soulèvent les exigences système à remplir (ou les anomalies à traiter), et travaille avec les fournisseurs des composants lors d' « ateliers » sur ces problèmes de conception. Ce travail débouche sur un panel de solutions techniquement réalisables en termes d'allocation d'exigences, car proposées et discutées avec les fournisseurs. Cela permet ensuite à l'ingénieur système de faire un choix entre ces solutions et de rédiger les exigences d'interfaces entre composants. Les exigences affinées par composants sont initiées par l'ingénieur système et complétées par les fournisseurs.

Cette méthode pragmatique est adoptée pour des raisons d'efficacité. Un des ingénieurs système interviewés, se positionnant au sein de son entreprise comme un garant des processus, tient un discours contradictoire: il regrette que "l'IS n'ait pas le vent en poupe" et que "les gens ne suivent pas les processus". Mais il constate aussi l'efficacité d'un travail préalable à la rédaction des exigences en collaboration avec les fournisseurs des différents composants du système. Un autre ingénieur interviewé, travaillant au niveau sous-système, se réjouit de ce travail avec les ingénieurs système, fustigeant « la forme contre le fond », et indiquant l'importance d' « avoir les bureaux les uns à côté des autres ».

En termes d'exigence d'utilisabilité, nous en déduisons que [le système doit être flexible (« pliant ») vis-à-vis des processus d'ingénierie système] [7], pour permettre l'utilisation des méthodes pragmatiques observées. Cette exigence est certes de haut niveau, mais nous pouvons l'affiner en exigences détaillées en analysant comment cette méthode de travail est actuellement instrumentée, à travers les scénarios d'usage suivants.

Lister les problèmes de conception en dessinant l'architecture

L'ingénieur système utilise un outil de dessin, tels que Visio pour construire un schéma de l'architecture. Le système d'étude est un système pour le contrôle aérien. Il est opérationnel, les composants du système et les interfaces entre ces composants sont donc connus. Il s'agit de faire évoluer le système existant avec un nouveau service. L'ingénieur système utilise ce schéma pour identifier les problèmes de conception, et cela de façon exhaustive. La construction du schéma lui permet d'avoir une vision d'ensemble, et d' « être sûr de ne rien louper » [le système doit permettre la conception exploratoire][6]. Il préfère utiliser des logiciels de dessin plutôt que des outils de modélisation système car les dessins vont évoluer dans le temps et doivent être facilement modifiables [le système permet de dessiner un schéma d'architecture facilement modifiable]. « Ce sont des hypothèses d'étude, et du fait que c'est un dessin, cela se change de manière très simple ». C'est au moment où il construit le schéma qu'il va discerner/découvrir les problèmes à résoudre (« avec ce genre de schéma, les sujets apparaissent d'eux-mêmes »). En parallèle à la construction du schéma, il va alors rédiger ces problèmes dans un fichier excel, en les structurant par thème/questions/référence vers le document de spécification système (problème d'indirection). Les liens entre les problèmes de conception listés et le schéma ne sont pas explicités, ils sont dans la tête de l'ingénieur. Quand ils le sont (Figure 5), ils sont fait dans un fichier Word, dans lequel le schéma est inséré, et sur lequel on rajoute une flèche et du texte se référant au numéro du problème, créant une indirection [le système permet d'éditer une liste de problèmes de conception, en lien avec le schéma d'architecture et les exigences système].

Partager et discuter les problèmes de conception et l'architecture

Dans un deuxième temps, le schéma d'architecture et la liste des problèmes de conception sont présentés aux fournisseurs des composants du système lors d'une réunion de travail. L'objectif de la réunion pour l'ingénieur système est de présenter son schéma et les problèmes de conception identifiés aux fournisseurs, et voir avec eux s' « ils ont déjà la solution de leur côté pour réaliser la fonction ». Nous n'avons pas assisté à ce genre de réunion, mais selon les sujets, la réunion aborde un par un les problèmes de conception identifiés dans le document, permet d'avoir une compréhension partagée des problèmes et envisager et confronter des pistes de solution réalisables [le système permet de partager la listes des problèmes de conception et le schéma en réunion colocalisée ou à distance]. Les pistes

de solution ne sont pas notées en séance dans les documents, mais précisées a posteriori par messagerie électronique. Lorsqu'on leur demande les résultats de ces réunions, l'ingénieur utilise sa messagerie en filtrant sur le mot « atelier » ou sur la date de la réunion, et prend du temps pour trouver l'information. Il n' y a pas vraiment de compte-rendu formel de ces ateliers : l'ingénieur compte alors essentiellement sur sa mémoire, sa connaissance du système, son expertise « je n'ai pas besoin de compte-rendu pour répondre aux questions techniques ». En revanche, en cas de transfert de l'activité vers un collègue, « ce serait plus difficile » [le système permet aux différents utilisateurs de renseigner leur analyse sur les problèmes de conception].

Les exigences système initiales peuvent être modifiées suite à ces analyses, ainsi que le schéma d'architecture, pour intégrer les pistes envisagées [le système permet de gérer plusieurs versions de schémas d'architecture, de visualiser les différences entre les schémas, de lier les différences aux analyses des problèmes de conception].

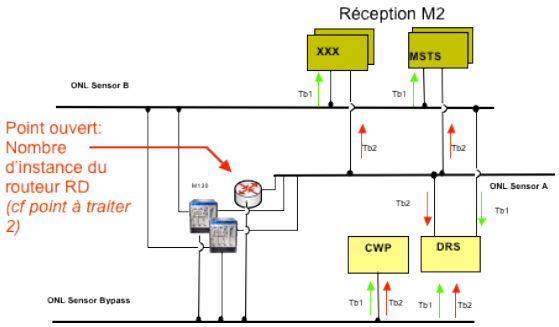


Figure 5: un schéma d'architecture incluant un référence indirecte à un point à traiter

Suivre l'avancement du travail sur les problèmes de conception, l'architecture et les exigences

Certains problèmes de conception sont clos, et l'ingénieur système va commencer à rédiger les exigences pour chacun des composants dans un fichier Word, en indiquant « TBD » (to be defined), que le fournisseur doit compléter. Pour cela, il va intégrer son schéma dans un fichier Word, ajouter au-dessus les numéros des exigences, et créer un paragraphe « exigences » dans lequel on va retrouver les numéros avec le texte associé (Figure 6). [le système permet de créer et modifier une ou plusieurs exigences à partir d'un élément graphique du schéma d'architecture][le système permet d'accéder aux exigences à partir d'un élément graphique du schéma d'architecture].

Certains problèmes ne sont pas clos, et doivent être étudiés par un ou plusieurs fournisseurs. Pour gérer ces problèmes non clos, l'ingénieur utilise un logiciel de gestion de projet, dans lequel il va reporter en tant qu'activités du planning les différentes solutions envisagées avec un temps alloué à chacune des solutions. Cependant, pour expliquer l'avancement du travail, l'ingénieur va revenir sur le schéma d'architecture, le passer en revue avec son curseur de souris et en verbalisant « ça c'est bon, ça c'est en cours, etc ». [Le système permet de suivre l'avancement des analyses des problèmes de conception à travers le schéma d'architecture].

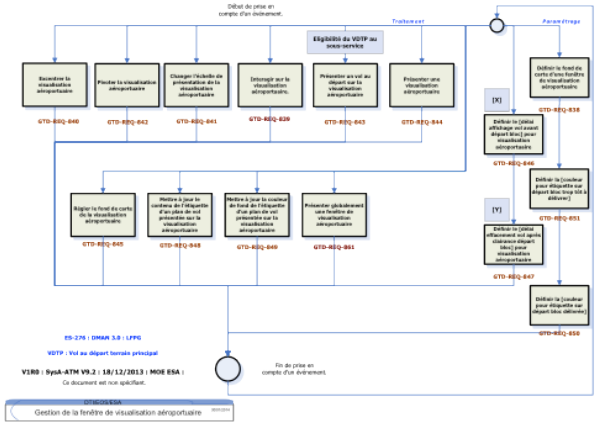


Figure 6: schéma d'architecture fonctionnelle, où les exigences sont en rouge sous les objets graphiques, avec une indirection sur l'énoncé de l'exigence via le numéro

2- Une modélisation fastidieuse et illisible

Trois ingénieurs interviewés utilisent des outils de modélisation système, utilisant des formalismes tels que SysML. Certaines entreprises s'orientent en effet vers ce genre d'outils, afin de contrer la croissance incontrôlée des exigences (« un schéma vaut mieux qu'une tonne de texte »). L'intérêt de ces outils est de pouvoir éditer l'architecture fonctionnelle, préciser des performances et des contraintes sur les fonctions dans un deuxième temps, pour ensuite pouvoir simuler le fonctionnement du système modélisé. La modélisation et la simulation de solutions alternatives permettent alors de comparer les solutions et de faire des choix argumentés.

Nous avons pu constaté lors des interviews que les outils de modélisation ne sont pas utilisés de cette façon. En effet, seule la solution retenue est modélisée, le travail de comparaison des solutions et de décision se faisant par jugement d'expert lors des ateliers entre ingénieurs que nous avons précédemment décrits. La conséquence est que les solutions alternatives ne sont pas enregistrées, ni les discussions et compromis qui ont conduit à cette solution.

Cette situation peut s'expliquer par deux points que nous avons pu observer : les problèmes d'utilisabilité lors de l'édition du modèle, et les problèmes de lisibilité du modèle produit. Par exemple sur le modèle de la figure 7, l'alignement des différents rectangles a été réalisé manuellement par l'ingénieur, un par un. Les rectangles complètement à droite représentent des instances d'un même acteur du système. Si l'ingénieur a oublié une sortie sur les rectangles à droite du modèle, il peut ajouter une sortie sur l'acteur, cela ajoutera la sortie sur tous les rectangles. Mais s'il veut positionner

graphiquement la sortie à un endroit donné sur le côté du rectangle, il faut qu'il réalise cette opération sur chacun des rectangles (« c'est boite par boite, il manque des macros »). Il faut de plus qu'il déplace les rectangles un par un, car trop près du bord de la fenêtre [Le système permet de modifier les propriétés graphiques d'un ensemble d'objets]. Si l'ingénieur veut modifier les exigences sur l'instance, il double-clique sur le rectangle, ce qui déclenche l'ouverture d'une fenêtre par dessus le modèle, qui contient 11 onglets (

Figure 7: exemples de visualisation de modèle et de fenêtre de gestion des propriétés) : « on a l'impression que les propriétés sont dans l'ordre inverse de ce dont on a besoin ». [Le système permet de rechercher facilement une propriété du domaine et de modifier cette propriété du domaine pour un ensemble d'objets].

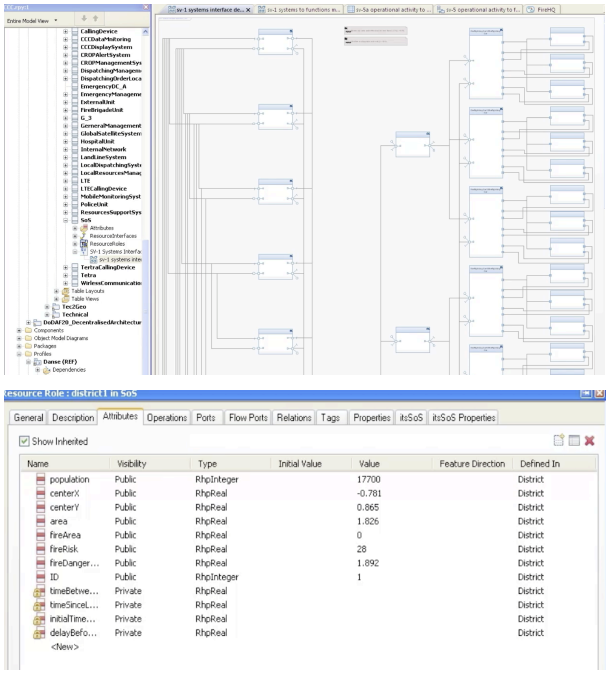


Figure 7: exemples de visualisation de modèle et de fenêtre de gestion des propriétés

Enfin, comme il n'y a pas de noms sur les boîtes (de gauche à droite il s'agit de quartiers d'une ville, de quartier général, de casernes, et de pompiers), le modèle n'est pas facilement compréhensible par une autre personne [Le système permet d'associer une représentation visuelle à un objet]. L'ingénieur reconnait que le modèle n'est pas lisible pour soi-même quelques semaines après : « le travail n'est pas quotidien, l'outil devrait permettre de se souvenir de ce qu'on a fait, où on en est, comment on fait, quelle est la prochaine étape à réaliser » [Le système permet d'identifier les dernières actions d'édition réalisées sur le schéma d'architecture].

3-Consitution et maintenance du référentiel d'exigences

Certains ingénieurs interviewés sont responsables de la gestion des exigences. "DOORS est super pour trois choses : maintenir la traçabilité, créer des baselines, permettre l'accès multi-utilisateurs avec la possibilité de verrouiller des zones." En effet, DOORS permet de construire et maintenir le référentiel des exigences pour une version de système. Il permet de générer selon des modèles les documents nécessaires à la certification, notamment les preuves de traçabilité exigées par la norme DO178B [4] : « Traceability between system requirements and software requirements should be provided to enable verification of the complete implementation of the system requirements and give visibility to the derived requirements ».

L'édition et la visualisation des liens dans DOORS soulèvent des problèmes d'utilisabilité. Pour construire la traçabilité sur chaque exigence il y a la possibilité d'insérer des liens (flèche jaune *in* ou flèche rouge *out*) qui renvoient seulement vers d'autres éléments existant déjà dans DOORS. Si ces élements n'existent pas, l'ingénieur système est obligé de créer d'abord ces éléments, pour ensuite pouvoir les référencer comme lien, lui faisant perdre son contexte [Le système permet de créer un lien à partir d'une exigence en conservant le contexte de l'utilisateur].

Le retour d'information sur une exigence système qui a été dérivée en une ou dix exigences est le même, ce qui donne l'impression que « tout est à plat , il y a un problème de structure». [Le système permet sans interaction de voir le nombre d'exigences dérivées pour une exigence] [Le système permet d'avoir une vue d'ensemble structurée des exigences]. Pour voir les exigences dérivées, il faut sélectionner la flèche jaune, déclenchant l'affichage de la liste des chemins de répertoire des exigences dans DOORS (ex : /Projet/Referentiel/Evolutions/R347) : il est alors difficile de comprendre le contenu des exigences dérivées [Le système permet d'atteindre l'énoncé des exigences dérivées à partir d'une exigence en une interaction au maximum]. De plus, les exigences dérivées ne sont accessibles qu'une par une à travers la liste [Le système permet d'atteindre l'ensemble des exigences dérivées à partir d'une exigence].

Notre ingénieur système est chargé de la vérification de la cohérence : il a mis en place dans un document Word un glossaire de termes, mais ce glossaire n'est pas stocké dans DOORS. La conséquence est l'utilisation de termes différents dans les exigences pour désigner le même concept (ex : « famille de vols » vs « liste de vols ») [Le système propose à l'utilisateur les termes conformes à un glossaire].

Le problème existe également avec les verbes : « si un nouvel ingénieur arrive sur le projet, il va créer une exigence *interagir avec la timeline de séquencement* alors qu'il existe déjà une exigence *manipuler la timeline de séquencement* ».[Le système permet à l'utilisateur de trouver les exigences similaires sémantiquement en mettant en valeur les similitudes]. Nous précisons « à l'utilisateur » dans cette exigence, afin de privilégier des

Enoncé	Justification
Le système doit être flexible vis-à-vis des processus d'ingénierie système : • éditer une liste de problèmes de conception, **en lien** avec le schéma d'architecture et les exigences système non finalisées ; • partager la liste des problèmes de conception et le schéma d'architecture en réunion colocalisée ou à distance (cscw) ; • éditer l'analyse sur les problèmes de conception par plusieurs utilisateurs (cscw) ; • visualiser les différences entre plusieurs versions de schémas d'architecture ; • lier les différences aux analyses des problèmes de conception.	scénario 1
Le système permet de dessiner un schéma d'architecture *facilement* modifiable : • modifier les propriétés graphiques d'un ensemble d'objets ; • rechercher facilement une propriété du domaine d'un objet ; • modifier une propriété du domaine pour un ensemble d'objets ; • associer une représentation visuelle à un objet.	scénarios 1 et 2
Le système permet de suivre l'avancement du travail **à partir** du schéma d'architecture : • voir les problèmes de conception clos, avec leur analyse ; • voir les problèmes de conception en cours d'analyse, avec une date de fin estimée ; • voir les dernières actions d'édition réalisées.	scénarios 1 et 2
Le système permet une navigation à travers le schéma d'architecture : • créer une exigence à partir d'un élément graphique du schéma ; • accéder facilement aux exigences à partir d'un élément graphique du schéma.	scénario 1 [21]
Le système permet de gérer facilement des liens entre exigences : • créer un lien à partir d'une une exigence en conservant le contexte de l'utilisateur ; • voir sans interaction le nombre d'exigences dérivées pour une exigence ; • atteindre à partir d'une exigence l'énoncé des exigences dérivées en une interaction au maximum .	scénario 3
Le système permet de naviguer facilement dans un ensemble d'exigences textuelles : • avoir une vue d'ensemble structurée des exigences ; • atteindre à partir d'une exigence l'ensemble des exigences dérivées ; • atteindre à partir d'une exigence dérivée l'exigence de plus haut niveau.	scénario 3 [21]
Le système permet à l'utilisateur de vérifier la cohérence d'exigences textuelles : • propose à l'utilisateur les termes conformes à un glossaire ; • aide l'utilisateur à trouver les exigences similaires sémantiquement en mettant en valeur les similitudes.	scénario 3

Figure 8: synthèse des exigences d'utilisabilité

solutions où l'utilisateur reste actif dans la détection et la correction des exigences, les solutions automatiques de correction et de vérification d'ontologie étant nombreuses mais pas nécessairement utilisables [1].

SYNTHESE DES EXIGENCES

Le Figure 8 est une synthèse des exigences d'utilisabilité identifiées dans les travaux antérieurs et les scénarios. Nous avons affiné ces exigences en exigences détaillées quand nous avions assez d'éléments pour le faire : comme dans tout processus d'ingénierie système, nos exigences seront affinées quand nous rentrerons dans la conception de solutions. Nous justifions la pertinence des exigences grâce à nos scénarios d'usage issus des enquêtes contextuelles réalisées.

CONCLUSION ET PERSPECTIVES

Dans cet article, nous avons élicité les exigences d'utilisabilité pour les outils d'ingénierie des exigences. A la différence des travaux existants : (1) nous nous sommes appuyés sur des enquêtes contextuelles de l'activité réelle des ingénieurs système (4 entreprises, 9 interviews retranscrites, des projets de nature différente) ; (2) nous avons formalisé ces exigences en respectant la norme ISO [13], ce qui les rend plus utilisables pour la communauté.

L'utilisabilité d'un outil concerne l'efficacité, l'efficience et la satisfaction de l'utilisateur dans un contexte d'utilisation spécifié [16]. Dans un environnement concurrentiel entre industriels, améliorer l'utilisabilité des outils d'ingénierie des exigences peut dans un premier temps permettre de gagner du temps sur la maintenance corrective et évolutive des systèmes. Dans un deuxième temps, cela peut contribuer à la maitrise de la complexité des systèmes dans le temps.

En termes de perspectives, nous prévoyons d'utiliser le travail réalisé sur les exigences pour concevoir des outils

d'ingénierie des exigences. Les concepts issus de la visualisation de grande quantité de données et des collecticiels seront particulièrement explorés, notre analyse de l'activité montrant que les activités d'ingénierie des exigences sont fortement collaboratives et génèrent des milliers d'exigences.

REMERCIEMENTS

Nous remercions les entreprises et les ingénieurs qui ont accepté d'être interviewés et observés dans leur environnement de travail.

BIBLIOGRAPHIE

1. Betty H. C. Cheng and Joanne M. Atlee.. Research Directions in Requirements Engineering. In Proc. *Future of Software Engineering (FOSE '07)*. IEEE Computer Society, (2007), pp 285-303.
2. Boehm B. "Some Future Trends and Implications for Systems and Software Engineering Processes." *Systems Engineering* 9, pp 1-19(2006)
3. Cooper, J.R.; Lee, Seok-Won; Gandhi, R.A.; Gotel, O., "Requirements Engineering Visualization: A Survey on the State-of-the-Art", In Proc. Of *Fourth International Workshop Requirements Engineering Visualization (REV*), (2009), pp.46-55.
4. DO-178B/ED-12B Software considerations in airborne systems and equipemnt certification, Requirements and Technical Concepts for Aviation , RTCA SC-167 / EUROCAE WG-12.
5. Gotel, O.; Mader, P., "How to Select a Requirements Management Tool: Initial Steps,", In Porc. Of *17th IEEE International Requirements Engineering Conference, RE '09*, IEEE (2009), pp.365-367.
6. Green,T.R.G.,Cognitive dimensions of notations, People & Computers V, 1989, Cambridge Univ. Press, 443-460.
7. Harris J. and Henderson A. 1999. A better mythology for system design. In *Proc. of the SIGCHI conference on Human Factors in Computing Systems* (CHI '99). ACM, New York, NY, USA, 88-95.
8. Hoffmann, M.; Kuhn, N.; Weber, M.; Bittner, M., "Requirements for requirements management tools, In *Proc. of the 12th IEEE International " Requirements Engineering Conference, 2004 »,* pp.301,308,
9. Hughes T.P., Rescuing Prometheus : four monumental projects that change the world, New York: Pantheon, 1998.
10. INCOSE, Requirements Management Tools Survey, http://www.incose.org/ProductsPubs/products/rmsurvey.aspx
11. ISO IEC 15288-2008. System engineering and software engineering – System life cycle processes. IEEE (2008).
12. ISO/IEC 26702 Standard for Systems Engineering - Application and Management of the Systems Engineering Process," (2007).
13. ISO/IEC/IEEE 29148:2011Systems and software engineering -- Life cycle processes --Requirements engineering", (2011).
14. ISO/IEC TR 24766:2009, Information Technology – Systems and Software Engineering – Guide for Requirements Engineering tool Capabilities, (2009).
15. ISO/IEC 9126, Information Technology – Software Product Evaluation – Quality Characteristics and guidelines for their use, (1991).
16. ISO 9241-11 Exigences ergonomiques pour travail de bureau avec terminaux à écrans de visualisation - partie 11 : Lignes directrices concernant l'utilisabilité, (1998).
17. Juan M. Carrillo de Gea, Joaquín Nicolás, José L. Fernández Alemán, Ambrosio Toval, Christof Ebert, Aurora Vizcaíno, Requirements engineering tools: Capabilities, survey and assessment, Information and Software Technology, Volume 54, (2012), pp 1142-1157
18. Karlson A., Benderson B. & Contreras-Vidal J. *Understanding one handed use of mobile devices*. In Handbook of Research on User Interface Design and Evaluation for Mobile Technology. IGI Global(2008), pp 86–101.
19. Klaus Pohl. 1994. The three dimensions of requirements engineering: a framework and its applications. *Inf. Syst.* 19, 3 (1994), pp 243-258.
20. Luzeaux D., Ruault R., Wippler J-L. Maitrise de l'ingénierie des systèmes complexes et des systèmes de systèmes, étude de cas. Hermès (2011).
21. Nan Niu; Reddivari, S.; Zhangji Chen, "Keeping requirements on track via visual analytics,In *Proc. Of 21st IEEE International " Requirements Engineering Conference (RE),* (2013), pp.205-214.
22. Sage, A.P., "Systems engineering education," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions* vol.30, no.2 (2000), pp.164-174.
23. Sire, S., Chatty, S., Gaspard-Boulinc, H., Colin, F.-R. How can groupware preserve our coordination skills? Designing for direct collaboration. In Proc. of Interact 99, IFIP (1999), pp. 304--312.
24. Thayer Richard H., Bailin Sidney C., and Dorfman M... *Software Requirements Engineerings, 2nd Edition* (2nd ed.). IEEE Computer Society Press (1997).