

**République Algérienne Démocratique et Populaire**  
**Ministère de l'Enseignement Supérieur et de la Recherche Scientifique**

**Université Cheikh Larbi Tébessa**  
**Faculté des Sciences de l'Ingénieur**  
**Département d'Informatique**

N° D'ORDRE : .....  
SERIE : .....

## **MEMOIRE**

Présenté en vue de l'obtention du diplôme de  
**Magister en Informatique**  
Option: Genie Logiciel (GL)

<p><b>LA VALIDATION DES BESOINS</b> <b>DANS LE PROCESSUS</b> <b>INGENIERIE DES BESOINS</b> <b>— APPLICATION AUX ENVIRONNEMENTS COOPERATIFS —</b></p>
--

**Présenté par :** Mme Maalem Dourdour Sourour Epouse

**Dirigé par :** Pr. Zarour Nacer Eddine

**SOUTENU LE 09/05/2010**

**DEVANT LE JURY COMPOSE DE:**

<b>Président:</b>	Dr. Boufaïda Mahmoud	Professeur	Université de Constantine
<b>Rapporteur:</b>	Dr. Zarour Nacer Eddine	Professeur	Université de Constantine
<b>Examineurs:</b>	Dr Benmohammed mohamed	Professeur	Université de Constantine
	Dr Maamri Ramdane	Maitre de conférences A	Université de Constantine

## REMERCIEMENT

*« Après avoir remercié ALLAH le tout puissant »*

*Je tiens avant tout, à remercier vivement le professeur Mr N. ZAROUR, pour avoir accepté de m'encadrer, afin de mener ce travail à terme. Qu'il trouve ici, l'expression de ma profonde et sincère reconnaissance, pour tous les efforts, sa rigueur, ses critiques constructives, son savoir et son honnêteté, des qualités dont je ne suis pas cessé de m'inspirer pendant toutes les phases de mon travail et le reste de ma vie.*

*Mes remerciements vont également à Mr M. BOUFAIDA, professeur au département d'informatique à l'université de Constantine et responsable de l'option Génie Logiciel Du pôle l'est de l'école doctorale informatique, que sans lui je n'aurai certainement pas réussi mon année théorique pour en arrivé là aujourd'hui, et pour avoir accepté de présider le jury.*

*Mes sincères remerciements vont à Mr M. BENMOUHAMED professeur au département d'informatique à l'université de Constantine et Mr R. MAAMRI Maître de conférence au département d'informatique à l'université de Constantine, pour m'avoir fait l'honneur de participer à ce jury.*

*Je tiens aussi à remercier chaleureusement Mr L. SELMI, pour son aide et sa grande contribution à la rédaction.*

*Il m'est agréable d'exprimer mes remerciements à tous ceux qui se sont mobilisés en me portant leurs soutiens pendant toutes les phases de la réalisation de ce travail. M. DERDOUR, Firas Tamime, ma famille, ma belle famille et mon Directeur T. Boukhalout.*

## *Dédicaces*

*A Celui Qui M'a Indiqué La bonne Voie En Me Rappelant Que La Volonté Fait Toujours Les Grands...*

***Mon père***

*A Celle Qui A Attendu Avec Patience Les Fruits De Sa Bonne Education...*

***Ma Mère***

*A Celui Dont Son Courage A Nourri Ma Persévérance.*

***Mon Epoux***

*A Mes Petits Anges firas et tamime dont la Présence Ma Ravitailler psychiquement et spirituellement*

***Mes Fils***

*A mes sœurs Kîma Raavda, Aziza, Samiha & Mes frères*

*A Rahîma, Hamed et toute ma belle famille*

*D'avoir été présent, passion et adorons. C'est pourquoi, je vous dédie cette thèse.*

# Sommaire

<b>Introduction Générale</b>		02
<b>Chapitre I : Ingénierie des Exigences (IE)</b>		
	Résumé.....	06
<b>I.</b>	Introduction & Problématique.....	07
<b>II.</b>	Définitions.....	08
	1. Exigence.....	08
	1.1 Besoin & Exigence.....	09
	1.2 Les propriétés des exigences.....	09
	1.3 Distinguer Plusieurs sortes d'exigences.....	10
	1.3.1 Exigences produit & exigences de processus.....	10
	1.3.2 Classification de D. Amyot.....	11
	1.3.3 Classification de Sommerville.....	11
	1.3.4 Classification fonctionnelle/non fonctionnelle.....	11
	2. Les Parties Prenantes.....	12
	3. Document des Besoins.....	13
	4. Ingénierie des exigences.....	14
	4.1 Définitions.....	14
	4.2 Ingénierie des exigences Logiciel/Systèmes.....	15
	4.3 Les Mondes de L'ingénierie des exigences.....	15
<b>III.</b>	Les Disciplines qui constituent les fondements d'IE.....	16
	1. La psychologie cognitive.....	16
	2. Anthropologie.....	16
	3. Sociologie.....	16
	4. La linguistique.....	17
<b>IV.</b>	Processus d'ingénierie des exigences.....	17
	1. Définition.....	17
	2. Quelques approches du processus IE.....	18
	3. Les activités du processus IE.....	20
	3.1 Elicitation des exigences.....	20
	3.2 Analyse des exigences.....	21
	3.3 Spécification des exigences.....	21
	3.4 Négociation des exigences.....	22
	3.5 Validation des exigences.....	23
	4. La documentation des exigences.....	24
<b>V.</b>	Les Techniques utilisées En IE.....	25
	1. Le brainstorming.....	25
	2. Enquête contextuelle.....	25
	3. Réunion.....	25
	4. Interviews.....	26
	5. Observation.....	26
	6. Prototypage.....	26
	7. Les patrons d'IE.....	26
	8. La Méthode QFD.....	26
<b>VI.</b>	La Gestion des Exigences.....	29
<b>VII.</b>	Outils d'Ingénierie des Exigences.....	30
<b>VIII.</b>	Coopération & Ingénierie des exigences.....	31
	Conclusion.....	33

## Chapitre II : La Validation des Exigences

	Résumé.....	35
<b>I.</b>	Introduction .....	36
<b>II.</b>	Définitions .....	37
	1. La Validation & La Vérification .....	37
	2. La Validation & La vérification des exigences.....	38
	2.1 Vérification des exigences.....	38
	2.2 Validation des exigences.....	38
	2.3 Vérification des systèmes.....	38
	2.4 Validation des systèmes.....	38
	3. Le processus de validation.....	39
	3.1 Phase 1 : La phase de pré validation.....	39
	3.2 Phase 2 : La phase de validation du procédé.....	39
	3.3 Phase 3 : La phase de maintien de la validation.....	39
	4. Le processus de validation des exigences.....	39
	5. La nécessité de la validation des exigences.....	40
<b>III.</b>	Problématique de la validation des exigences.....	40
<b>IV.</b>	Critères de qualité à valider.....	41
<b>V.</b>	Les techniques de validation des exigences.....	42
	1. Pré-révision des exigences.....	42
	2. La révision des exigences.....	43
	2.1 Inspection des exigences.....	46
	2.1.1 Inspection Dirigée par les cas de Test.....	46
	2.1.2 La Méthode d'inspection N-FOLD.....	47
	2.1.3 La Matrice de tracabilité.....	47
	2.2 Les techniques de lectures.....	47
	2.2.1 Technique de lecture basé Ad-hoc.....	48
	2.2.2 Technique de lecture basée check-list.....	48
	2.2.3 Technique de lecture basée perspective.....	48
	3. Le Prototypage d'exigences.....	49
	3.1 Prototype rapide.....	50
	3.2 Prototype évolutif.....	50
	4. Validation des exigences basée-Modèle.....	50
	4.1 Modèle flux de données.....	50
	4.2 Modèle Compositionnel.....	51
	4.3 Modèle de Classification.....	51
	4.4 Modèle Stimulus réponse.....	51
	4.5 Modèle de processus.....	51
	4.6 Modèle de simulation.....	51
	5. La validation des exigences basée-Test.....	52
	6. La validation des exigences orientée-point de vue.....	53
<b>VI.</b>	Approche Pour la validation et Vérification des exigences.....	55
	1. Preuve mathématique et logique pour la vérification et la validation des exigences.....	55
	2. Validation des exigences par traitement de langage naturel.....	55
	3. Une méthode automatisée avec XML pour la Validation des exigences .....	56
	4. Validation des exigences d'une spécification en UML.....	57
	5. Une approche empirique pour la Validation des exigences.....	57
	6. L'Approche systèmes expert pour la validation des exigences.....	58
	7. Les méthodes formelles pour la vérification et la validation.....	59
	8. Synthèse des approches.....	60
<b>VII.</b>	La validation des exigences dans les systèmes coopératifs.....	60
	Conclusion.....	63

## Chapitre III : Les systèmes d'information coopératifs

	Résumé.....	65
<b>I.</b>	Introduction .....	66
<b>II.</b>	Principale Définition .....	67
	1. Communication.....	67
	2. Coordination.....	67
	3. Coopération.....	68
	4. Collaboration.....	69
<b>III.</b>	Le travail Coopératif.....	70
	1. Définition.....	70
	2. Les Méthodologie de management.....	71
	2.1 Gestion des processus.....	71
	2.2 Gestion de la performance.....	72
	2.3 Gestion des Connaissances.....	72
	3. Les techniques et outils logiciels support du travail coopératif.....	73
	3.1 CSCW.....	75
	3.2 Le groupeware.....	75
	3.3 Le Workflow.....	76
<b>IV.</b>	Systèmes D'information Coopératifs.....	79
	1. Définition.....	79
	2. Typologie des systèmes d'information coopératifs.....	79
	2.1 L'approche fédérée.....	79
	2.2 Les approches à base de médiation.....	79
	3. Propriétés des SIC.....	80
	3.1 L'autonomie.....	80
	3.2 La Distribution.....	80
	3.3 L'hétérogénéité.....	81
	4. Types de connaissances dans les SICs.....	81
	4.1 Entreprise Etendue.....	82
	4.2 Conception centrée sur l'utilisateur.....	82
	4.3 Collecticiel.....	82
	4.4 Outils d'aide à la Décision de Groupe.....	82
	4.5 Interfaces Adaptatives Intelligentes.....	83
	5. Types (Formes) des systèmes d'information coopératifs .....	83
	6. Ingénierie des systèmes d'information coopératifs.....	84
	Conclusion.....	85

## Chapitre IV : Vers une Méthodologie de Validation des Exigences Collaborative dans Les Organisations Distribuées –VECOD–

	Résumé.....	87
<b>I.</b>	Introduction .....	88
<b>II.</b>	Motivation .....	88
<b>III.</b>	Présentation de la solution .....	89
	1. Les Problèmes liés à l'ambiguïté du terme dans la littérature .....	89
	1.1 Le Rapport entre les activités de VV par rapport aux propriétés à contrôler .....	90
	1.2 Définition.....	90
	1.3 Propriétés émergentes du travail coopératif.....	91
	2. Les Problèmes liés à la validation dans le processus d'IE .....	91
	3. Les Problèmes liés à la Plate forme coopérative.....	94
<b>IV.</b>	Un Meta Modèle pour la validation et la vérification des exigences.....	96
<b>V.</b>	Présentation des différentes Phases du processus de la validation collaborative.....	97
	1. Phase d'organisation.....	100

1.1	La Planification.....	100
1.2	La Qualification.....	100
1.3	L'inspection.....	102
2.	Phase de vérification des exigences.....	102
2.1	Processus de vérification des critères non sémantiques.....	103
2.2	Processus de validation des critères d'aspect sémantiques.....	104
2.3	Processus de validation des critères non fonctionnels.....	106
2.4	Processus de génération de la QFD (Quality Function Deployment).....	107
3.	Phase de validation collaborative des besoins.....	109
3.1	Evaluation des conflits.....	110
3.2	Négociation des conflits.....	110
3.3	Unification des vues.....	111
3.4	Génération de prototype.....	113
<b>VI.</b>	Travaux voisins et positionnement par rapport à quelques travaux d'actualité.....	115
	Conclusion et Perspective.....	117

## Chapitre V : Déploiement et étude de cas

	Résumé.....	119
<b>I.</b>	Introduction.....	120
<b>II.</b>	Déploiement Des Différentes Phases De VECOD.....	121
1.	Phase d'organisation.....	122
1.1	La Planification.....	122
1.2	La Qualification.....	122
1.3	L'inspection.....	123
2.	Phase de vérification des exigences.....	124
2.1	Processus de génération de la QFD (Quality Function Deployment).....	124
3.	Phase de validation collaborative des besoins.....	126
3.1	Détection des conflits.....	126
3.2	Gestion des conflits.....	126
3.3	Fusion des QFDs.....	126
3.4	Génération de prototype.....	127
<b>III.</b>	Description De L'étude de cas : Commerce Electronique.....	128
1.	Déroulement de l'exécution.....	129
1.1	L'organisation 1 : Spécialiste en Application Site Web et sécurité.....	129
1.2	L'organisation 2 : Spécialiste en Commerce Classique.....	129
1.3	Fusion des Matrices et gestion des conflits.....	130
2.	Génération de prototype.....	132
3.	Evaluation de la coopération.....	132
	Conclusion.....	133

## Conclusion Générale & Perspectives

135

## Références bibliographiques

139

# Liste des figures

Fig.1. Dimensions du concept « Exigence ».....	08
Fig.2. Enjeux de l'ingénierie des exigences.....	09
Fig.3. Classification des exigences non fonctionnelle.....	12
Fig.4. Rôle des parties prenantes dans le processus IE.....	13
Fig.5. Les entrées /Sorties du processus d'IE.....	14
Fig.6. Les Mondes de l'ingénierie des exigences.....	15
Fig.7. Les trois dimensions de l'ingénierie des exigences.....	18
Fig.8. Processus d'ingénierie des exigences de Loucopoulos.....	18
Fig.9. Processus d'ingénierie des exigences de Kotonya.....	19
Fig.10. Processus d'ingénierie des exigences de Larry Boldt.....	19
Fig.11. Processus d'ingénierie des exigences Modèle d'activité Coarse-Grain.....	20
Fig.12. Processus d'ingénierie des exigences de Ian-Sommerville.....	20
Fig.13. Processus de spécification.....	22
Fig.14. Principe de la méthode QFD.....	27
Fig.15. Activités principale de la Gestion des exigences.....	30
Fig.16. Processus de validation des exigences .....	39
Fig.17. Le monde et la machine.....	40
Fig.18. Contrôle de pré-révision.....	43
Fig.19. Processus de révision des exigences.....	44
Fig.20. Révision Active.....	45
Fig.21. Inspection de Fagan.....	45
Fig.22. Approche de Prototypage.....	49

Fig.23. Décomposition Fonctionnelle du DFD.....	51
Fig.24. Points de vue sur un problème.....	54
Fig.25. Communication, Coordination, Collaboration, Coopération.....	69
Fig.26. Elément clés du travail coopératif.....	70
Fig.27. Transformation de la connaissance.....	73
Fig.28. Intégration des coopérations.....	74
Fig.29. Taxonomie des outils de TACO.....	76
Fig.30. Les trois dimensions de travail coopératif.....	78
Fig.31. Taxonomie des exigences.....	90
Fig.32. Lien des propriétés par rapport aux VV.....	90
Fig.33. Processus de validation en Ingénierie des Exigences.....	92
Fig.34. Diagramme cas d'utilisation de la démarche.....	93
Fig.35. Processus coopérative.....	95
Fig.36. Meta-Modèle pour la validation & la vérification.....	97
Fig.37. Diagramme d'activité du fonctionnement de la méthode.....	99
Fig.38. Processus de Vérification de quatre critères non sémantiques.....	103
Fig.39. Processus de validation de quatre critères sémantiques.....	105
Fig.40. Processus de validation des critères non fonctionnels.....	106
Fig.41. Exemple de QFD.....	109
Fig.42. Points des conflits par rapport à l'expertise.....	110
Fig.43. Distances sémantiques et distances articulatoires.....	114
Fig.44. Interface De NetMeeting.....	121
Fig.45. Interface Emettre un appel.....	122
Fig.46. Fiche Technique de l'exigence, Check-list, Grille d'évaluation.....	123
Fig.47. La Diffusion des documents.....	123
Fig.48. Modèle QFD Excel.....	125
Fig.49. Partagé des Application.....	125

<b>Fig.50. Interface Tableau Blanc de NetMeeting.....</b>	<b>127</b>
<b>Fig.51. QFD de la partie concernant le site web et sécurité.....</b>	<b>129</b>
<b>Fig.52. QFD de la partie concernant le commerce.....</b>	<b>130</b>
<b>Fig.53. QFD Générale du Problème.....</b>	<b>131</b>
<b>Fig.54. Exemple de Création de Facture en Collaboration.....</b>	<b>132</b>
<b>Fig.55. Taux de Complémentarité, 'Exemple Commerce électronique'.....</b>	<b>132</b>



# Liste des tableaux

---

TABLE I.	AVANT ET APRES L'UTILISATION DE LA QFD.....	28
TABLE II.	PROPRIETE DES EXIGENCES A VERIFIER ET VALIDE.....	41
TABLE III.	QUESTION DE CHECK-LIST .....	48
TABLE IV.	SUNTHESE DES APPROCHES DE VERIFICATION & VALIDATION.....	60
TABLE V.	ROLES ET COMPETENCES DES PARTIES PRENANTES.....	94
TABLE VI.	EXEMPLE DE CONFLIT DANS LA QFD.....	112
TABLE VII.	EXEMPLE DE GESTION DE CONFLIT.....	112
TABLE VIII.	COMPARAISON ET POSITIONNEMENT DE VECOD.....	115

---

*INTRODUCTION*  
*GÉNÉRALE*

---

# Introduction Générale

---

La complexité croissante des systèmes et l'interdisciplinarité nécessaire à leur réalisation poussent les organisations à recourir aux équipes, car elle semble constituer un moyen pour améliorer la qualité et le temps de réalisation des produits industriels où plusieurs organisations unissent leurs compétences et ressources pour répondre à une opportunité qu'elles n'auraient pu prendre en charge seule. Ce qui nécessite la présence de plusieurs domaines d'expertises différents faisant intervenir la coopération de plusieurs organisations supportées par un Système d'Information Coopératif SIC [118]. Le plus grand nombre de travaux dans le domaine des SIC est consacré aux deux processus de conception et d'implémentation justifié par le grand nombre d'outils logiciels qui les supportent, mais encore très peu d'attention au processus amont d'ingénierie des exigences (IE).

De nombreux développements informatiques échouent à cause d'une mauvaise Ingénierie des Exigences IE et particulièrement pour les SIC. Il est bien connu que ces erreurs peuvent entraîner des coûts de développement élevés, des retards dans la livraison de produits et de perte de réputation. Fairley dans [152] présente les cas dans lesquels il a été 5 fois plus coûteux de corriger une faute lors de la phase de conception que pendant la phase initiale d'analyse des besoins, 10 fois plus coûteux au cours de codage, de 20 à 50 fois plus coûteux pendant le test d'acceptation et de 100 à 200 fois plus coûteux de corriger les problèmes qui ont surgi au cours de la mise en service.

Le processus d'IE comprend généralement trois grandes phases importantes, l'élicitation, la spécification et la validation [155]. Afin d'assurer une meilleure prise en charge des besoins utilisateurs \ exigences systèmes ; Les besoins / exigences doivent être de qualité, la garantie de cette qualité est assurée par les étapes de validation et de vérification. Ces activités se déroulent tout au long du cycle de vie, lors de l'approbation des documents intermédiaires et pendant des phases réservées.

La plus part des méthodes ou pratiques existantes vise seulement à identifier et recueillir les besoins. Les activités de validation des besoins par rapport aux clients s'intègrent naturellement dans le processus d'après [156]. José et all dans [155] montrent que seules quelques approches, fournissent des techniques pour la validation des exigences. La plupart d'entre elles définie seulement des lignes directrices sur la façon dont les développeurs et les clients devront revoir la spécification des exigences en vue de trouver des inconsistances et des erreurs, et pour terminer une étude comparative entre les méthodologies de développement Web et leurs prises en charge des activités du processus d'IE, ils mentionnent que seulement 4/10 des méthodes ont pris en considération la validation et cela par mention de technique sans aucune démarche ou méthodologie. Lulu He & all [79] aperçoit que la validation des exigences n'est

souvent pas suffisamment couverte non seulement dans le monde pratique mais de même dans le monde académique. Tant dit que Siew et al [39] considèrent que la plupart des livres la présentent comme une liste de "bonnes pratiques" et présente la validation des exigences autant qu'un processus hétérogène basée sur l'application d'une grande variété de techniques indépendantes. À leurs tours Nuseibeh B et al dans [78] confrontent le problème de la validation des exigences avec le problème de la validation des connaissances scientifiques.

Le travail que nous présentons dans ce mémoire est consacré à une activité aussi importante que délicate de l'ingénierie des exigences qui est *la validation* qui vient compléter le travail de Amroune [153] qui s'est intéressé à l'activité d'élicitation des besoins, où il a proposé une approche cognitive centrée utilisateurs dans les organisations distribuées. Le résultat de cette activité est un catalogue de besoins écrit en langage naturel que nous utilisons comme entrée à notre activité et celui de Yagoub [154] qui s'est intéressé à la spécification où il a fourni d'une part, une démarche coopérative centrée sur l'utilisation des outils groupware pour réaliser l'activité de spécification et d'autre part, elle est basée sur une orientation multi présentation dans la modélisation des besoins suivant un processus itératif et incrémental dont le résultat est un ensemble de spécifications du problème qui passe droit à la validation.

La difficulté de la validation des exigences se focalise d'après [78] dans ces raisons :

- La première raison est de nature philosophique, et concerne la question de ce qui est réel et ce qui est habituel, connu, personnel...
- La deuxième raison est sociale, et concerne la difficulté de parvenir à un accord entre les différentes parties prenantes.

Comme les systèmes ne sont pas simples et lors de la fusion des domaines (SIC, IE, validation) les problèmes suivants sont apparus:

- Des problèmes liés à la plate forme coopérative : Univers ouverts, distribués, hétérogènes, autonomie locale vs cohérence globale, organisation sociale, communicabilité, des situations conflictuelles variées dues aux grands nombres de sources d'informations, des différences d'horaires (décalage horaire) entre les différents acteurs, Différences de culture entre acteurs (langue), Diversité culturelle professionnelle (ou de métier) dans le cadre d'une coopération entre les organisations de métiers différents ou possédant des cultures d'organisation différentes (artefact, mode de management, moyens et technologie, modèle organisationnel...), Le choix des outils adéquats de communication à distance, Les problèmes de motivation, d'engagement et de confiance entre acteurs, etc.
- Des problèmes liés à la validation durant le cycle de vie logiciel: la nature des informations ?, le quoi ?, le comment ?, le quand ?, le où ?, le qui ?, avec quelle technique et par quel moyen ?, la durée ?, la position de l'activité de vérification & validation par rapport au cycle de vie logiciel, etc.

- Des problèmes liés à la validation durant le processus d'ingénierie des exigences : est ce qu'une activité ou phase ?, quel est le résultat de la validation ?, Comment pouvons-nous valider les exigences?, Quels types de processus de validation et de vérification est le mieux approprié pour un projet?, Comment s'assurer que la solution répond aux besoins des parties prenantes et de l'entreprise?, quelle est la meilleure technique à utiliser pour valider ? . Comment mettre d'accord toutes les parties prenantes ?. Les différents modes de validation (formel, semi formel, informel), la vérification au niveau modèle où exigences, la validation des besoins fonctionnels & non fonctionnels , la maîtrise de l'évolution des exigences, l'insuffisance dans les techniques de négociation pour la validation, l'absence de l'activité de V & V dans certaines méthodes d'IE, l'absence de méthodes de validation, le manque des analystes experts, le manque d'experts métiers avec un niveau d'analyse et de communication élevé et des utilisateurs expérimentés, etc.

- Des problèmes liés à l'ambiguïté du terme validation: confusion entre validation & Vérification, Validation & Test, Validation & Crédibilité, Validation & Acceptation, etc. manque de standard, normes et consensus, etc., insuffisance dans les patrons d'analyse, les ontologies de domaine, etc.

Afin de répondre à ces insuffisances. Nous proposons VECOD une méthodologie de validation des exigences collaborative dans les organisations distribuées. La proposition est basée point de vue selon les compétences cognitives des parties prenantes en guidant leur collaboration tout au long du processus de validation.

Pour cela nous proposons ce qui suit :

- Une définition de la validation et de la vérification des exigences, prenant en compte la séparation des aspects selon leurs aptitudes sémantiques, quelques critères de qualité à vérifier et à valider, enrichis par d'autres critères émergents du travail coopératif distribué.

- L'utilisation des outils groupware (collecticiels), pour assurer le travail coopératif, selon une dimension spatio-temporelle, la plus adéquate par rapport à la fonctionnalité assurée par chaque étape d'IE, avec quelque recommandation de leurs adaptations avec notre démarche.

- Proposer une démarche plus générale que celle trouvée dans la littérature, et qui traite la validation en tant qu'activité et non en tant que phase, qui présente la validation comme un processus continu, incrémentale, collaboratif, qui se déroule tout au long du processus IE, basé point de vue.

- Déterminer, de façon détaillée, les rôles et les compétences des parties prenantes, les résultats de chaque sous activité du processus de validation, et les moyens permettant sa mise en œuvre.

Afin de réaliser les objectifs cités précédemment, nous avons retenu pour ce mémoire une organisation en cinq chapitres.

- Le 1<sup>er</sup> chapitre évoque le domaine de L'IE, les différentes définitions des vocabulaires du sujet, Exigences, Besoins,...les disciplines qui constituent les fondements pour une IE effective suivie d'un aperçu des recherches en cours, présentées en fonction de ces principales activités. Nous présenterons les techniques qui peuvent être utilisées durant les différentes phases du processus, nous parlerons de la gestion des exigences et quelques outils d'une manière succincte. Et nous ferons le lien entre « IE » et « système coopératif » à travers le projet CREWS.

- le 2<sup>ème</sup> Chapitre, Concerne la validation en ingénierie des exigences, nous présenterons les différentes définitions des vocabulaires, ensuite nous décrirons les problématiques de la validation des exigences selon le monde de la machine suivie d'un aperçu sur quelques critères de qualité à vérifier et à valider dans cette activité. Les techniques qui peuvent être utilisées durant le processus de validation des exigences, les recherches en cours à travers les approches les plus prometteuses, commentée chacune, pour décrire d'une manière succincte la validation des exigences et les systèmes coopératifs.

- Le 3<sup>ème</sup> chapitre présente un aperçu sur les SICs. Il fait le point sur la définition des différents vocabulaires de ce domaine, en suit il s'agira de traiter la mission de travail coopératif, ses éléments, les techniques et les outils logiciels qui le supportent, enfin on terminera par décrire les SIC en terme de finalité et Propriétés, typologie et topologie.

- Le 4<sup>ème</sup> chapitre présente notre méthodologie VECOD, une méthodologie de validation des exigences collaborative dans les organisations distribuées. La proposition est basée Point de vue selon les compétences cognitives des parties prenantes en guidant leur collaboration tout au long du processus. Ce processus est divisé en plusieurs activités, afin de supporter la plate forme coopérative tout en précisant les rôles et les résultats ainsi que les outils de chaque sous processus.

- Le 5<sup>ème</sup> chapitre présente une concrétisation de la méthodologie VECOD, à travers son déploiement pas à pas par un groupware, NetMeeting de Microsoft, présentant pour cela une explication détaillée sur ces options qui permettent d'assurer le travail collaboratif, et spécialement son tableau blanc interactive virtuel. Il est suivi d'une étude de cas d'une application de commerce électronique dans laquelle plusieurs expertises doivent collaborer afin de valider son cahier de charge.

---

*CHAPITRE 1 :*

*L'Ingénierie des Exigences (IE)*

---

# Chapitre I :

## L'Ingénierie des Exigences (IE)

---

---

### Résumé

---

*Les exigences sont des entités à part entière au cœur du processus de développement des systèmes. Elles formalisent l'expression des besoins des demandeurs et les engagements des parties prenantes. Ces exigences peuvent être des contraintes techniques, fonctionnelle ou non fonctionnelle.*

*L'ingénierie des exigences (IE) qui est un ensemble d'activités relatives à l'identification et la communication, agit comme un pont entre le monde réel et les besoins des utilisateurs, des clients et des autres parties prenantes. Elle est multidisciplinaire (psychologie cognitive, Anthropologie, Sociologie, linguistique,...), qui fait participer plusieurs intervenants (analyste, Expert du domaine, client...), utilisant plusieurs techniques telles que (Le brainstorming, Enquête contextuelle, Réunion, Interviews, Observation...), qui peuvent être utilisées durant les différentes phases du processus d'IE.*

*De manière générale, le processus d'ingénierie des exigences inclut les phases d'Elicitation, d'Analyse, de Spécification, de Validation et de Gestion des exigences. L'Elicitation des exigences consiste en la collecte, la capture, la découverte et le développement des exigences à partir d'une variété de sources y compris les parties prenantes humaines. L'analyse se focalise sur l'examen, la compréhension et la modélisation des exigences élicitées et leur vérification La spécification est l'enregistrement et la documentation des exigences de sortes qu'elles soient utilisables par les parties prenantes et en particulier les développeurs qui doivent concevoir et construire le système. La validation est la confirmation de la qualité des exigences et de leur conformité aux besoins et désirs des parties prenantes. Quant à la gestion, elle est exécutée tout au long du processus d'ingénierie des exigences. Elle inclut les activités de planification, de contrôle de version et de traçabilité des exigences.*

---

## INTRODUCTION & PROBLEMATIQUE

L'ingénierie des exigences résulte de la tentative, déjà ancienne, d'imaginer une méthode qui permettrait aux commanditaires d'un projet de formaliser sans ambiguïté l'expression de leurs besoins, tant fonctionnels que techniques, afin d'en garantir la bonne compréhension par les équipes en charge de les concrétiser en un système d'information.

On comprend les raisons d'une telle tentative. Outre une clarification de la relation maîtrise d'ouvrage – maîtrise d'œuvre qui identifie les rôles et responsabilités de chacun, elle permet d'approcher un idéal, celui de réaliser des systèmes d'information qui répondent effectivement à des objectifs « métiers » clairs et bien compris.

Une urgence, quand on sait que le dernier rapport Chaos du Standish Group nous apprend qu'un seul projet logiciel sur trois est une réussite et qu'environ 70 % d'entre eux ne répondent pas aux spécifications initiales, budgets, délais ou impératifs de qualité... un taux d'échec en augmentation ces dernières années !

La formalisation des attentes par l'Ingénierie des exigences vise :

- **La recherche des bénéfices « métier » :** La nécessité de justifier les exigences exprimées impose une réflexion approfondie sur les bénéfices espérés tant du système lui-même que de chacune de ces exigences. L'ingénierie des exigences contribue à l'argumentation des décisions et à la justification des investissements. Une telle approche facilite aussi l'estimation de la « rentabilité » d'une exigence et d'un système : ce qu'on en attend comme bénéfice justifie-t-il son coût ?
- **L'exhaustivité de l'expression des besoins:** En pilotant la réflexion de la maîtrise d'ouvrage, un processus d'ingénierie des exigences contraint à se poser l'ensemble des questions utiles, à envisager tous les volets du fonctionnement d'un système, contribuant ainsi à réduire le risque d'en oublier des aspects essentiels.
- **La traçabilité des exigences :** Le lien entre une exigence et sa réalisation (« l'agent ») est clairement identifié par une démarche d'ingénierie des exigences, facilitant la maintenance ultérieure du système, tant corrective qu'évolutive. Toutefois, l'ingénierie des exigences vise avant tout à l'expression des besoins plus qu'à leur gestion. Les outils de « gestion des exigences » sont plus habituels. Ils interviennent ensuite pour suivre cette traçabilité, veiller à la réalisation des exigences exprimées et à leur conservation (la « mémoire des attentes ») ;
- **L'évaluation de la satisfaction :** Les exigences clairement exprimées préparent à la recette du système. La satisfaction des exigences est un critère de qualité ;
- **La rationalisation plus complète du processus de développement des systèmes d'information**  
La formalisation des exigences se place en amont de la modélisation des solutions. C'est donc un

nouvel outil qui « modélise », donc formalise les problèmes auxquels doit répondre le futur système d'information en amont des modèles employés jusqu'alors, en utilisant un formalisme plus accessible aux commanditaires. C'est donc un nouveau maillon dans l'objectivation, voire l'automatisation de la conception d'un système

Après avoir présenté la problématique et l'objectif du domaine d'IE; nous les présenterons les différentes définitions des vocabulaires du sujet, dans la 2eme section, nous décrirons ensuite les disciplines qui constituent les fondements pour une IE effective dans la 3eme section avant de donner en section 4 un aperçu des recherches en cours, présentées en fonction de ces principales activités. En section 5, nous présenterons les techniques qui peuvent être utilisées durant les différentes phases du processus. En section 6, nous parlerons de la gestion des exigences pour décrire ensuite en section 7 et ce d'une manière succincte quelques outils. Enfin en section 8 nous ferons le lien entre « IE » et « système coopérative » à travers le projet crews.

## DÉFINITION

### 1. Exigence (requirement en anglais)

Selon AIRBUS<sup>1</sup> les dimensions du concept « Exigence » ce présente tel que Fig.1

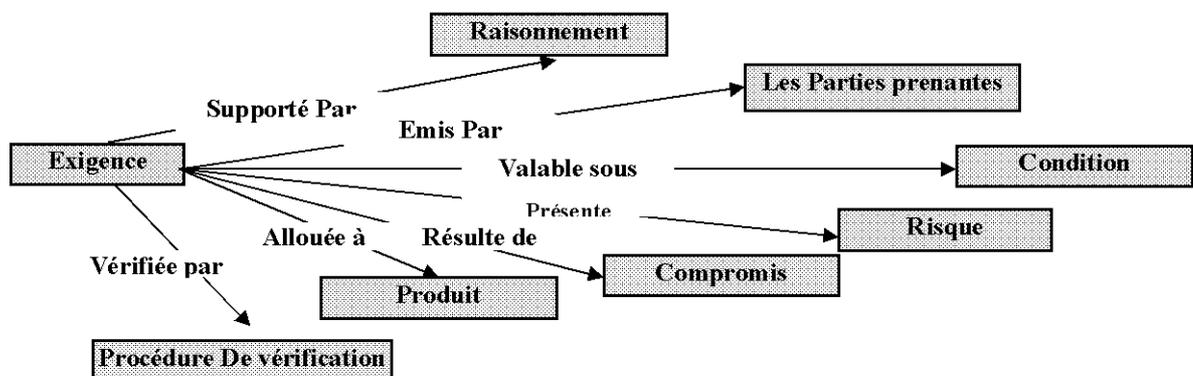


Fig.56. Dimensions du concept « Exigence »

Selon la norme IEEE 830-1993, Une exigence est définie comme :

- Une condition ou capacité dont un utilisateur a besoin pour résoudre un problème ou atteindre un objectif;
- Une condition ou capacité qui doit être satisfaite ou possédée par un système ... pour satisfaire un contrat, une norme, une spécification, ou tout autre document formellement imposé

<sup>1</sup> [http://conferences.esa.int/02C24/2\\_dechazelles\\_fraboulet.pdf](http://conferences.esa.int/02C24/2_dechazelles_fraboulet.pdf)

- Un énoncé à propos du système à concevoir avec lequel tous les intervenants impliqués sont d'accord et qui doit devenir vrai afin de résoudre adéquatement le problème du client (information claire et concise, concerne le système, toutes les parties prenantes ont confirmé sa validité, aide à résoudre le problème)

D'après l'encyclopédie libre, en ingénierie, les exigences sont l'expression d'un besoin documenté sur ce qu'un produit ou un service particulier devrait être ou faire. Elles sont le plus souvent utilisées dans un sens formel dans l'ingénierie des systèmes et dans l'ingénierie logicielle et plus particulièrement dans les procédures d'appel d'offres publiques et privées.

Dans l'approche classique de l'ingénierie, les exigences sont considérées comme des pré-requis pour les étapes de conception et de développement d'un produit.

### 1.1 Besoin & Exigence

Sibertin [6] fait la distinction entre les deux termes *Besoin (need)* « Caractéristiques, aptitudes que l'environnement attend du système » *Exigence (requirement)* « propriétés que le système doit satisfaire ». De même Essame [7] *Figure 2* montrent que les besoins et les exigences sont deux concepts différents et très importants dans l'ingénierie des exigences. Alors que les exigences représentent la vision du système du point de vue des concepteurs (point de vue technique), les besoins quant à eux représentent la vision du système uniquement d'un point de vue utilisateur.

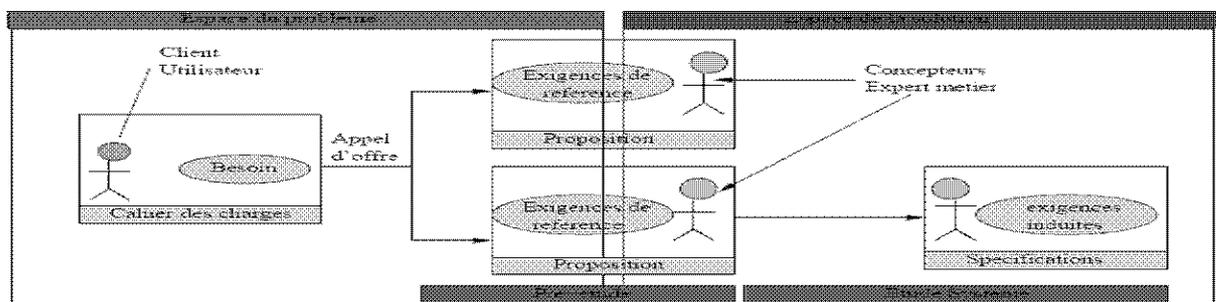


Fig.57. Enjeux de l'ingénierie des exigences. [7]

### 1.2 Les propriétés des exigences

Une Exigence/Besoin (E/B) est caractérisée par :

- Un nom
- Un n° de version : la version du système dans laquelle le B/E sera réalisé
- Une charge estimée (en jours/homme)
- Une origine : la source (personne) qui a identifié l'B/E en question. Les besoins identifiés par les utilisateurs sont les plus importants.

- Un niveau de priorité : quelle est la valeur ajoutée du B/E. Est-il indispensable (le système n'a pas de sens sans lui), majeur (il sera à réaliser, mais peut-être pas tout de suite), mineur (c'est un plus)
- Un niveau de risque : probabilité que la réalisation de cette exigence dérape au niveau des coûts, soit non faisable, etc. Sont 3 niveaux : élevé ( $p > 50\%$ ), moyen ( $10 \geq p \leq 50$ ) et faible ( $p < 10\%$ ).
- Un niveau de stabilité : évaluation de la probabilité que la description du E change. Deux niveaux haut (vrai quand le domaine n'est pas encore bien connu, ou quand la description n'est pas formalisée) ou bas

### 1.3 Distinguer plusieurs sortes d'exigences

Dans l'ingénierie système, une exigence peut être la description de ce qu'un système doit faire. Ce type d'exigence spécifie quelque chose que le système livré doit être capable de faire. Un autre type d'exigence spécifie quelque chose sur le système lui-même, et de quelle manière il exécute ses fonctions (exigences non fonctionnelles). De telles exigences s'appellent souvent «exigence non fonctionnelles», «exigences de performance» ou «qualité des exigences de service». Exemples de ce type d'exigences : la disponibilité, la testabilité, la facilité de maintenance et la facilité d'utilisation.

Un ensemble d'exigences définit les caractéristiques ou propriétés du système désiré (exigé). Une «bonne» liste d'exigences évite de spécifier la manière pour le système de mettre en œuvre ces exigences, laissant ce genre de décision pour les activités de conception. Un élément parmi les exigences qui décrit comment mettre en œuvre le système s'appelle un biais de mise en œuvre.

#### 1.3.1 Exigences produit & exigences de processus

Les projets sont soumis à trois sortes d'exigences :

- Les **exigences métier** qui décrivent le *quoi* dans les termes du métier. Elles décrivent ce qui doit être fourni ou réalisé pour produire de la valeur.
- Les **exigences produit** qui décrivent le produit ou le système à un haut niveau. Elles répondent aux exigences métier et sont couramment formulées comme les fonctionnalités que le système doit réaliser. On les appelle également exigences fonctionnelles ou spécifications fonctionnelles.
- Les **exigences de processus** qui décrivent le *comment*. Ces exigences prescrivent les processus que l'on doit suivre et les contraintes auxquelles on doit se conformer pour la réalisation du système. Dans ce cas, on trouve par exemple des exigences de sécurité, d'assurance qualité, ou de management.

Les exigences "produit" et de "processus" sont liées. Les exigences de processus sont souvent imposées pour atteindre les exigences "produit" de haut niveau. Par exemple un coût maximum de développement (ce qui est une exigence de processus) peut être imposé afin d'atteindre une exigence sur le prix de vente minimum (qui est une exigence produit). Une exigence de maintenabilité du produit

(exigence produit) est souvent accompagnée d'exigences de suivre un certain style de programmation (exigence de processus) telles que la programmation orientée objet, les motifs de conception ou encore le respect de charte de nommage. Les trois types d'exigences sont vitaux pour tout développement de système.

### 1.3.2 Classification de D. Amyot

D. Amyot Présente une autre classification [5]:

- **Une exigence fonctionnelle** est une exigence définissant une fonction du système en développement *Décrit le quoi, c.-à-d. ce que le système doit faire.*
- **Une exigence non-fonctionnelle (ou extra-fonctionnelle)** est une exigence qui caractérise une propriété ou une qualité désirée du système telle que sa performance, sa robustesse, sa convivialité, sa maintenabilité, etc. *Une contrainte qui doit être prise en compte lors du développement*
- **Un but** est un objectif ou une préoccupation utilisé pour découvrir et évaluer des exigences fonctionnelles et non-fonctionnelles. *Un but n'est pas encore une exigence.*
- **Une exigence utilisateur** est une fonctionnalité ou un but désiré par un utilisateur ou une autre partie prenante. *Elle ne devient pas nécessairement une exigence du système...*
- **Une exigence du domaine** est une exigence dérivée du domaine d'application. *Elle peut être fonctionnelle ou non-fonctionnelle*

### 1.3.3 Classification de Sommerville

D'après Sommerville et all dans [28] :

- **Une exigence client** : exprime *en termes généraux* ce que le système doit faire et sous quelles contraintes,
- **Une exigence système** : définit *de façon précise* le fonctionnement du système,
- **Une exigence d'implémentation** : indique comment le système doit être implémenté,
- **Une exigence de performances** : établit des performances minimales pour que le système soit acceptable.

### 1.3.4 Classification fonctionnelle/non fonctionnelle

Les exigences sont classées généralement en trois catégories selon l'encyclopédie libre:

- **Exigences fonctionnelles** : Elles décrivent les caractéristiques du système ou des processus que le système doit exécuter. On trouve dans cette catégorie les règles métier et les exigences fonctionnelles de sécurité informatique (confidentialité,...)
- **Exigences non fonctionnelles** : Elles décrivent les propriétés que le système doit avoir *figure 3*; par exemple les exigences techniques de sécurité informatique (confidentialité, intégrité, disponibilité), de performance, d'accessibilité, selon des critères définis,

- **Contraintes** : Les limites du développement en quelque sorte, comme la façon de définir un système d'exploitation sur lequel le système doit fonctionner, ou celle de définir quel langage de programmation doit être utilisée pour mettre en œuvre le système.

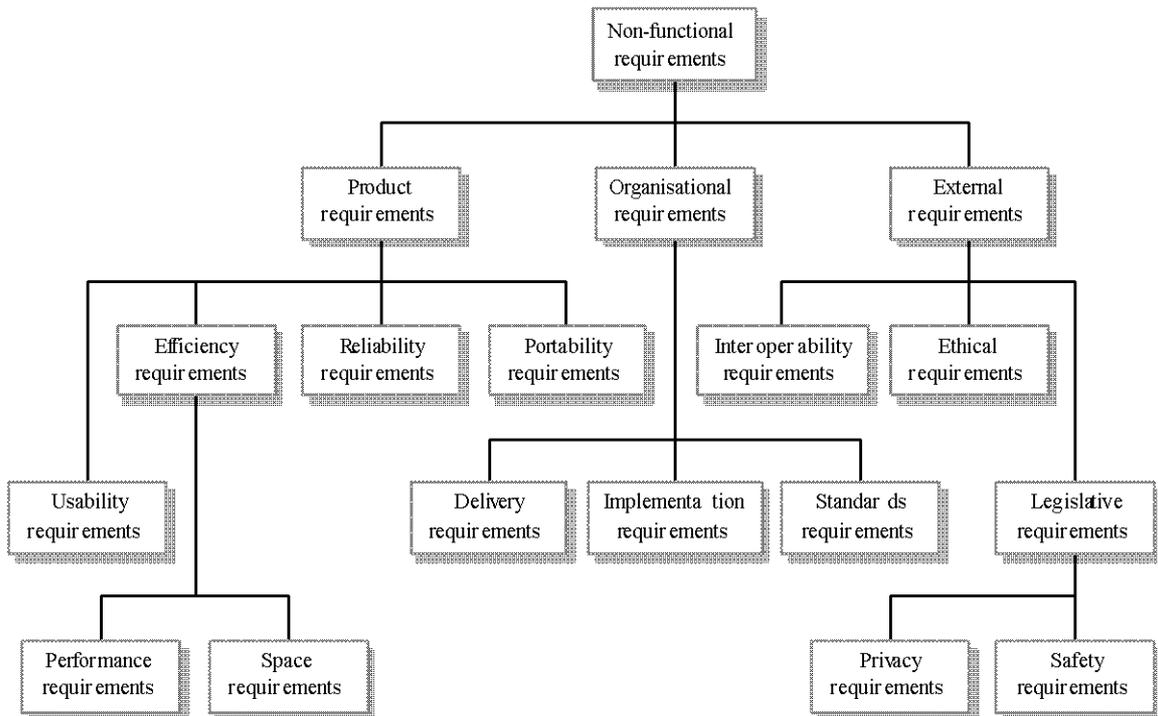


Fig.58. Classification des exigences non fonctionnelle [1]

## 2. les parties prenantes : (Stakeholders)

Les parties prenantes sont les acteurs du processus d'IE, ils sont les individus impliqués dans son exécution. Ils sont identifiés par leur rôle et non individuellement.

L'ingénierie des Exigences fait apparaître des acteurs qui sont intéressés par le problème à résoudre ou par sa solution:

- Clients, usagers, experts du domaine
- Ingénieurs en logiciel, Ingénieurs des besoins
- Chefs de projets.

D'après l'encyclopédie Libre « Une **partie prenante** : est un acteur individuel ou collectif concerné par une décision ou un projet ».

Les intervenants '**partie prenante**' veulent dire selon Gerald Kotonya et Sommerville [1] des personnes ou des organisations qui seront impliquées par le système et qui ont une influence sur la configuration système *figure 4*. Ils pourraient être des utilisateurs finaux, des gestionnaires et autres acteurs impliqués, influencés par le système. Ils sont aussi par exemple pour les ingénieurs responsables du développement du système et de sa maintenance, les clients de l'organisation.

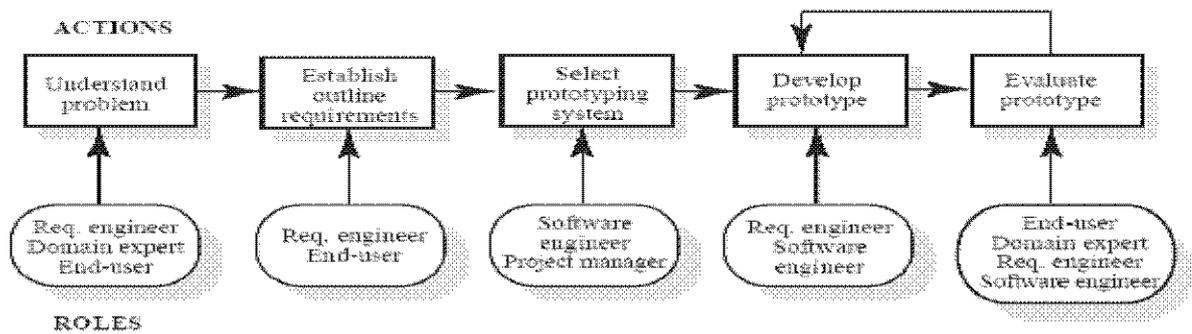


Fig.59. Rôle des parties prenantes dans le processus IE [1]

### 3. Document des besoins (cahier de charges)

Utilisé pour communiquer les besoins entre clients, ingénieurs et managers, Il décrit:

- Les **services** et les **fonctions** que le système doit fournir,
- Les **contraintes** sous lesquelles le système doit opérer,
- Les **propriétés globales** du système,
- Définitions d'autres systèmes avec lesquels le système décrit doit **coopérer**.

Les Utilisateurs du Cahier des Charges sont :

- *Clients du système : Identifient les besoins (début) et les examinent (fin) pour vérifier que ceux-ci correspondent à leurs attentes*
- *Chefs de projets : Utilisent le cahier des charges pour planifier le processus de développement du système*
- *Ingénieurs du système : Utilisent le cahier des charges pour comprendre le système en cours de développement*
- *Ingénieurs du test du système : Utilisent le cahier des charges pour développer des tests*
- *Ingénieurs de maintenance : Utilisent le cahier des charges pour comprendre le système en cours de maintenance.*

## 4. Ingénierie des Exigences IE (origine du terme : IEEE 85)

### 4.1 Définitions

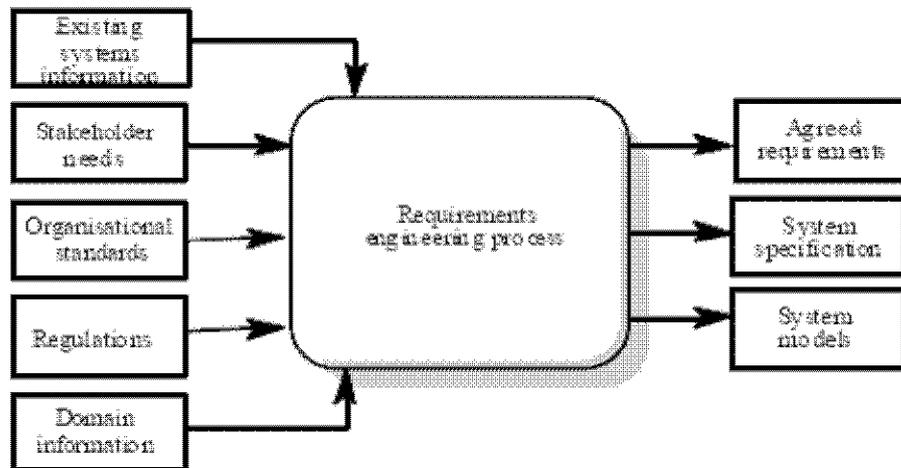


Fig.60. Les entrées /Sorties du processus d'IE [1].

- selon Gerald Kotonya et Sommerville [1], « l'IE est une activité systématique et répétitive de découvrir, analyser, documenter et maintenir un ensemble d'exigences pour le développement d'un système logiciel » *figure 5*. l'office 2000 de la langue française, la définit comme « l'étude systématique des besoins d'un utilisateur, visant à définir un système. »
- C'est l'ensemble des bonnes pratiques permettant de définir clairement le contexte de travail au sein d'un projet, aussi bien d'un point de vue **contractuel** que **technique**.

C'est un point d'entrée majeur au Management des Projets.

- « C'est l'activité qui transforme une idée floue en une spécification précise de besoins, souhaits et exigences exprimés par une communauté d'utilisateurs et donc définit la relation existante entre un système et son environnement »
- « C'est la branche de l'ingénierie des systèmes réels, les contraintes et les services fournis par un système complexe. Elle est aussi concernée par les relations entre ces facteurs afin de préciser les spécifications du comportement du système, et son évolution au cours du temps et par rapport aux autres systèmes. Plus directement, l'ingénierie de exigences s'attache à porter des améliorations sur le plan du cycle de vie du développement du système. Ceci en mettant en évidence les besoins qui ont conduit au processus de développement et en organisant cette information sous une forme servant de base à la conception et à l'implémentation du système »[8].
- ADNSOFT<sup>2</sup> la définit par « Comment comprendre, modéliser puis faire correspondre les besoins qu'ils soient informels, semi-formels ou formels, exprimés par les différents donneurs d'ordre, aux abstractions des modèles métier livrables qui constituent l'aboutissement de l'étape

<sup>2</sup> <http://www.adnsoft.biz/>

de convergence indispensable entre les parties prenantes que sont la Maîtrise d'ouvrage et la Maîtrise d'œuvre, et qui représentent le fondement même du projet et de sa réalisation ?

- L'Ingénierie des Exigences consiste, au travers de méthodes, règles et processus, à établir et maintenir un référentiel unique. Elle permet de démontrer la satisfaction des besoins et des engagements tout au long du cycle de vie [4].

#### 4.2 L'Ingénierie des Exigences Logiciel / Système :

Dans la littérature, il apparaît une confusion entre les deux termes *IE Logiciel et/ou IE Système*

Si la spécification décrit les deux parties matériel et logiciel, il est appelé la spécification des exigences du système; s'il décrit seulement la partie logiciel, il est appelé la spécification des exigences du logiciel (cf. [IEEE-830, 1984]).

#### 4.3 Les Mondes de l'Ingénierie des Exigences

L'ingénierie des exigences utilise des informations qui appartiennent à quatre mondes différents, monde de l'usage, monde du sujet, monde de développement et monde du système figure 6.

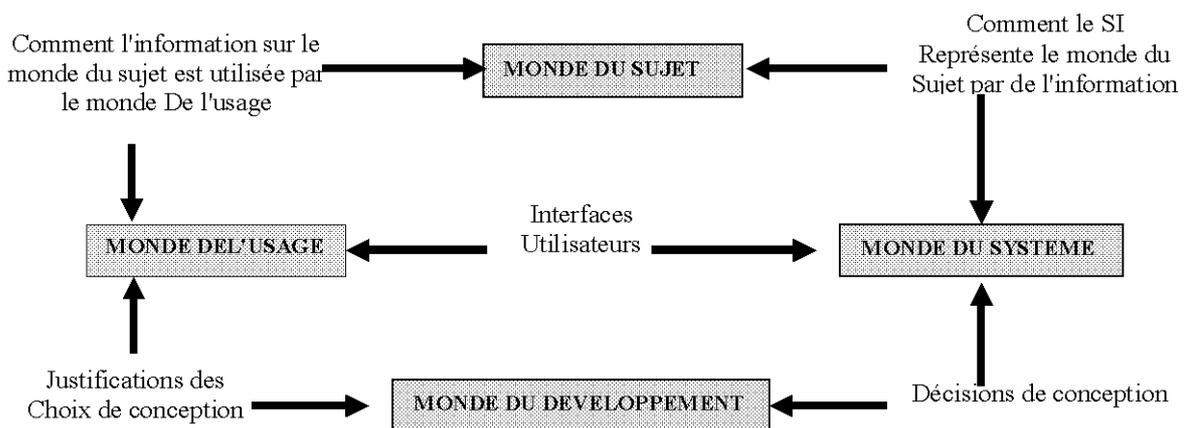


Fig.61. Les Mondes de l'ingénierie des exigences

- Le monde de l'usage est l'environnement social et organisationnel dans lequel le système doit fonctionner.
- Le monde du sujet est le domaine pour lequel le système d'information mémorise de l'information où il discerne de :
  - ✓ Grande variété de domaines
  - ✓ Des similarités évidentes et cachées
  - ✓ Caractérisation et classification de domaines
  - ✓ Exploitation des similarités à des fins de réutilisation
  - ✓ Contrôle de l'IB par de la connaissance générique de domaine
- Le monde du développement est relatif aux entités et activités qui participent au processus d'IB

- **Le monde du système** est celui des spécifications du système dans lequel les besoins issus des deux autres mondes doivent être formulés et documentés. Ce monde détient aussi les représentations des entités, événements, processus des mondes du sujet et de l'usage ainsi que leur transformation en spécifications techniques et implémentations logicielles.

## LES DISCIPLINES QUI CONSTITUENT LES FONDEMENTS DE L'IE

L'IE est un processus multidisciplinaire, centré sur l'homme. Les outils et les techniques utilisés dans l'IE s'appuient sur toute une variété de disciplines, et il est attendu de l'ingénieur des exigences qu'il maîtrise les compétences d'un certain nombre de disciplines différentes.

Une nouvelle IE doit être sensible à la façon dont les gens perçoivent et comprennent le monde autour d'eux, comment ils interagissent, et comment la sociologie du travail influence sur leurs actions. L'IE s'appuie sur les sciences cognitives et sociales pour fournir à la fois des bases théoriques et techniques pratiques pour éliciter et modéliser les exigences:

### **1. La psychologie cognitive**

Permet de comprendre les difficultés que peuvent avoir les gens en décrivant leurs besoins [12]. *Par exemple*, le problème des experts du domaine qui ont souvent de grandes quantités de connaissances tacites qui ne se prêtent pas à l'introspection, d'où leurs réponses aux questions posées par les analystes, qui peuvent ne pas correspondre à leur comportement. Aussi, l'ingénieur des exigences peut avoir besoin de comprendre des modèles utilisateurs des interfaces logicielles, plutôt que de compter uniquement sur les préférences d'implémentation.

### **2. Anthropologie**

Fournit une approche méthodologique pour l'observation des activités humaines qui aide à développer une meilleure compréhension de la façon que les systèmes informatiques peuvent aider ou entraver ces activités [13].

Par exemple, les techniques de l'ethnométhodologie [14] ont été appliquées dans l'IE pour développer des techniques d'observation en vue d'analyser le travail collaboratif et d'interaction de l'équipe.

### **3. Sociologie**

Permet de comprendre les changements politiques et culturels provoqués par l'informatisation. L'introduction d'un nouveau système informatique modifie la nature du travail effectué au sein d'une organisation, peut affecter la structure et les voies de communication au sein de cette organisation, peut même changer les besoins initiaux pour lesquels il a été construit pour les satisfaire [15]. L'exercice de collecte des exigences ne peut donc se politiser. Les approches de l'IE qui abordent cette question comprennent l'approche " Scandanavian ", qui vise à impliquer dans le processus de définition des besoins, ceux les plus touchés par les résultats [16].

#### 4. La linguistique

Est importante parce que l'IE est pour une grande part un sujet de communication. Les analyses linguistiques ont changé la façon dont la langue anglaise est utilisée dans les spécifications, *par exemple* pour éviter toute ambiguïté et améliorer l'intelligibilité.

Des outils de la linguistique peuvent également être utilisés dans l'élicitation des exigences, par exemple pour analyser les schémas de communication au sein d'une organisation [17].

Enfin, il ya un élément philosophique important dans l'IE.

L'IE s'intéresse à interpréter et à comprendre la terminologie des parties prenantes, des concepts, des points de vue et ses objectifs. Par conséquent l'IE doit se préoccuper de la compréhension des croyances des différentes parties (l'épistémologie), la question de ce qui est observable dans le monde (phénoménologie), et la question de ce qui peut être d'accord sur le plus objectivement vrai (ontologie). Ces questions prennent de l'importance quand on veut parler de validation des exigences, en particulier lorsque les parties prenantes peuvent avoir des objectifs divergents et les systèmes de croyances incompatibles. Ils ont également pris de l'importance lors du choix d'une technique de modélisation, car le choix de la technique affecte l'ensemble des phénomènes qui peuvent être modélisés, et peuvent même restreindre ce qu'un ingénieur d'exigences est capable d'observer.

### PROCESSUS D'INGENIERIE DES EXIGENCES

#### 1. Définition

Le processus d'IE peut être considéré comme un ensemble d'activités contenant une structure dans chaque activité. Ces activités consistent à dire par exemple qui sera responsable de chaque activité, les intrants nécessaires à une activité et les sorties générées par cette activité, etc. Selon Leite et al. [2], "L'ensemble du processus d'ingénierie des exigences est un tissu de sous-processus, et il est très difficile de faire une distinction claire entre eux».

De manière générale, un processus d'ingénierie des exigences inclut les phases d'*Elicitation*, d'*Analyse*, de *Spécification*, de *Validation* et de *Gestion des exigences*. **L'Elicitation des exigences** consiste en la collecte, la capture, la découverte et le développement des exigences à partir d'une variété de sources y compris les parties prenantes humaines. **L'analyse** se focalise sur l'examen, la compréhension et la modélisation des exigences élicitées et leur vérification pour la qualité en termes d'exactitude, de complétude, de clarté et de consistance. **La spécification** est l'enregistrement et la documentation des exigences de sorte qu'elle soit utilisable par les parties prenantes et en particulier les développeurs qui doivent concevoir et construire le système. **La validation** est la confirmation de la qualité des exigences et de leur conformité aux besoins et désirs des parties prenantes. Quant à **la gestion**, elle est exécutée tout au long du processus d'ingénierie des exigences. Elle inclut les activités de planification, de contrôle de version et de traçabilité des exigences.

## 2. Quelques approches du processus d'ingénierie des exigences

- Pohl (1994) a présenté le processus d'IE en trois dimensions *figure 7*: **représentation, agreement et spécification**.
- ✓ **Dimension représentation**: les besoins se découvrent et se décrivent en accord avec un système de représentation
- ✓ **Dimension consensus** : les besoins se négocient en fonction de leur priorité, des coûts et des risques de leur réalisation
- ✓ **Dimension spécification** : les besoins doivent être conformes aux standards

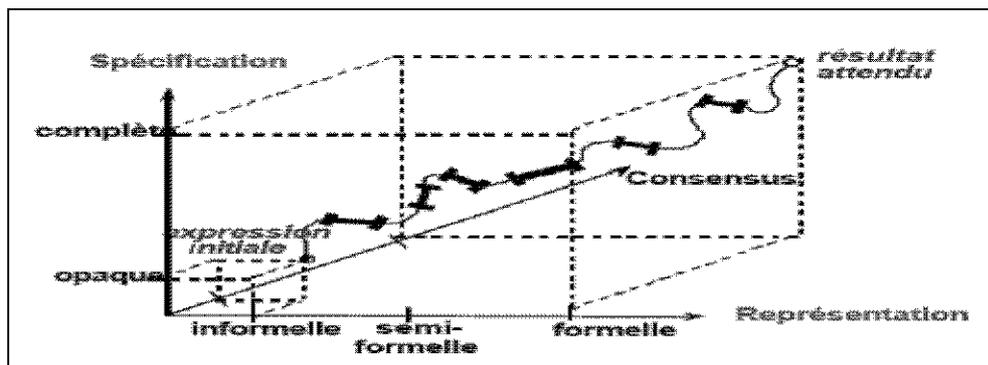


Fig.62. Les trois dimensions de l'ingénierie des exigences [9]

- Loucopoulos, Karakostas [10] utilise la nomination *activités* du processus l'IE: Elicitation, Specification, Validation *figure 8* (et non phase).

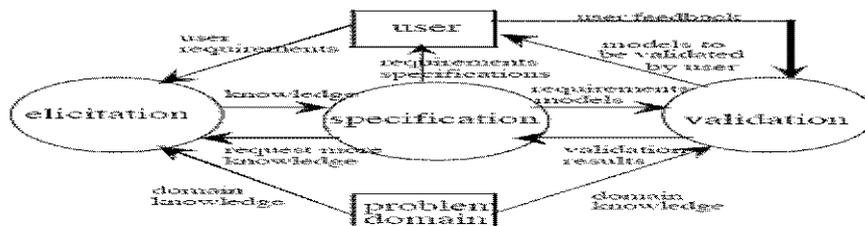


Fig.63. Processus d'ingénierie des exigences de Loucopoulos [10]

- Pohl en (1996) voit le processus d'IE en quatre tâches [11]: **Elicitation, négociation, spécification, et validation**.
- Kotonya, Sommerville (1998) garde la même nomination des activités, avec plus de détail pour la deuxième activité.
  - ✓ **Elicitation, Analysis and negotiation, Documentation, Validation**
  - ✓ **Management** : c'est le processus de gestion des changements des exigences système.

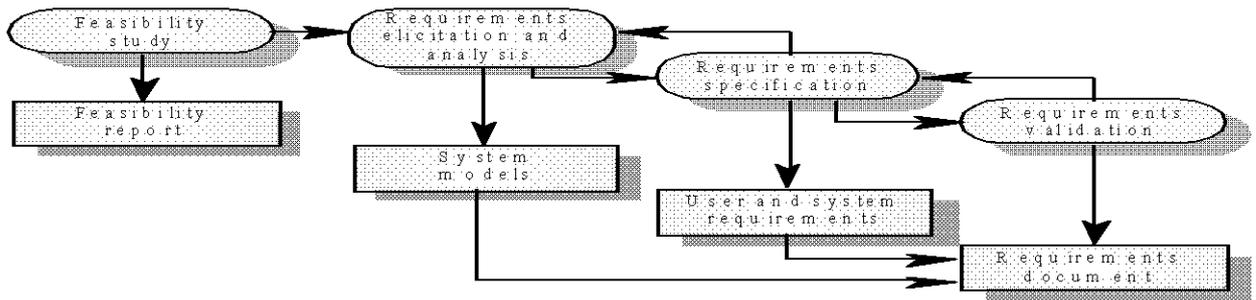


Fig.64. Processus d'ingénierie des exigences de kotonya [1]

- Larry Boldt présente le processus d'IE sous forme d'hierarchie *figure 10*, séparons la tâche de création des exigences de celle de son développement.

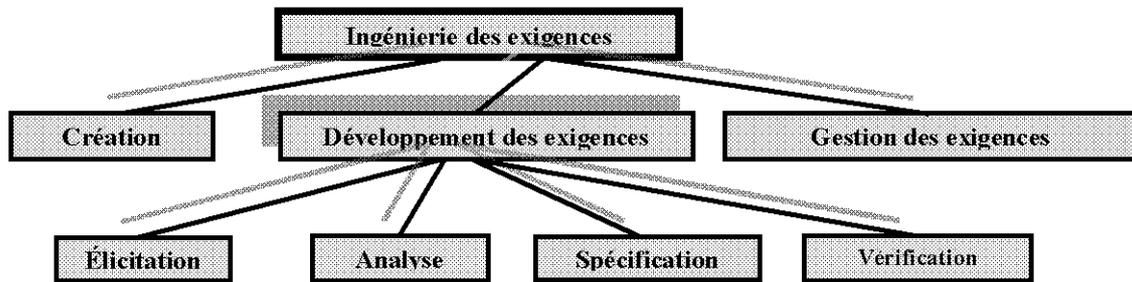


Fig.65. Processus d'ingénierie des exigences de Larry Boldt [19]

- Gerald Kotonya , Ian Sommerville (2004) présente le processus d'IE en activités et activité continu *figure 11* où :

- ✓ Les Activité sont : l' elicitation, l' Analyse et négociation et les activités continus qui sont la documentation, la validation, l'agrément.

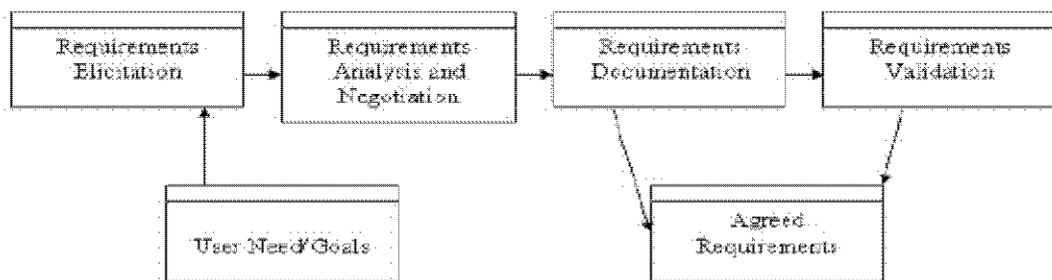


Fig.66. Processus d'ingénierie des exigences Modèle d'activité Coarse-Grain [24]

- Ian Sommerville en 2005 [3], présente un consensus général des activités du processus d'ingénierie des exigences traditionnelles sur mesure qui se compose de cinq principaux domaines

de processus, comme indiqué dans la *Figure 12* : Analyse, elicitation, négociation, documentation, validation et de gestion.

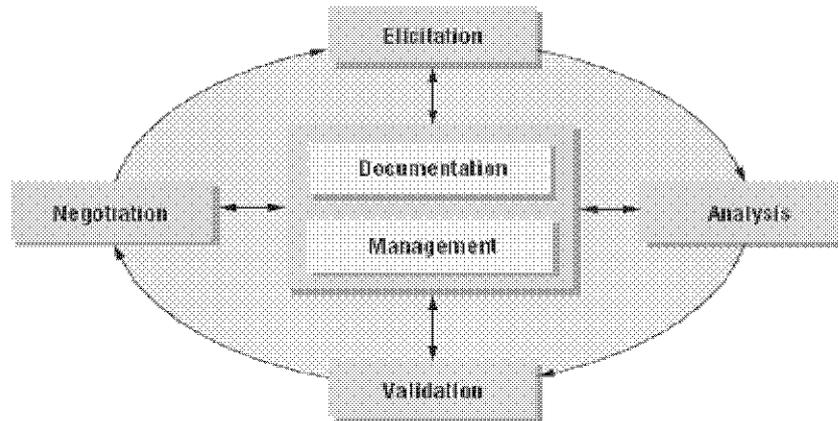


Fig.67. Processus d'ingénierie des exigences de Ian-sommerville [3]

### 3. Les Activité du Processus IE

#### 3.1 Elicitation des exigences

La découverte ou l'élicitation des besoins est une tâche qui consiste à rassembler les informations concernant les besoins des utilisateurs, exprimés initialement par les utilisateurs eux même.

Le responsable de cette phase va construire un outil adapté au contexte de son investigation. Parmi les moyens utilisés, citons :

- ✓ Les entretiens individuels
- ✓ Les réunions
- ✓ Les enquêtes à base de questionnaire
- ✓ L'évaluation des réponses, leur analyse et leur synthèse.
- ✓ Interviews & Brainstorming une ou plusieurs personnes posent des questions à un ou plusieurs utilisateurs et/ou experts afin de rassembler de l'information et des connaissances sur un sujet particulier. Le brainstorming (ou remue-méninges) consiste à rassembler un groupe de personnes choisies à qui l'on demande d'exprimer librement leurs idées, pensées et intuitions sur un ou plusieurs thèmes. Un animateur gère la rencontre et prend note des idées émises, qui seront, par la suite, analysées, classées et éventuellement approfondies.
- ✓ Workshops d'IE
- ✓ Observation des utilisateurs

#### 3.2 L'analyse des exigences

L'analyse est l'une des premières phases en ingénierie des exigences et le but est d'analyser les exigences élicitées. Lorsque les exigences sont découvertes, les conflits, les chevauchements, les omissions et inconsistances doivent être analysés [1] [28].

### 3.3 Spécification des exigences

La spécification des besoins consiste à retranscrire sur papier les exigences dégagées lors de l'élicitation. Les « modèles » peuvent être graphiques (schémas entité / association, UML, etc.) ou textuels, avec un degré de formalisme plus ou moins élevé allant du langage naturel (mais dans ce cas, on ne peut parler de « modèle » au sens strict) à des descriptions formelles fondées sur des bases logiques ou mathématiques.

Il est indispensable de déterminer les besoins du logiciel pendant la première phase. Les besoins peuvent se traduire sous plusieurs formes: des spécifications générales, des spécifications fonctionnelles, et des spécifications d'interface.

Les spécifications générales sont un ensemble d'objectifs, de contraintes (utilisation de matériels et outils existants) et de généralités qu'il faudra respecter au cours du développement.

- Les spécifications fonctionnelles sont la description des fonctionnalités du logiciel de manière aussi détaillée que nécessaire *figure 13*.
- Les spécifications d'interface sont la description des interfaces du logiciel avec le monde extérieur (hommes, autres logiciels, matériels...) de manière aussi détaillée que nécessaire.
- Les spécifications des besoins servent à définir ce que doit faire le logiciel et non comment il est fait. Ceci est décrit dans le document de spécifications des besoins.

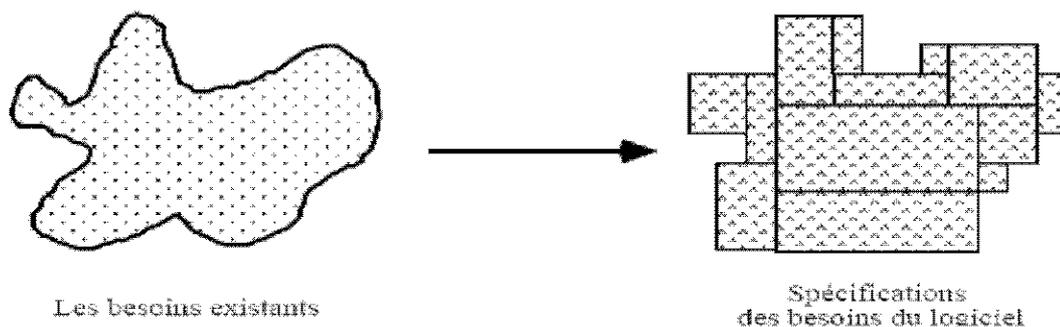


Fig.68. Processus de spécification

### 3.4 Négociation des exigences

Sommerville dans [1] explique que les désaccords au sujet des besoins sont *inévitables* lorsqu'un grand nombre d'intéressés sont impliqués. Les conflits ne sont pas des « échecs » mais reflètent les

*divergences* dans les *exigences* et les *priorités* des intéressés. Il englobe la *discussion des conflits* de besoins et la recherche d'un *compromis approuvé* par tous les intéressés. Il est important de laisser *assez de temps* pour la négociation dans la planification pour le processus d'ingénierie des besoins. La conclusion d'un compromis acceptable peut prendre du temps.

Les Activités de négociation sont :

- **Discussion** : Les besoins identifiés comme problématiques sont discutés et les intéressés exposent leurs visions sur chacun de ces besoins.
- **Établissement de priorités** : Chacun des besoins discutés reçoit une priorité afin d'identifier ceux qui sont critiques et par là faciliter la prise de décision.
- **Consensus sur les besoins** : Des solutions aux problèmes des besoins sont trouvées et un ensemble compromis de besoins consensuels est établi. Généralement, cela implique des changements dans certains besoins. L'objectif de cette tâche est d'établir un accord (consensus) des différents partenaires du système autour des besoins dégagés dans la phase de découverte des besoins, car il se peut que les exigences émises par les différentes catégories d'acteurs soient divergentes voire contradictoires. Dans ce cas, il est nécessaire de négocier afin de dégager un consensus et de parvenir à un compromis sur les exigences à atteindre.

Amyot [18] considère que la négociation d'un ensemble cohérent d'exigences n'est pas une chose facile :

- ✓ Premièrement, on doit *détecter* quand les exigences utilisateur sont incohérentes.
- ✓ Ensuite, on doit *convaincre* toutes les parties prenantes de comprendre leurs exigences essentielles respectives.
- ✓ Enfin, on doit arriver à une *entente* sur un ensemble d'exigences cohérentes qui satisfait le plus possible les parties prenantes.

Les différentes difficultés de la négociation sont :

- ✓ Il y a trop d'exigences!
- ✓ Les budgets, le temps, et les autres ressources sont limitées.
- ✓ Les exigences fusent de partout!(Quelles sont les plus importantes? Pour qui?, Comment les prioriser ou en faire le triage? , Sur quels critères?, À quelle version/phase seront-elles associées)
- ✓ Les développeurs ne connaissent pas la valeur des exigences demandées, les clients ne connaissent pas la complexité de leur développement...
- ✓ Différents intervenants auront différents buts et différentes priorités (Différents marchés, Les clients n'ont pas tous le même poids)
- ✓ Les entreprises manquent souvent de données, métriques et techniques systématiques pour supporter le triage (Souvent fait manuellement, de façon ad hoc, Difficiles à établir et à communiquer)
- ✓ Attitude!

•« Pas besoin de priorités, on peut tout faire ce qu'on retrouve dans la spécification! »

•Oui mais quand et à quel coût?

•Parfois, alors que l'échéance arrive à grands pas, quelques exigences sont mises de côté pour pouvoir livrer quelque chose à temps...

✓ Il est difficile de satisfaire tout le monde, d'atteindre tous les buts fixés, de prendre les bonnes décisions!

✓ Besoins de compromis, de négociation, de priorités... de triage!

### 3.5 Validation des exigences

Les besoins doivent être validés soigneusement pour s'assurer, qu'ils sont complets, corrects et non ambiguës, etc. La validation, qui fait l'objet de ce travail, consiste à confronter le cahier de charges aux acteurs afin de s'assurer de son adéquation avec leurs besoins. On communique donc aux acteurs les modèles élaborés en leur demandant de les accepter, de les commenter, de les amender ou de les refuser. Les modèles graphiques ou en langage naturel se prêtent assez bien à cette validation par des utilisateurs, mais dans le cas de modèles formels plus complexes, il est recommandé de faire une traduction plus facilement compréhensible ou de recourir à d'autres techniques comme celle de la réalisation d'un prototype.

*D'après* Kotonya [1,] l'objectif de la validation des exigences est de certifier le fait que le cahier des charges (surtout les spécifications du système) constitue une description acceptable du système à implanter

Le(s) document(s) produits sont examinés pour vérifier essentiellement :

- ✓ Complétude et consistance
- ✓ Conformité aux standards
- ✓ Conflits entre besoins
- ✓ Erreurs techniques
- ✓ Ambiguïté dans les besoins.

## 4. Documentation des exigences

Les exigences doivent être documentées pour permettre la communication et l'agrément des exigences ainsi que la traçabilité des produits des autres travaux. Ces documents, à savoir, les spécifications, peuvent être écrites dans un langage naturel ou en utilisant un langage de description des exigences (par exemple, les méthodes formelles). Aussi, des techniques de modélisation devraient être utilisées pour décrire les exigences (par exemple, UML, Diagramme de flux de données, etc.)

Les documents de spécification des exigences devraient suivre un format standard, plusieurs formats ont été proposés dans la littérature, par exemple, IEEE Std 830A-1998; [25] et [26].

Les spécifications sont définies dans un arbre de spécification qui suit la hiérarchie système. Chaque élément livrable (ou d'un assemblage d'éléments) devrait avoir une spécification, dans laquelle l'acceptation des éléments est reflétée. Après que les spécifications sont prêtes, une conception de référence globale est établie [27]. Différents documents relatifs aux exigences peuvent être nécessaires aussi pour les différents points de vue [26] :

- Le système est très complexe → Un document «parent», qui définit les exigences de l'ensemble du système (spécification des exigences système), y compris le matériel, le logiciel, les acteurs, et les procédures, et un autre document définit les exigences pour seulement la partie logicielle (spécification des exigences logicielles à-dire, SRS (software requirements specification i.e., SRS)).
- Les besoins du client sont documentés avant la documentation des exigences détaillées → Un document définit les caractéristiques du système en termes généraux (document de vision), et un autre définit les exigences en termes plus spécifiques (SRS).
- Le système peut être membre d'une famille ou de produits enchaînés ou le système en cours de construction ne satisfait qu'à un sous-ensemble de toutes les exigences identifiées → Un document définit l'ensemble des exigences pour une famille de produits (document exigences relatives à la famille du produit ou vision de document de la famille de produits), et un autre définit les exigences pour une seule application spécifique et une version spécifique (SRS).
- L'objectifs du marketing et du commerces ont besoin d'être séparés des détaillées sur les exigences produits → Un document décrit les exigences de commerces globales et le contexte commercial dans lequel réside le produit (documents des exigences commercialisation ou documents des exigences marketing), et un autre définit le comportement extérieur du système en cours de construction (SRS).

## LES TECHNIQUES UTILISEES EN IE

Ce Point présente des techniques d'ingénierie des exigences, qui ont été rassemblées et documentées dans la littérature. Des méthodes générales, qui peuvent être utilisés durant les différentes phases du processus d'IE,

### 1. Le brainstorming

Est une technique de groupe pour générer de nouvelles idées utiles et la promotion de la pensée créatrice. Il vise à obtenir un grand nombre d'idées à partir d'un groupe de personnes en peu de temps. [21] En ingénierie de exigences, Les brainstorming peuvent être utilisés pour découvrir les exigences non identifiés et de solliciter les exigences des parties prenantes.

### 2. Enquête contextuelle

Enquête contextuelle est un champ structuré de méthode d'entrevue pour collecter les données de la structure de travail pratique, rendant ainsi les connaissances au sujet du travail explicites afin que les concepteurs puissent le comprendre, et d'obtenir les détails de bas niveau du travail qui est devenue habituelle et invisible. Ces enquêtes sont basées sur un petit nombre de principes fondamentaux:

- Comprendre le contexte dans lequel un produit est utilisé.
- Compter l'utilisateur comme un partenaire dans le processus de conception.
- L'utilisabilité de processus de conception doit avoir une orientation.
- Interprétation.

CI est plus un processus de découverte d'un processus d'évaluation. Il est également plus l'apprentissage que test [22].

### 3. Réunion

Le but de ces techniques est d'essayer d'obtenir un effet global par laquelle un groupe de personnes qui peuvent apporter une meilleure connaissance de leurs besoins que en travaillant individuellement. En utilisant les techniques de réunion on peut aboutir des exigences plus riches et plus cohérentes qui pourraient être réalisables [23].

### 4. Interviews

L'entrevue est une technique applicable à l'élicitation des exigences. [1] En pratique les entretiens sont des discussions entre l'ingénieur d'exigences et les différents intervenants sur le système avec un but de constituer une compréhension des exigences des parties prenantes. Il existe deux types d'interviews:

✓ **Interview Structuré (privées)** : ce qui est fait avec prédéfini de questions par l'ingénieur d'exigence.

✓ **Interview ouvert** : qui est une discussion sur ce que les intervenants attendent du système sans un agenda prédéfini à accomplir dans une composition non limitée.

### 5. Observation

Le but de cette technique est d'observer les parties prenantes sur le terrain. [1] L'ingénieur d'exigences apprend à propos des tâches des utilisateurs en se plongeant dans les environnements et en observant comment les utilisateurs interagissent avec leurs systèmes et l'autre.

### 6. Prototypage

Un prototype d'un système est une première version du système, idéalement rapide à faire et peu coûteux, qui est disponible dans une phase précoce du processus de développement. Ce prototype peut être utilisé pour aider à l'élicitation et l'analyse des exigences système.

Il existe plusieurs techniques et différentes approches pour le prototypage, mais elles entrent habituellement dans l'une des deux catégories principales: prototypage jetable et prototypage évolutif. La principale différence entre ces deux est que dans le prototypage jetable, celui-ci est créé pour aider à recueillir et à analyser les exigences système, qui sont difficiles à comprendre pour le client. En prototypage évolutif, un prototype de système durable avec des fonctionnalités limitées est mis à la disposition des utilisateurs au début du processus de développement et modifié et la suite étendu à la production du système final [1].

7. **Les patrons d'ingénierie des exigences** : est un formulaire qui contient une solution documenté du problème d'ingénierie des besoins pour une utilisation ultérieure

- Avantages
  - Pattern résout les conflits
  - mélange les méthodes formelles et informelles
- Problèmes
  - Besoin d'une bonne technique de modélisation permettant d'éviter de ré analyser et de redéfinir le patron
  - Impossible de gérer les spécifiques exigences / solution: anti-pattern

## 8. METHODE QFD

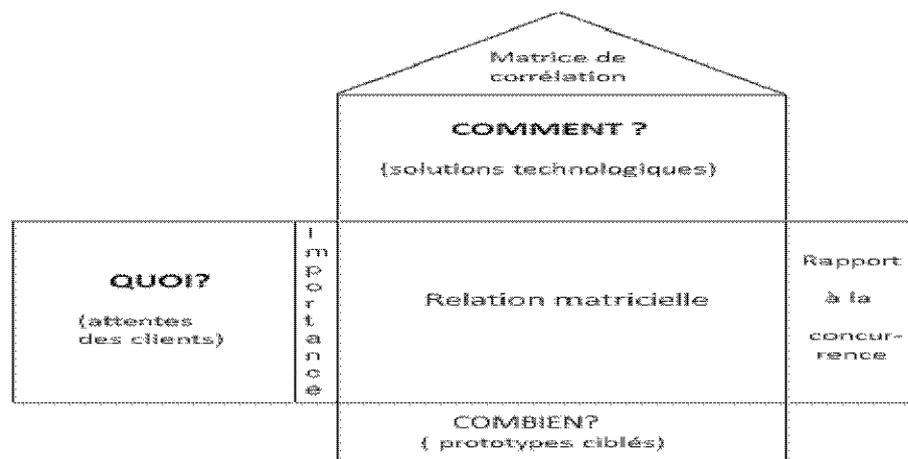
- **La méthode QFD (Déploiement des Fonctions Qualité)**

Le déploiement de la fonction qualité (QFD) est une technique utilisée pour analyser les besoins du client et les relier aux actions qui les satisfont. Elle utilise une matrice mettant en relation les besoins et les spécifications pour les satisfaire.

- **La technique est utile dans la gestion de la qualité totale**

Elle n'est pas d'origine une technique d'ingénierie des exigences, mais plutôt une méthode systématique pour traduire les exigences du client dans la conception spécifiques des produits cibles. Les premiers produits n'étaient pas des Échantillon logiciels telles que les parties du corps Madu [31] et de l'automobile Akao [32]. Le concept fonctionne bien avec les logiciels en raison de l'importance du rôle du client dans le génie logiciel .L'introduction du client est l'un des moyens les plus sûrs pour améliorer les chances de réussite d'un projet logiciel. La méthode QFD considère le client comme *l'acteur essentiel* et ce dès la conception, d'un produit ou d'un service. L'idée centrale est de traduire la demande du client dans le langage

du



concepteur, de manière à concevoir et à réaliser un produit qui réponde précisément à cette demande et ce, sans coûts superflus. D'après les travaux de Bocquet et Chuen,[33] ,[34] cette méthode permet de répondre à trois questions : " **quelles sont les attentes clients à considérer en priorité pour assurer la réussite du produit ?** ", " **quelles sont les réponses techniques à privilégier ?** " et " **quelles sont les difficultés potentielles du cycle de développement du « produit » ?** ". La réponse à ces questions est apportée par l'équipe du projet formée de personnes complémentaires réunies autour de la problématique à solutionner. Les attentes explicites et implicites des clients (les QUOI), point de départ du QFD, sont documentées par des enquêtes menées auprès de la clientèle. L'analyse fonctionnelle technique quant à elle peut être utilisée pour structurer et organiser le découpage du produit en un ensemble de fonctions sans caractérisées (les COMMENT).

Fig.69. Principe de la méthode QFD [35]

- **Avantages et travaux voisin en IE**

La méthode QFD est utilisé dans la littérature pour plusieurs raisons et de différente façons J. SAADI et al [37] a proposé une approche basée sur le déploiement de la fonction qualité QFD pour l'adaptation des programmes universitaires aux besoins du marché .la QFD a été utilisé pour permettre d'évaluer l'adéquation à l'emploi de filières existantes ainsi que l'aide à la conception de nouvelles filières intégrant les exigences du client qui est le recruteur potentiel. Dans Estelle FREY et all [36] [2007] ils ont parvenu à extraire des connaissances expertes en conception à partir de la méthode QFD, où ils ont utilisé des matrices QFD de différents niveaux qui croisent la "voix du client" (besoin) avec le "voix du technicien" (solution) et donnent la possibilité de comparer les solutions (partielles), deux à deux afin d'évaluer leur compatibilité via le toit de corrélation de la matrice, pour une intégration dans une solution globale.

En ingénierie des exigences, la QFD a été mentionné dans Estelle FREY et al [6] pour la comparer avec d'autres techniques de négociation afin de surmonter les conflits entre les parties prenantes où il est exprimé par exemple, que dans la méthode Qualité-Fonction-Déploiement (QFD Quality Function Deployment) , les matrices sont construites de manière à comparer les exigences fonctionnelles avec un autre et le taux de leur importance, sans pour autant identifier clairement les objectifs des parties prenantes.

Avant QFD	Après QFD
Travail individuel	Travail d'équipes
Basé sur certains clients	Basé sur L'intention du client
Développement lourd	Supports l'ingénierie simultanée
Pauvre Documentation	Supports l'intégration dans le développement produits
Mauvaise Communication	Amélioration de la communication/ documentation

TABLE IX. AVANT ET APRES L'UTILISATION DE LA QFD

*Cette méthode QFD va être utilisée par notre méthodologie pour formuler les résultats des deux activités d'elicitation et spécification des exigences afin que les parties prenantes puissent affecté des poids associé chaque entrée de la matrice (Exigence, spécification). Désignons leur point de vues a fin de passer ensuite à leur négociation.*

## LA GESTION DES EXIGENCES

La gestion des exigences (GE) peut être considérée comme un processus parallèles pour les autres processus l'ingénierie des exigences [28]; [1]. Elle continue également après la phase de spécification des exigences, parce que les besoins continuent d'évoluer au cours du développement du système et ces changements doivent être gérés [1]. [28] définit des principes concernant la gestion des exigences:

- ✓ Les changements à l'agrément des besoins des exigences doivent être gérés.
- ✓ Les relations entre les exigences doivent être décrites et gérées.
- ✓ Les relations entre les documents d'exigences et d'autres documents produits au cours de l'ingénierie des systèmes et des logiciels ont besoin d'être gérées.

Selon ces principes, les principales préoccupations de la gestion des exigences sont le contrôle des changements et la traçabilité. Le dernier principe met l'accent sur la traçabilité entre les exigences et la

conception, l'implémentation, et le test des artefacts. D'autre part, selon [1], l'identification des exigences est une condition préalable essentielle pour la gestion des exigences. Elle met l'accent sur l'attribution d'un identifiant unique pour chaque exigence [28]. Ces identifiants uniques sont utilisés pour désigner les exigences durant le développement et la gestion du produit. [28] propose des directives pour la GE, qui élargissent les concepts GE pour couvrir aussi la planification GE et les issues d'automatisation.

GE peut être organisé en quatre activités principales voir (Figure 13):

- Identification des exigences: toutes les directives qui sont en rapport avec l'identification et le stockage des éléments d'exigence.
- Traçabilité des Exigences: toutes les directives qui sont en rapport avec la traçabilité des exigences.
- Exigences de gestion du changement: toutes les lignes directrices, qui sont en rapport avec la gestion des changements des exigences.
- La planification de la gestion des exigences: lignes directrices, qui sont en rapport avec la planification et la documentation des activités d'identification, de traçabilité et de gestion du changement ainsi que la définition d'autres objectifs GE, les responsabilités et les politiques pour un projet. Le planning fournit les moyens de choisir et de définir des procédures appropriées lors de l'examen pour GE un projet.

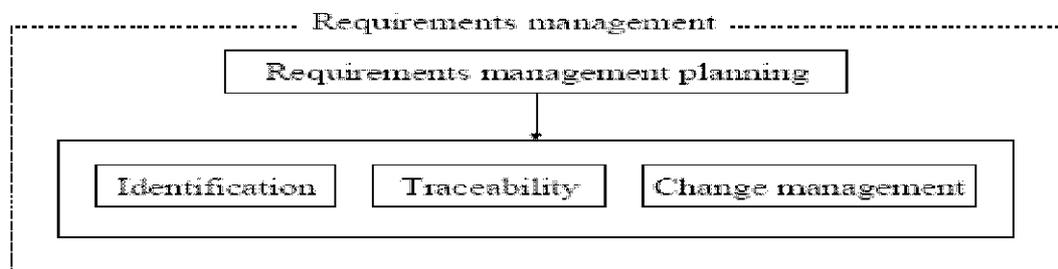


Fig.70. Activités principale de la Gestion des exigences [20]

## OUTILS D'INGENIERIE DES EXIGENCES

Gerald kotonya [1] indiquent que les outils qui support l'ingénierie des exigences, peuvent être divisés suivants deux types:

- **Outils de modélisation et validation :** support le développement de modèles systèmes où la complétude et la consistance des modèles peuvent être vérifiées.
- **Des outils de gestion :** permettent de gérer la base de données des exigences et support la gestion des changements des exigences.

Selon kotonya, les outils de modélisation peuvent être basés sur des méthodes structurées (par exemple, SADT) ou des langages de modélisation des exigences spécialisées (par exemple, RSL). Ces outils permettent la création de modèles graphiques et textuels des exigences et la vérification de leur consistance. Les outils de validation, conformément aux [1], peuvent «analyser la description inconsistance mathématiques qui impliquent soit des erreurs dans la spécification formelle ou des erreurs des exigences ».

D'autre part, les principales fonctions de gestion des exigences sont selon [1] la gestion des changements, les relations et les dépendances entre les documents d'exigences et d'autres documents pendant tout le cycle de vie du produit. Les outils de gestion de GE ont été développés en raison des problèmes de gestion instable des exigences et la grande quantité de données collectées pendant le processus d'IE.

Les outils de gestion supportant la gestion de base de données des exigences et les changements de ces exigences.

Les organisations choisissent les outils appropriés à l'utilisation. La sélection d'outil peut dépendre de divers facteurs, tels que [28].

- ✓ Les outils qui sont déjà utilisé dans l'organisation, et de nouveaux outils qui sont compatibles avec les systèmes existants.
- ✓ Les outils par rapport au budget de formation.
- ✓ La taille du système en cours de développement.
- ✓ Les lignes directrices spécifiques que vous souhaitez implémenté.
- ✓ Stabilité de l'outil offrant aux entreprises (qui signifie également la disponibilité des outils).

### ***Exemples d'outils IE***

Il existe de nombreux outils, qui contiennent des fonctionnalités liées pour supporté l'IE.

Différents outils se sont concentrés sur différents aspects d'IE selon leur origine. Ainsi, l'intention est de fournir au lecteur une idée de la variété des outils de haut niveau relié à l'IE. Il faut garder à l'esprit que la présentation qui suit des outils IE est loin d'être exhaustive.

Des Collection d'information sur les outils IE peuvent être trouvées à partir de ces sites Web:

- Les descriptions de Rabi Archrafi d'outils d'ingénierie des exigences disponibles sur le site :<http://www.volere.co.uk/tools.htm>.

- La page INCOSE, Contiens une taxonomie d'outils D'ingénierie des exigences, divisé en hiérarchie de catégories, conformément au but d'utilisation.  
[http://www.incose.org/tools/tooltax/reqengr\\_tools.html](http://www.incose.org/tools/tooltax/reqengr_tools.html)

- La page Ian Alexander, qui présente une liste de fournisseur d'outils d'IE payé et gratuit :  
<http://easyweb.easynet.co.uk/~iany/other/vendors.htm>.

## COOPERATION & INGENIERIE DES EXIGENCES

Le Projet CREWS (Cooperative Requirement Engenieirie With Scenario) avec ces différentes approches est le seul lien entre les deux domaines (Coopération & ingénierie des exigences) à notre connaissance, il vise à développer, évaluer et démontrer des méthodes et outils pour l'élicitation et la validation des exigences fondés sur des scénarios coopératifs est applicable. CREWS fusionne et étend deux résultats de recherche : ICARUS et NATURE, qui sont des projets ESPRIT [29].

Le développement de cette méthode a conduit à des progrès dans deux directions, A travers quatre approches CREWS guident les différentes utilisations de scénarios lors de l'acquisition et la validation des exigences systèmes. Deux approches guident l'acquisition des exigences. Justifier par le fait que les approches existantes (à cette décennie 90) ne donnent pas de méthode d'orientation pour représenter des descriptions riches, ou des scènes. Par conséquent, la première approche guide l'élicitation coopérative des exigences système à partir de scènes enregistrées dans des représentations multimédia comme les films vidéo et enregistrements audio, d'obtenir différents types de représentations de modèle et des exigences système. Une méthode et prototype logiciel multimédias support ce processus. La deuxième approche guide l'extraction semi-automatique des exigences systèmes en une description en langage naturel des scénarios et des cas d'utilisation [30]. Où il soumette les cas d'utilisation / scénario à une analyse informatique. Un outil logiciel applique ensuite une série de frames pour la description des cas d'utilisation / scénario à extraire : candidats agents, objets, actions et des exigences du système.

---

## **Conclusion**

---

*Les chercheurs dans la littérature actuelle de génie logiciel arrivent assez rapidement à la conclusion qu'aujourd'hui, il n'existe aucun consensus sur ce qui constitue l'ingénierie des exigences logiciel, ses champs d'application, ses étapes, ses objectifs et ses livrables. Le problème est causé en partie par le fait que l'IE n'a été reconnue que récemment comme une discipline de développements logiciels. L'absence de consensus est aussi causée par les différents degrés de rigueur et formalité par laquelle IE est traitée dans différentes méthodes de développement logiciels.*

*L'absence de consensus dans la définition de l'ingénierie des exigences réside dans le fait que la nature du processus des systèmes d'analyse est mal définie et illustré. Et comme en s'intéressent à la validation en ingénierie des exigences Nous avons jugé utile de présenter dans le chapitre suivants la validation des exigences un aperçu de ce quelle vaut dans la littérature.*

---

---

*CHAPITRE 2 :*

***La Validation Des Exigences***

---

# Chapitre II :

## La Validation des Exigences

---

### *Résumé*

*La validation des Exigences est une activité critique du processus d'ingénierie des exigences, elle permet de s'assurer que celles-ci sont correctes, consistantes, complètes et exactes. Elle est utilisée pour en déterminer les meilleures alors que la vérification des exigences détermine que leur implémentation est correcte par rapport aux besoins.*

*Le principal objectif de la validation est de certifier que les documents et les modèles des exigences sont une description acceptable du système, qui va être mis en œuvre.*

*Les techniques de validation des exigences jouent un rôle pivot pour détecter d'éventuels défauts dans les exigences. Elles se résument dans : la révision, l'inspection, les techniques de lecture, le prototypage, la validation basée modèle, la validation basée test et la validation basée point de vue. Ces techniques peuvent aider à la réalisation des projets, au sein du calendrier, de budget et selon les fonctionnalités désirées.*

*Une discussion des approches les plus récentes, telle que : l'approche expérimentale, l'approche système expert et traitement de langage naturel, présentent des méthodes (automatisées, semi-automatiques ou manuelles) d'aspect, (Formel, semi-formel, informel), apportant des privilèges pour des types de systèmes par rapport à d'autres.*

---

## I. INTRODUCTION

L'erreur est humaine et il n'y a aucune raison de penser qu'elle ne survienne pas au cours du développement du système. Les problèmes peuvent résulter d'une incompréhension entre le spécificateur et le client, d'une ambiguïté dans la documentation, etc. Les erreurs qui surviennent à cette étape et qui ne sont pas corrigées sont souvent les plus persistantes et les plus coûteuses. Il est donc important de mettre en branle des étapes qui permettront de minimiser les erreurs, de les détecter et les corriger le plus tôt possible.

La prévention des erreurs est une question de bonne pratique en génie logiciel, Il est toutefois avisé de prendre pour acquis que des erreurs vont survenir et d'établir des procédures pour s'en prévenir.

Ainsi, le processus d'ingénierie des exigences (comme sous-processus du processus plus global de l'ingénierie système et du génie logiciel) doit être vérifié. La vérification a pour finalité de s'assurer que la fonctionnalité correcte du système-solution a été définie :

- ✓ si le domaine du problème se comporte tel que décrit (dans le document de spécification des exigences) ;
- ✓ si les exigences sont correctement enregistrées ;
- ✓ si le nouveau système se comporte tel que décrit (dans le document de spécification des exigences) ;
- ✓ et si l'étape d'invention (la conception des interactions) est correcte, alors les exigences sont rencontrées.

Ceci met en évidence les zones d'erreurs potentielles et celles qui doivent être vérifiées :

- ✓ La description du domaine du problème reflète-t-elle de façon correcte ses propriétés ?
- ✓ Les exigences (les effets sur le domaine du problème) sont-elles correctement enregistrées?
- ✓ Le comportement inventé du nouveau système produit-il les effets escomptés ?
- ✓ La spécification permet-elle une conception conforme aux attentes ?

La vérification devrait être intégrée dans le processus de spécification des exigences, elle est réalisée à l'aide de différents tests, procédures, vérification et points de contrôle, et peut se produire à plusieurs niveaux. Il faut assurer que les artefacts d'analyse représentent correctement le problème et les exigences.

Vérifier que la spécification est adéquate est une activité très difficile, en raison de l'intangibilité de la spécification. Le prototypage peut être une solution partielle à ce problème en permettant de détecter et de prévenir l'erreur le plus tôt possible.

Lorsque la vérification et la validation sont complétées, on peut entamer par la suite les phases d'Architecture, de conception détaillée, de programmation, d'essais fonctionnels, de déploiement et de mise en exploitation.

L'objectif de la validation (Selon l'encyclopédie libre) est de vérifier que toutes les étapes de fabrication aboutissent à un produit conforme aux exigences de l'Autorisation de Mise sur le Marché (A.M.M.) de manière stable et reproductible. Alors que l'objectif de la validation des exigences est de :

- ✓ Certifier que les exigences du cahier de charge sont conformes à la description du système à mettre en œuvre ;

- ✓ Vérifier que le cahier de charge est essentiellement: complet, consistant, conforme aux normes standard, ses besoins ne sont pas conflictuels, ne contient pas d'erreurs techniques, les besoins ne sont pas ambigus, etc.

Après avoir présenté dans la 1<sup>ère</sup> section de ce chapitre un aperçu et objectif de la validation des exigences ; nous présenterons les différentes définitions des vocabulaires du sujet dans la 2<sup>ème</sup> section, ensuite nous décrirons les problématiques de la validation des exigences selon le monde de la machine dans la 3<sup>ème</sup> section avant de donner en section 4 un aperçu sur quelques critères de qualité à vérifier et à valider dans cette activité. En section 5, nous présenterons les techniques qui peuvent être utilisées durant le processus de validation des exigences. En section 6, nous parlerons des recherches en cours à travers les approches les plus prometteuses, commentée chacune, pour décrire Enfin en section 7 et ce d'une manière succincte la validation des exigences et les systèmes coopératifs.

## II. DÉFINITION

Beaucoup de domaines fusionnent entre les définitions de la validation et la vérification. Ainsi, il est nécessaire de s'entendre sur leurs explications.

### 1. La validation & la vérification

D'après l'encyclopédie libre, La **validation** est une opération destinée à démontrer, documents à l'appui, qu'une procédure, un procédé ou une activité, conduit effectivement aux résultats escomptés. Elle comprend la qualification des systèmes et des équipements. Dans le cas d'une validation d'un procédé, il s'agit d'établir, avec un niveau d'assurance élevé, une preuve documentée qu'un procédé particulier donnera constamment un produit conforme à ses spécifications et à des caractéristiques de qualité prédéterminées. La validation d'un procédé peut prendre la forme d'une validation prospective, concomitante ou rétrospective, ou d'une certification ou revalidation du procédé.

Selon la norme ISO 9000 : 2000, la **Validation** est la Confirmation, par des preuves tangibles, que les exigences pour une utilisation spécifique ou une application prévue ont été satisfaites. **VALIDER**, c'est répondre à la question : « Faisons- nous le bon produit ? » tandis que la **Vérification** est la Confirmation par des preuves tangibles que les exigences spécifiées ont été satisfaites. **VERIFIER**, c'est répondre à la question : « Faisons- nous le produit correctement ? ». Selon La définition de Boehm [66] celle la plus

utilisée pour expliquer la vérification et la validation: La *vérification* est le processus qui s'assure de bien construire le système. La *validation* est le processus qui s'assure de la construction du bon système.

Dans ce même article, Boehm explique qu'il y a souvent une confusion entre validation et vérification, mais la vue conventionnelle est que la vérification est le processus qui vérifie si le logiciel remplit les exigences spécifiées par les utilisateurs, tandis que la validation est le processus qui vérifie si le logiciel remplit les exigences réelles des utilisateurs. »

## **2. La validation & la vérification des exigences**

Selon Artem Katasonov [62] La Validation des exigences est le processus qui permet de vérifier si les exigences, telles que définies, ne contredisent pas les attentes au sujet du système des divers intervenants, et ne se contredisent pas les unes les autres ; c'est le Contrôle de la Qualité des exigences. La validation des exigences est concernée par le processus d'examen du document d'exigences pour s'assurer qu'il définit le bon logiciel (c'est le logiciel auquel les utilisateurs s'attendent). Selon Kotonya et al. [1], "la validation des exigences est préoccupée de vérifier le document des exigences pour sa consistance, complétude et exactitude», Et en [68] précise que les exigences doivent être vérifiées pour: Valider, comprendre, avoir une traçabilité consistante, une complétude, un réalisme et une vérifiabilité. Parce que les modalités de vérification et de validation sont souvent confondues, Terry Bahill [67] a examiné les définitions suivantes:

### **2.1. Vérification des exigences**

Prouver que chaque exigence a été satisfaite. La vérification peut se faire par logique, 'inspection, modélisation, simulation, analyse, examen, test ou démonstration.

### **2.2. Validation des exigences**

Veiller à ce que (1) l'ensemble des exigences soient : correctes, complètes et cohérentes, (2), un modèle peut être créé, qui remplit les exigences, et (3) solution du monde réel doit être construite et testée pour prouver qu'elle répond aux exigences.

### **2.3. Vérification d'un système**

La construction du système correctement : faire en sorte que le système soit conforme aux besoins du système et aussi à sa conception.

### **2.4. Validation d'un système**

Construire le bon système: C'est faire en sorte que le système fasse ce qu'il est sensé faire dans son environnement. La validation détermine l'exactitude et l'exhaustivité du produit final et veille à ce que le système satisfasse les besoins réels des intervenants.

Il y a chevauchement entre la vérification système et la vérification des exigences. La vérification système s'assure que le système est conforme à sa conception et est également conforme aux exigences du

système. La vérification des exigences veille à ce que les exigences soient satisfaites et que la technique de dérivés et de produits nécessaire soit vérifié.

Ainsi, la vérification de la configuration système requise est commune à ces deux processus.

Il existe également un chevauchement entre la validation des exigences et la validation du système. La validation de haut niveau des exigences système est similaire à la validation du système, mais la validation de bas niveau est très différente de la validation du système.

### **3. Le processus de validation**

L'encyclopédie libre présente les activités liées aux études de validation qui peuvent être regroupées en trois phases :

**Phase 1 :** La phase de pré validation ou de qualification, qui couvre toutes les activités liées à la recherche et au développement du produit, à la formulation, aux études des lots pilotes en développement, aux études de mise à l'échelle, au transfert de la technologie pour les lots à l'échelle commerciale, à l'établissement des conditions de stabilité, à la conservation et à la manipulation des formes posologiques finies et en cours de fabrication, à la qualification de l'équipement, à celle de l'installation, aux documents types de production, à la qualification opérationnelle et à la capacité du procédé.

**Phase 2 :** La phase de validation du procédé (phase de qualification du procédé) : elle vise à vérifier que toutes les limites établies des paramètres critiques du procédé sont valides et qu'il est possible de fabriquer des produits satisfaisants, même dans les conditions les plus défavorables.

**Phase 3 :** La phase de maintien de la validation nécessite un examen fréquent de tous les documents liés au procédé, notamment des rapports d'audition sur la vérification de la validation qui garantissent qu'il n'y a pas eu de modification, d'écart, de défaillance ou de modification du procédé de production et que toutes les PON ont été respectées, y compris les procédures de contrôle des changements.

À cette étape, l'équipe de validation s'assure, en outre, qu'il n'y a pas eu de modification, ni d'écart, qui aurait entraîné une requalification et une revalidation.

### **4. Le processus de validation des exigences**

La figure 16, illustre le processus de validation des exigences où les documents d'exigences, les connaissances organisationnelles et les standards d'organisation sont ses entrées. La liste des problèmes proposés, les actions acceptées pour résoudre ces problèmes sont les sorties du processus de validation des exigences.

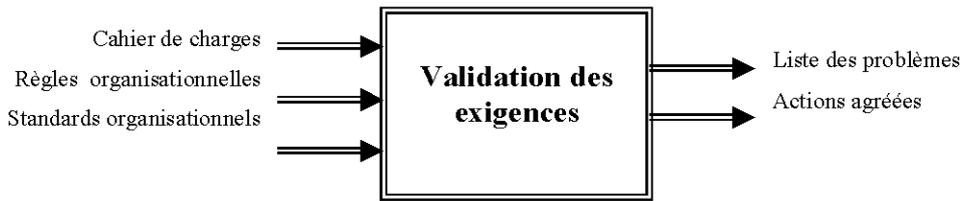


Fig.71. Processus de validation des exigences

Selon la norme EIA632, Le Processus de validation des Exigences permet d'assurer que les exigences sont nécessaires et suffisantes pour la phase de conception approprié afin de répondre aux critères de sortie applicables aux phases de cycle de vie logiciel et les phases de cycle de vie de l'entreprise dans lequel des efforts se produisent pour la phase d'ingénierie ou de réingénierie.

### 5. La nécessité de la validation des exigences:

Les documents relatifs aux exigences peuvent faire l'objet de procédures de validation et de vérification. Davis dans [63] explique que les exigences peuvent être validées pour s'assurer que l'analyste a compris les exigences, et il est également important de vérifier qu'un document est conforme aux normes (standards) d'entreprise, et qu'il est compréhensible, consistant et complet. De même Bryne [64] et Rosenberg [65] affirme qu'il est normal de prévoir explicitement un ou plusieurs points dans le processus des exigences où ces dernières sont validées. Le but est de recenser ou rassembler tous les problèmes avant que les ressources ne soient engagées afin de répondre aux exigences.

### III. PROBLEMATIQUE DE LA VALIDATION DES EXIGENCES

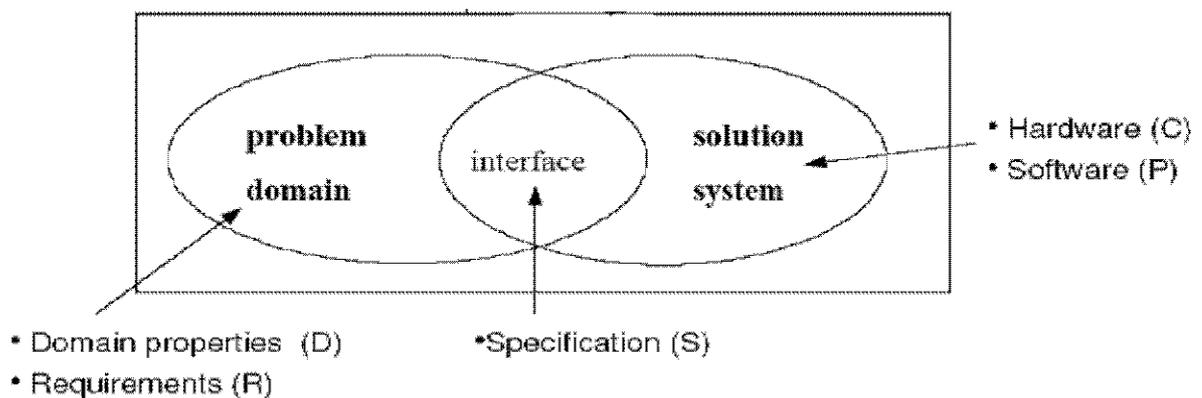


Fig.72. Le monde et la machine [69]

Selon Jackson [69], une Machine (M) est un Système à construire (logiciel et matériel) qui a :  
 Un domaine d'application (A) avec une partie du monde qui interagit (directement ou indirectement) avec M et un Environnement, entités externes qui Aide à définir la portée du système  
 Les éléments de description (E, D, S, C) sont :

- ✓ Exigences (E) : Propriétés attendues c'est l'assertion prescriptives (doit) sur A que M doit contribuer à rendre vraies
- ✓ Description du domaine (D) : Assertions sur A supposées vraies peu importe M c'est l'hypothèse descriptives (pas de doit)
- ✓ Spécification (S) : Assertions prescriptives sur le fonctionnement de M c'est le phénomène partagé entre A et M
- ✓ Conception (C) : Assertions sur le fonctionnement interne de M du niveau de la solution

Ce qu'on cherche à démontrer en validant c'est  $S, D \Rightarrow E$

- ✓ Si l'environnement (A) se comporte comme l'indiquent les hypothèses D
- ✓ Et si la machine (M) se comporte comme indiqué dans sa spécification S,
- ✓ Alors le système respecte les exigences E qui lui sont imposées

Exemple d'analyse P. Heymans.

- ✓ Si les hypothèses D ne sont pas satisfaites par l'environnement A, alors E pourrait ne pas être atteint. M ne peut pas garantir D !

Exemple (cas réel):

- E : "Si l'avion est en mouvement au sol, le pilote doit pouvoir enclencher la poussée arrière."
- D : "Si l'avion est en mouvement au sol, les roues de l'avion tournent."
- S : "Si les roues de l'avion tournent, le pilote doit pouvoir enclencher la poussée arrière."
- Donc:  $S, D \Rightarrow E$
- Mais l'hypothèse D est fautive car si l'avion atterrit sur une piste humide, il peut y avoir de l'aquaplanage !

#### IV. CRITERES DE QUALITE A VALIDER

Comme toute évaluation, la validation au niveau analyse doit être portée sur des caractéristiques qui doivent être remplies.

Les Caractéristiques souhaitées des exigences, seront présentées dans ce qui suit :

Critère	Définition
Modifiable	Permet de vérifier que les spécifications peuvent être changées soit durant le développement du logiciel, soit durant la phase de maintenance. Ces modifications doivent pouvoir être reportées facilement dans le cahier des charges;
Traçabilité	Vérifier : - L'origine de chaque exigence doit être claire ! - facilite le référencement des exigences par une future documentation! - le traçage est la possibilité d'avoir des références croisées entre les spécifications de

	plusieurs versions du cahier des charges (parfois entre les spécifications du cahier des charges et la conception du logiciel). Le traçage arrière consiste à pouvoir, à partir d'une spécification, retrouver la spécification dont elle découle (dans la version précédente du cahier des charges). Le traçage avant consiste, à partir d'une spécification, à trouver les spécifications auxquelles elle a donné naissance;
identifiable	Indiquer l'unicité et l'importance et / ou la stabilité de chaque exigence !
Vérifiable	existe un processus de test de satisfaction de chaque exigence! " le comportement de toutes les exigences est spécifié " Elles doivent permettre de déterminer si elles ont été atteintes ou non selon l'une des quatre méthodes possibles : inspection, analyse, démonstration, ou test.
Concises	–Elles doivent être énoncées dans un langage qui soit précis, bref et agréable à lire, et qui de plus communique l'essence de ce qui est exigé.
Complétude	Décrire les réactions et les sorties attendues du logiciel pour toute entrée licite ou illicite Définir toutes les unités utilisées <i>Complètes</i> : Elles doivent être énoncées entièrement en un endroit et d'une façon qui ne force pas le lecteur à regarder un texte supplémentaire pour savoir ce que l'exigence signifie. La complétude peut prendre deux formes : 'Interne', c'est à dire que tous les concepts utilisés sont clairement spécifiés, 'Externe', par rapport à la réalité décrite. Forme imaginaire dans la pratique
Conflit	Se dit quand les exigences sont contradictoires et par conséquent, le système pourra avoir un comportement erroné. Il faut toutefois noter que des avis contradictoires d'experts sont parfois enrichissants, et qu'il peut donc être intéressant de conserver ces formes de conflits, surtout avec les systèmes coopératifs.
Consistance	Vérifier que les exigences exprimées sont compatibles entre elles Cohérentes – Elles ne doivent pas contredire d'autres exigences établies, ni être contredites par d'autres exigences. De plus, elle doit, d'un énoncé d'exigence au suivant, utiliser des termes et un langage qui signifie la même chose.
Non ambiguës	la non ambiguïté exige notamment une grande précision dans l'utilisation des termes introduits (si possible définir les termes dans un glossaire faisant partie du cahier des charges); Elles doivent être susceptibles de n'avoir qu'une seule interprétation.
Nécessaires	Elles doivent porter sur des éléments nécessaires, c'est-à-dire des éléments importants du système que d'autres composants du système ne pourraient pas compenser.
Faisable	réalisables avec le budget, les savoir-faire, les technologies matérielle et logicielle.

TABLE X. PROPRIETE DES EXIGENCES A VERIFIER ET VALIDE

## V. LES TECHNIQUES DE VALIDATION DES EXIGENCES

La plupart des méthodes ou pratiques existantes visant à identifier et recueillir les besoins. En raison de délais et autres considérations, la validation est effectuée de façon informelle, soit sur une base *ad-hoc* ou simplement comme un examen (*peer review*) [86].

Différentes organisations sont possibles, de la simple relecture personnelle à la révision très organisée et formalisée.

Il existe différents types de techniques de validation des exigences disponibles dans la littérature, certains d'entre eux sont les suivants :

### 1. Pré-révision des exigences :

Les pré-révisions des exigences sont utilisées pour économiser les coûts et les délais de révision de toutes les exigences, alors que la révision complète implique des personnes d'origines différentes et il vaut mieux gagner du temps de retravailler. Les erreurs qui peuvent être facilement détectées sans révision complètes sont découvertes dans la pré-révision, par exemple pour des fautes d'orthographe et de la non-conformité aux standards d'organisation, etc. [1]. En pré-révision une personne est consacrée à la vérification des documents pour les erreurs typographiques, les exigences manquant et non-conformité aux standards. Les documents déjà révisés sont envoyés aux équipes de révision. Cela coûterait cher si lors des réunions de la révision, tout le monde passerait du temps à vérifier les standards ou les erreurs typographiques. Donc, les pré-révisions sont recommandées avant d'aller dans les réunions des révisions des documents des exigences pour validation.

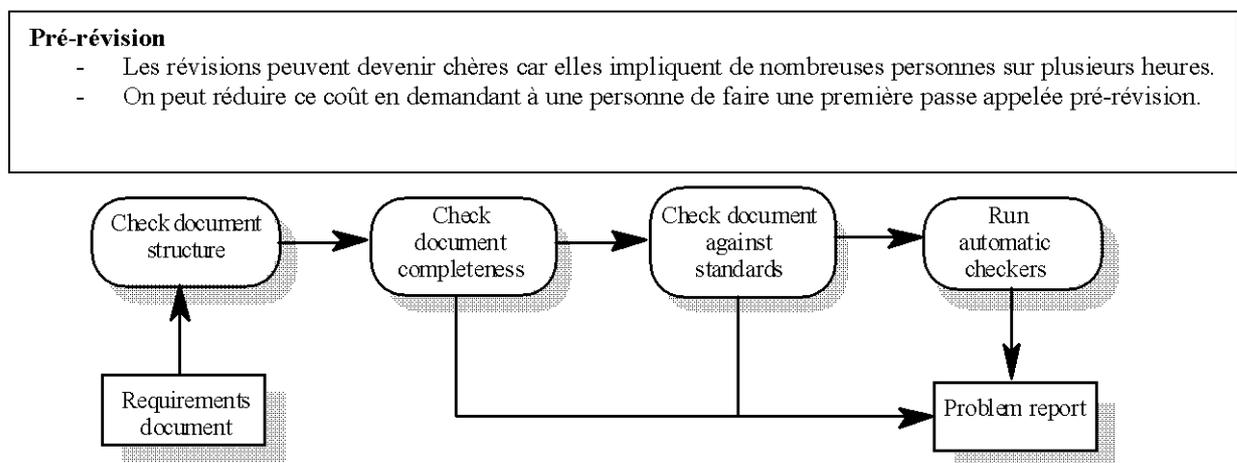


Fig.73. Contrôle de pré-révision [1]

## 2. La révision des exigences :

La révision des exigences est une technique utilisée pour valider les exigences par un groupe de personnes. Il s'agit d'un processus formel qui implique les lecteurs des deux côtés, clients et développeurs. La révision aide les clients et les développeurs à résoudre les problèmes au début de Cycle de vie. Le temps passé au cours des révisions des exigences, se rattrape en minimisant les changements et les modifications dans le logiciel. Les organisations qui ont habituellement des équipes de la révision indépendantes produisent des systèmes de qualité [39].

Il n'existe aucun processus de révision général pour les activités de développement logiciel. Les organisations doivent concevoir leurs processus de révision en fonction de besoins, clients et /ou marché [39]. C'est l'une des techniques les plus largement utilisées pour valider les exigences. Le processus de révision des exigences comprend les étapes suivantes, comme la montre la *figure 19*:

- a. *Plan de Révision*: l'équipe est sélectionnée, le temps et le lieu sont décidés pour la réunion de révision.
- b. *Distribution des documents*: les documents sont distribués aux membres de l'équipe de révision.
- c. *Préparation de la révision* : les individus lisent les documents pour relever les incohérences, les inconsistances, les conflits, les omissions et d'autres problèmes avant la réunion de révision.
- d. *Tenue d'une réunion de révision*: les commentaires individuels et les problèmes sont discutés, un ensemble d'actions pour résoudre le problème est fixé et accepté.
- e. *Suivi des Actions*: vérifie si les actions agréées sont en place ou pas.
- f. *Réviser le document*: le document final est révisé pour l'acceptation ou la ré-révision.

### Révision

Un groupe de personnes lit et analyse les exigences, cherche des problèmes potentiels, se rencontre pour discuter des problèmes et s'accorde sur une liste d'actions nécessaires pour les régler.

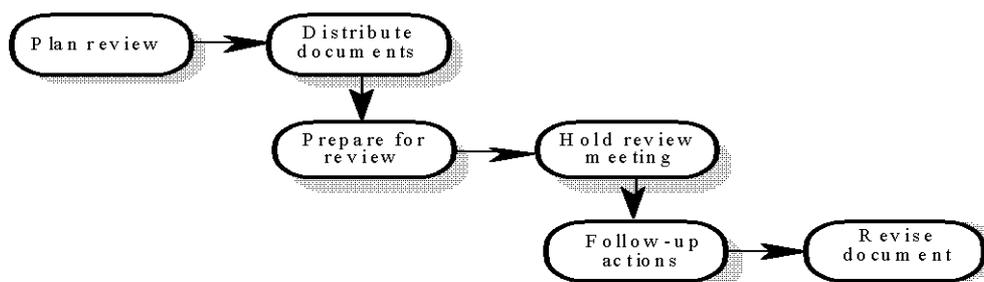


Fig.74. Processus de révision des exigences [1]

Le temps requis pour la révision dépend de la taille du document. Selon [1], l'effort de 50 personnes/heure, est requis par une équipe de 4 personnes pour inspecter les documents des exigences ayant trait à 400 exigences. Les exigences qui sont inconvenantes par erreur pendant l'élicitation des exigences devraient être clarifiées. Si l'information est manquante dans le document alors il est de la responsabilité des ingénieurs des exigences de se renseigner auprès des parties prenantes. Les exigences qui ne sont pas réalistes devraient être supprimées ou modifiées pour le développement à venir du système. Les conflits entre les exigences doivent être négociés avec les parties prenantes pour résoudre les problèmes exposés.

D'après Ian sommerville [40], l'équipe de révision vérifie généralement la complétude et la consistance des exigences. Elle devrait souligner les conflits, les omissions, les inconsistances et les erreurs dans les réunions de révision et devrait les noter correctement. Elles peuvent également vérifier les exigences pour:

- *Vérifiabilité*: L'exigence est testable ou pas?
- *Compréhensibilité*: Est-ce que l'exigence est comprise par l'utilisateur final.

- *Traçabilité*: Est-ce que l'exigence est traçable à sa source d'origine?
- *Adaptabilité*: Est-ce que l'exigence ne peut pas être modifiée sans effets à grande échelle sur les autres exigences.

Il est très important d'inviter les bonnes personnes pour réviser les exigences vis-a-vis de l'inconsistance, l'ambiguïté et d'autres problèmes. La première chose qui doit être tranchée est la bonne affectation des personnes aux exigences, et qui peuvent fournir des perspectives différentes, d'après Karl wiegiers [41], les intervenants suivants devraient être impliqués dans le processus de la révision des exigences:

- *Les Clients*: Ce sont les gens qui donnent les entrées des exigences ou les gens qui financent les projets.
- *Les Utilisateurs*: Ils sont directement ou indirectement impliqué avec le produit.
- *Les Analystes des exigences*: écrivent les exigences et communiquent avec les autres parties prenantes, en particulier l'équipe de développement.
- *Les Développeurs*: sont impliqués dans la conception, l'implémentation et la maintenance du produit.
- *Les rédacteurs de documentation*: sont impliqués dans la rédaction des manuels utilisateurs et la formation des utilisateurs du système.
- *Chef de projet*: envisage le projet et dirige l'équipe de développement jusqu'à la réussite du produit.
- *Personnel juridique*: sont responsables de la conformité des produits avec le règlement et toutes les lois pertinentes en vigueur.
- *Personnel de la fabrication*: C'est le groupe chargé de construire le produit contenant les logiciels.

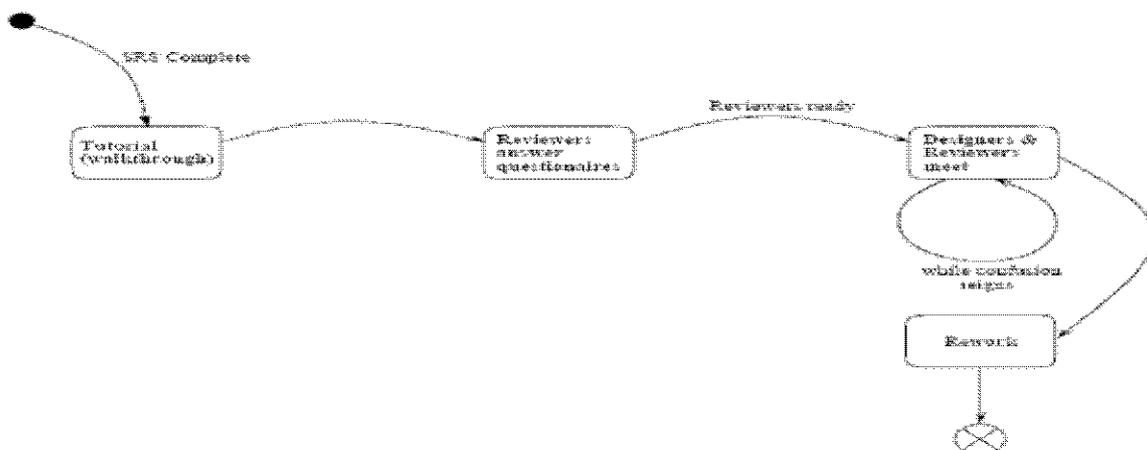


Fig.75. Révision Active [1]

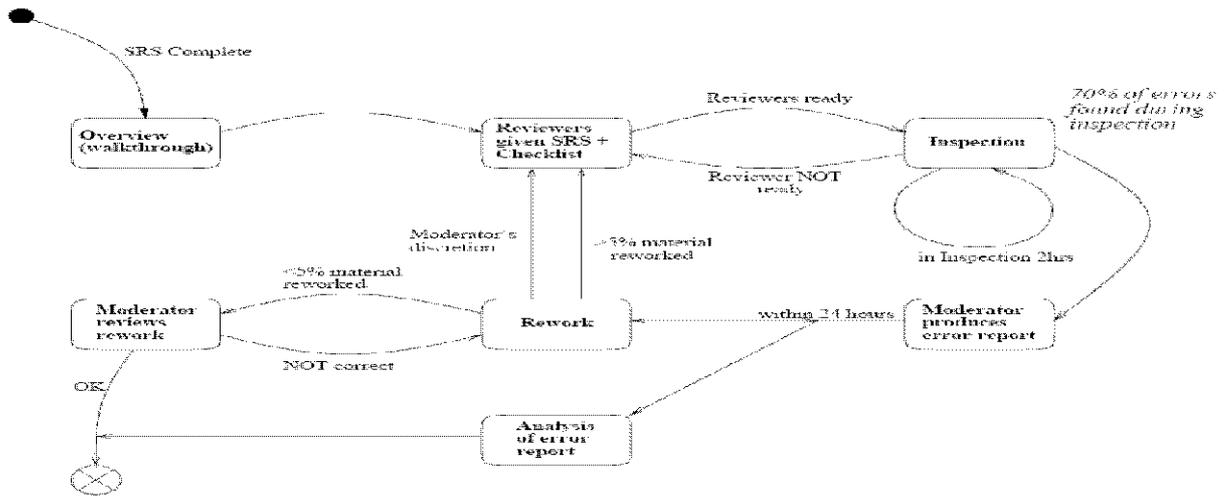


Fig.76. Inspection de Fagan [1]

## 2.1 Inspection des Exigences

L'inspection est semblable à la révision, mais ne lui est pas identiques. Fagan [42], a présenté l'inspection dans les années 1970 chez IBM, et c'est pourquoi celle ci est parfois appelée « *Inspection de Fagan* » voir figure 21. L'inspection des logiciels est un moyen très efficace pour trouver les défauts d'un produit logiciel, ce qui permet d'éliminer les défauts à un stade précoce. Selon Laitenberger et al. [43], l'inspection du logiciel peut détecter près de 50% à 90% des défauts. L'inspection est coûteuse en temps et en raison du grand nombre d'artefacts logiciels qui nécessite d'être analysées, recherchées et triées à cet effet Halling [44]. L'inspection des exigences n'est pas une pratique courante dans l'industrie en raison du coût qu'elle engendre. Le développement des logiciels d'un produit particulier est moins complexe dans les stades précoces, contrairement à celui opéré plus tard dans le cycle de développement, et il permet d'éviter beaucoup de travail à refaire si nous inspectons correctement et avec grande précision les exigences à temps Halling [44]. Les inspections sont surtout utilisées par les gens qui étudient l'état de l'art, et sélectionnent les meilleures pratiques. Le processus d'inspection consiste à suivre six étapes, qui sont:

- **Planification:** organiser des réunions lorsque tous les documents sont disponibles.
- **Qualification:** rendre le produit compréhensible pour les spectateurs.
- **La détection des défauts:** identifier les défauts dans les artefacts logiciels.
- **La collecte des défauts:** les défauts sont collectées et documentées, des décisions sont prises sur ces défauts pour décider d'une nouvelle inspection du logiciel ou pas.
- **La correction des défauts:** les défauts sont corrigés sur la base des réactions (*feedback*).
- **Répéter:** tous les défauts signalés sont retirés des artefacts logiciels.

Les rôles et responsabilités dans l'inspection des exigences sont exprimés comme suit:

- **Modérateur:** Il est responsable de la sélection de l'équipe et il / elle veille à ce qu'elle effectue ses tâches conformément à chacune et à toutes les phases. C'est la personne clé dans le processus d'inspection.
- **Inspecteur:** C'est la personne chargée de trouver des défauts.
- **Organisateur:** Il planifie toutes les activités au sein ou en dehors du projet.
- **Producteur:** C'est une personne qui a développé des artefacts logiciels et elle est désignée pour expliquer les artefacts logiciels à tous les membres de l'équipe.
- **Recorder:** est chargé de noter la liste de toutes les fautes dans la liste des défauts lors de la réunion d'inspection. C'est une position optionnelle.
- **Lecteur:** est chargé de présenter tout le matériel. Il / elle doit expliquer et interpréter tout le matériel.

### 2.1.1 Inspection Dirigée par les Cas de Test

La technique d'inspection Dirigée par Cas Test (Driven case-test DCT) a été proposée par Gorschek et al. [45]. Cette technique est utilisée pour éliminer les défauts avant que le projet ne commence (connu comme phase de pré-projet) et pendant le projet. Le but de cette technique est de s'assurer que les exigences sont assez bonnes pour le produit et que la planification des activités l'est aussi. Le but de cette technique de validation est de minimiser le coût et rendre l'inspection des exigences effective Nina et al [46].

L'inspection TCD est bénéfique pour la minimisation des ressources et la réutilisation des artefacts logiciels et personnels. Elle se compose de trois étapes, qui sont entreprises dans la phase pré-projet. Ces étapes sont:

- **Étape 1:** Le chef de projet sélectionne et révisé les exigences devant être incluses dans la spécification initiale, et certaines exigences sont également rejetées. Ceci est principalement un processus ad-hoc, les exigences sélectionnées connues tels que des exigences formulées ou spécifiées.
- **Étape 2:** D'habitude, les deux personnes impliquées dans l'inspection TCD, sont chef de projet et le testeur. Ces deux personnes sont déjà formées et possèdent une expertise dans leur domaine, ce sera là un moyen de minimiser le coût qui aurait dû être consacré à la formation du personnel. L'inspection des exigences est effectuée dans cette étape pour créer le cas de test de ces exigences. Les exigences qui ont une priorité plus faible devraient être supprimées dans cette étape.
- **Étape 3:** La planification du projet et les priorités des exigences sont faites dans cette étape. Les exigences qui ne sont pas retenues, sont rejetées.

Après la fin de l'inspection dirigée par les tests, les exigences sont sélectionnées pour implémentation.

### 2.1.2 La méthode d'inspection N-fold

Martin, J et all la présente dans [72] comme une technique où de multiples inspections sur le même artefact sont menées en parallèle, par un certain nombre (N), d'équipes. L'amélioration des performances de l'inspection peut être basée sur l'hypothèse selon laquelle une seule équipe d'inspection permet de détecter un sous-ensemble seulement du nombre total de défauts et que plusieurs équipes d'inspection permettent par contre de détecter (plus) de défauts.

### 2.1.3 La matrice de traçabilité

La matrice de traçabilité consiste en une comparaison entre les objectifs demandés et les exigences système Duran et all [85]. Une correspondance est établie entre les objectifs et la façon dont ils sont couverts par chacune des exigences. De cette façon, les inconsistances objectives et celles qui ne sont pas couvertes seront détectées.

## 2.2 Les Techniques de Lecture.

Les Techniques de lecture définissent une série d'étapes pour réviser / inspecter les artefacts logiciels. Dans les techniques de lecture, des méthodes spécifiques pour montrer la façon de lire et ce qu'il faut regarder dans l'Artefact Logiciel, sont communiquées au lecteur. Il existe plusieurs techniques de lecture qui sont disponibles dans la littérature, par exemple: celles qui sont basé ad-hoc, basé liste de contrôle, basé perspective, et celles basé défaut. Les lectures Ad-hoc et la liste de contrôle sont les plus répandues dans l'industrie pour la détection de défauts Gilb et all [47].

### 2.2.1 La Technique de lecture Basé Ad-Hoc

Les Lectures basée Ad-hoc offrent très peu de soutien. Dans cette technique de lecture, les artefacts sont donnés aux réviseurs / inspecteurs pour révision /inspection sans aucune ligne directrice. La détection de défaut est en fonction de la connaissance et de l'expérience du réviseur / inspecteur O.Laitenberger [48].

### 2.2.2 La Technique de lecture Basé liste de contrôle « Check-List »

Les Techniques de lecture basée Liste de contrôle est l'une des techniques couramment utilisée. D'après Laitenberger [49], une check-list est utilisée comme une technique de lecture standard dans la plupart des organisations. Elle contient un ensemble de questions qui guide le réviseur ou l'inspecteur lors de révision / inspection. Elle comprend des questions qui sont liées à la qualité des exigences O.Laitenberger [48].

Check-list Questions	Quality Attribute
Is each requirements is easily Identified?	Traceability, conformance to the standard
Are specified terms are defined in the glossary	Understandability
Do individual requirements use the same term in	Ambiguity

different ways?	
If a requirement makes references to some other facilities; are these described elsewhere in the document?	Completeness
Are related requirements group together?	Organization
Are there any contradictions in the requirement?	Redundancy
Do you have to examine other requirements to understand what it means?	Completeness

TABLE XI. QUESTION DE CHECK-LIST [1]

Un exemple de questions en fonction de la Check-list peut être vu dans le tableau 1, qui sont donnés par Kotonya & al. [1]. Selon Laitenberger et al. [50], l'enquête sur les faiblesses de la techniques de lecture basée check-list sont les suivants:

- Les questions de la Check-list sont surtout générales
- Les instructions sur comment la Check-list est faite ou opérée, sont souvent absentes.
- Les questions sont conçues en fonction des défauts détectés précédemment et appartiennent à un type particulier de défaut.

### 2.2.3 Les Techniques de Lecture basé Perspective :

En lecture-basé perspective, les réviseurs sont affectés d'une perspective spécifique, lors de la révision / inspecter d'un artefact logiciel. Quand un réviseur a perspective spécifique, il n'est plus utile pour détecter certains types de défauts. Dans cette technique de révision il est plus ciblé parce qu'il / elle essayer de trouver le type spécifique de fautes techniques alors que dans d'autres ils sont chargés de détecter tous les types de défauts. Il ya un autre avantage de la lecture-basé perspective est qu'ils travaillent de manière plus structurée et leurs lectures sont actives T.Thelin [51].

Les lectures basées perspective représentent les clients, testeurs et les concepteurs. A partir de leurs trois perspectives différentes, les réviseurs / inspecteurs sont invités à utiliser l'approche de lecture basé scénario pour lire les exigences Jeffrey [52]. Tout en répondant aux questions des inspecteurs en prenant des notes de défauts potentiels découverts qui pourraient affecter le jeu des responsabilités des parties prenantes Paulo et all [53]. La raison principale derrière la lecture-basé perspective est que si plusieurs lecteurs ont différentes perspectives du document, cela aboutira à un document qui représente tous leurs exigences.

### 3. Prototypage des Exigences

Le prototypage est principalement utilisé pour donner un aperçu des exigences imposées ou implémentées dans le système proposé. Il est très difficile pour l'utilisateur de comprendre comment les exigences sont écrites avant d'être transformées en une forme exécutable du système [1]. Selon

Sommerville [40], le but principal de prototypage des exigences est de comprendre les exigences du système et il est principalement utilisé pour:

- **élicitation des exigences:** dans cette phase, le prototypage permet aux utilisateurs de comprendre comment le système va supporter leur travail.

- **validation des exigences :** Dans cette phase, le prototypage permet d'identifier les erreurs pour ne pas les avoir dans le document d'exigences.

Le prototypage se présente sous diverses formes, de croquis ou de photos jusqu'à un niveau, plus haut avec les langages de la quatrième génération. Principalement les organisations génèrent plusieurs prototypes, par exemple: d'abord ils créent des croquis sur papier ou sur un tableau blanc et ensuite ils conçoivent un prototype opérationnel en utilisant un langage de haut niveau. Selon Davis [54], il existe deux types de prototypes, comme le montre la figure 22, qui est le prototype jetable et évolutif:

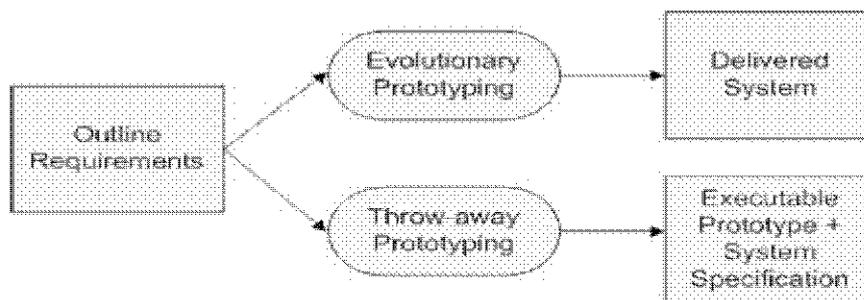


Fig.77. Approche de Prototypage

### 3.1 Prototypage rapide

Le prototype jetable est un type, dans lequel seulement une fonctionnalité limitée est disponible et il sert à développer les exigences mal comprises avant de commencer le développement réel [54]. Selon Sommerville [40], le but de prototype jetable est de clarifier les exigences et d'évaluer les risques liés aux processus potentiels. Il ne devrait pas être considéré comme un système final. Il est développé à partir de la phase initiale jusqu'à la phase expérimentation où il sera détruit. En prototypage jetable, nous développons des exigences qui sont difficiles à comprendre pour les clients. Les exigences, qui sont claires, ne doivent pas être développées par ce prototype. Si le client et le développeur sont d'accord sur les exigences, alors le prototype devrait être abandonné Siddiqi et al [55].

### 3.2 prototypage évolutif

Le prototype évolutif est un type de prototypage dans lequel nous développons les exigences qui peuvent être faciles à comprendre par les clients et qui ont une haute priorité. L'objectif du prototypage évolutif est de livrer un système réalisable pour l'utilisateur final, et il devrait présenter tous les attributs de la qualité du produit final davis [54]. Il est plus adapté lorsque les développeurs ont un contact minimal avec les intervenants. Il ya certains facteurs de qualité qui touchent généralement le prototypage évolutif, comme la performance, la qualité du design et de maintenabilité. Ces attributs vont souffrir si des mesures ne sont pas correctement suivies S.Gordon [56]. Le prototype Evolutif doit être utilisé lorsqu'on ne peut pas développer la spécification en avance, par exemple les systèmes d'interface utilisateur et les systèmes d'IA (intelligence artificielle) Ian sommerville [40].

#### 4. Validation Des Exigences Basée-Modèle

La formulation, la structuration et la modélisation des exigences peuvent être guidées par une méthode. C'est une approche systématique pour documenter, analyser et valider les exigences système. En outre, une notation qui donne des moyens d'expression des exigences est également associée à la méthode [1]. Il n'existe pas de méthode d'exigence unique idéale disponible, mais une variété de techniques de modélisation qui sont utilisées pour formuler les exigences système. Selon Kotonya et al. [1], plusieurs techniques de modélisation sont utilisées pour la validation des exigences, certaines d'entre elles sont les suivantes:

##### 4.1 Le Modèle flux de données

L'approche flux de données utilise le diagramme de flux de donnée (DFD), pour démontrer graphiquement la façon dont les données sont traitées à des stades différents du système. Tout en représentant des entités externes, les processus, les flux de données, et le dictionnaire (stockage) des données.

Selon Maciaszek et al [57], la colonne vertébrale de DFD est une décomposition fonctionnelle car elle peut être vue suivant la figure 23. La décomposition fonctionnelle est l'activité top-down, qui est lancée à partir d'un diagramme de contexte et qui se termine par un module de spécification.

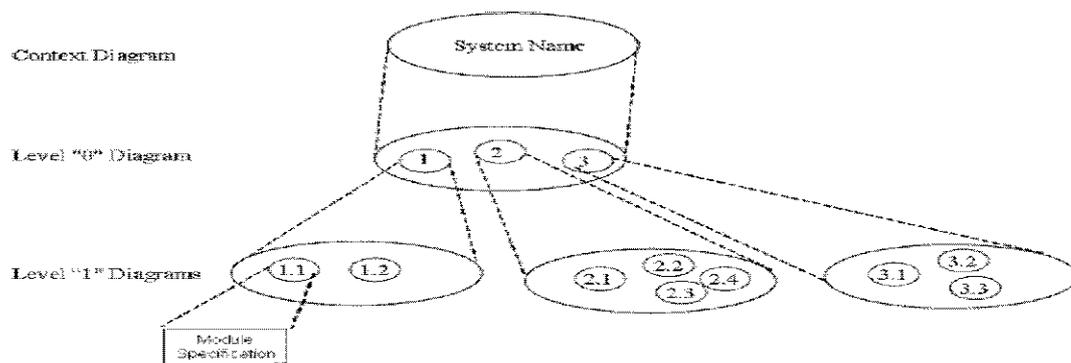


Fig.78. Décomposition Fonctionnelle du DFD [57]

#### 4.2 Le Modèle compositionnel

Le modèle entité-relation (ER) peut être utilisé pour illustrer comment les entités sont composées d'autres entités [58]. L'ER est principalement constitué de quatre composants graphiques différents, qui sont: entité, relation, cardinalité, et attributs.

#### 4.3 Le Modèle de Classification

Les diagrammes d'objets Leszek et al [57] peuvent être utilisés pour démontrer comment les entités ont des attributs communs, Selon Kotonya et al. [1], la modélisation orientée objet, selon notamment les concepts fondamentaux: d'objets et classes, méthodes, messages, l'encapsulation et l'héritage. Elle peut définir et analyser des situations réelles et abstraites.

#### 4.4 Modèles de Stimulus Réponse

Le Diagramme d'état transition peut être utilisé pour démontrer comment le système réagit aux événements internes et externes.

#### 4.5 Modèles de Processus

Les modèles de processus sont les activités principales et délivrables qui sont impliquées dans l'exécution de certains processus.

#### 4.6 Modèles de simulation

Selon Maria [59], les modèles de simulation sont décrits comme suit:

- La démonstration du développement et du fonctionnement de certains systèmes d'intérêt est connue comme un modèle. L'opération d'un modèle du système est connue comme *simulation*.
- La performance d'un système existant ou proposé peut être évaluée en utilisant la *simulation*, sous différentes configurations d'intérêt et sur des longues périodes de temps
- La *simulation* peut être utilisée pour réduire les risques liés aux échecs des spécifications d'exigences, avant la modification d'un système existant ou le développement de nouveau système, pour éliminer les défilés imprévus, pour obtenir l'utilisation exacte des ressources et l'optimisation des performances du système.

Le principal objectif de la validation basé-modèle est de montrer la façon dont chaque partie du modèle est auto-consistant, aussi bien dans sa consistance interne qu'externe. La validation du modèle reflète précisément les exigences système des parties prenantes. L'implication des parties prenantes dans le processus de validation basé-modèle est très importante. Il est difficile pour les personnes non techniciennes de comprendre les diagrammes de flux de données, les diagrammes d'événements ou les

diagrammes d'objet ; c'est pourquoi il est suggéré de travailler avec des descriptions en langage naturel, même lorsque les modèles disent exactement la même chose à l'aide de diagrammes [1]. De cette manière, il sera facile de détecter les erreurs, les inconsistances et l'incomplétude dans les modèles. Les modèles ont une sémantique bien plus complète et il est facile de prouver leurs consistances mathématiquement.

## **5. Validation Des Exigences Basée-Test**

Selon Liu et al. [60], les tests des exigences impliqueront principalement la définition adéquate et l'intégrité de données, la consistance, la non ambiguïté, et la testabilité des exigences. La validation des exigences basé-test est une approche systématique pour la conception de cas de test afin d'examiner chaque exigence et d'en tirer une série de tests. Elle peut définir un ou plusieurs tests qui peuvent être exécutés lorsque le système sera développé. Il est toujours souhaitable que toutes les exigences puissent être testables.

Le test proprement dit est exécuté après le développement du système. Il s'agit d'un moyen efficace d'exposer des problèmes tels que : l'incomplétude, l'inconsistance et l'ambiguïté des exigences en suggérant les tests possibles [1]. S'il est difficile de tirer des cas de test pour la spécification des exigences système alors cela signifie qu'il y a un certain type de problème avec l'exigence [1, 60]. Cela pourrait être dû à un manque dans l'information ou dans sa description qui ne peut pas donner une image claire. Ainsi chacune des exigences fonctionnelles dans le document de spécification des exigences doit être soigneusement analysé, et des tests sont définis pour ce dernier afin de vérifier objectivement si le système satisfait aux exigences ou pas.

Selon Sommerville [40], les exigences devraient être rédigées de telle façon que la conception des tests devrait être facile pour chaque cas de l'exigence.

Donc, Comme le résultat des développeurs peut vérifier si les exigences sont implémentables ou non. Le but de suggérer des cas de test pour les exigences est de les valider plutôt que de valider le système. Pendant la définition des cas de tests pour les exigences, Certaines considérations importantes doivent être gardées à l'esprit [1], qui sont:

i. L'utilisation de scénario des exigences qui définit le contexte dans lequel le test doit être appliqué.

ii. il est nécessaire de s'assurer que l'exigence contient suffisamment d'informations pour que le cas de test puisse être défini.

✓ *Si l'information n'est pas disponible dans cette exigence spécifique alors d'autres exigences liées devraient être étudiées pour obtenir cette information.*

✓ *En cas d'obtention d'informations à partir d'autres exigences, les informations recherchées doivent être enregistrées.*

✓ *Ces exigences peuvent dépendre d'autres, il est donc nécessaire de les enregistrer.*

iii. Toujours vérifier cette exigence en utilisant un ou plusieurs cas de test. Parallèlement à cela, il est également possible qu'il pourrait y avoir plus d'une exigence décrite dans une description d'exigence unique.

iv. les cas de test devraient être évident si l'exigence est déclarée de nouveau.

Kotonya et al. [1] ont proposé une forme d'enregistrement de test pour chaque exigence qui est testé, devant être conçue et remplie avec les informations suivantes:

*Identificateur de l'exigence*

*1er. Exigences reliées*

*2e. Description du test*

*3e. Problèmes d'Exigence*

*4e. Commentaires et recommandation*

## **6. La validation des exigences orientée-point de vue**

La définition des exigences orienté-point de vue est une méthode qui est essentiellement prévue pour l'identification et la spécification des systèmes interactifs (basée sur les points de vue qui met l'accent sur les demandes des utilisateurs et les préoccupations de l'organisation), mais il est également possible de l'utiliser pour spécifier d'autres classes du système [1]. Selon Sommerville et al. [61], l'approche bases-point de vue en ingénierie des exigences n'identifie qu'une seule perspective des exigences du système et ne peut pas couvrir l'ensemble des exigences réunies. Il devrait y avoir un nombre différent de points de vue pour collecter et organiser toutes les exigences. Leite et al. [2] suggèrent cette approche, qui peut être utilisée très tôt pour la validation des exigences.

L'objectif de la validation des exigences basé point de vue est d'identifier et de classer les problèmes liés à l'exactitude, la complétude et la consistance [1, 2]. Une encapsulation des informations partielles sur les exigences d'un système peut être appelée comme un point de vue [61]. Le processus de résolution du point de vue concerne l'identification des différences entre deux points de vue, de classer et d'évaluer ces écarts, pour l'intégration des différentes solutions en une représentation unique [2].

Une déclaration explicite de la perspective adoptée par un point de vue est connue comme l'accent du point de vue [61]. La figure 24, illustre un exemple de différents points de vue qui se chevauchent, ce qui est important pour découvrir d'éventuels conflits entre les exigences. Selon Leite et al. [2], cette approche comporte certaines notions qui sont nécessaires pour comprendre la validation des exigences orienté-point de vue. Ces concepts sont:

- **L'univers du discours** : C'est le contexte général pour développer le logiciel. Parallèlement à cela, il contient toutes les sources d'information, y compris toutes les personnes associées au logiciel.
- Les gens sont connus comme **des acteurs** dans l'univers du discours.

- Un **point de vue** est connu comme une position ou place mentale d'un individu lors de l'examen ou l'observation de l'univers du discours.
- Une **perspective** est un ensemble de faits observés et modélisés selon un aspect particulier de modélisation et de point de vue
- La relation entre les entités et les attributs est connu comme un **fait**.
- Une **vue** est une intégration des perspectives et des hiérarchies.
- Il existe deux types de **hiérarchie** de concepts qui sont utilisés, connu comme « est-un » et « partie-de » des hiérarchies dans l'univers du discours.

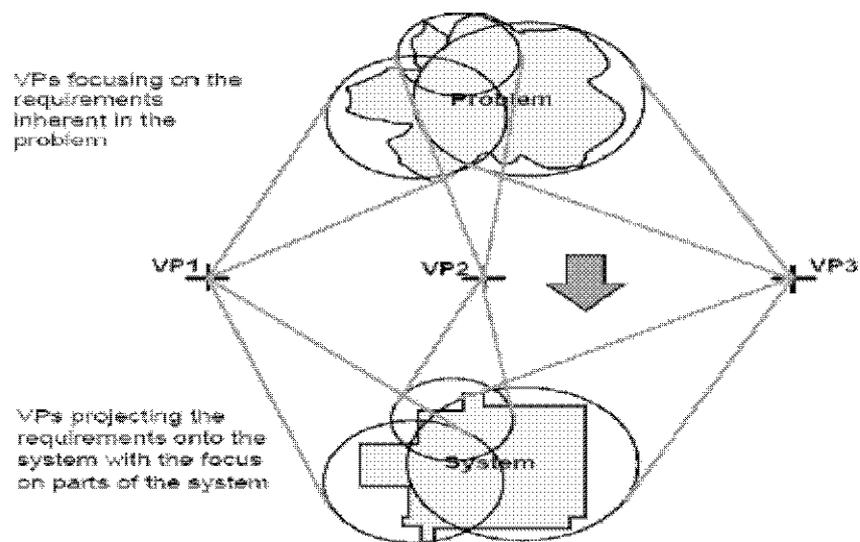


Fig.79. Points de vue sur un problème

## VI. APPROCHE POUR VALIDATION ET VERIFICATION DES EXIGENCES

### 1. Preuve mathématique et logique pour la vérification et la validation des exigences

La transformation mathématique des règles peut être appliquée à une série d'assertions exprimées en un langage de spécification mathématique précis pour prouver les propriétés désirées de l'ensemble d'assertions.

Puisque le logiciel est une description formelle, l'analyse de son comportement est justifiable par raisonnement formel. La logique fournit un véhicule pour effectuer une telle analyse selon Abramsky et all [87]. En IE, la logique peut être utilisée pour améliorer la rigueur de l'analyse effectuée, et de faire la démarche du raisonnement explicite. Des logiques différentes peuvent être utilisées pour exprimer les différents aspects d'un système. Par exemple, *la logique temporelle* peut être utilisée pour décrire des informations de temps, *la logique déontique* pour décrire les autorisations et les obligations, et la *logique linéaire* pour décrire les ressources et leur utilisation. Un autre avantage des langages de spécification fondés sur une logique, c'est qu'ils sont potentiellement susceptibles d'automatisation du raisonnement et de l'analyse. Dans le contexte d'ingénierie des systèmes, la compréhension et l'application de la théorie et la pratique des systèmes est également pertinente pour IE d'après Stevens [88]. Cela comprend des travaux sur les caractéristiques de systèmes, l'identification de leurs limites et la gestion de leur cycle de développement et de vie [89, 90].

### Discussion

La preuve mathématique et logique des exigences donne seulement un résultat vrai ou faux, et en réalité tous les besoins de client sont flous et leur formulation en assertion mathématique ou clause logique ne marche que sur les expressions bien exprimées alors que les phases mal exprimées présentent la majorité.

## 2. Validation des exigences par traitement de langage naturel

Wilson et al [73] ont développé un outil l'évaluation des spécifications des besoins qui sont en langage naturel dès le début du cycle de vie. L'outil recherche dans le document neuf indicateurs de qualité, qui sont identifiés par la recherche des mots fréquemment utilisés, des phrases et structures du document sélectionné qui sont liées aux attributs de qualité IEEE [75] et pourraient être facilement identifiées et comptées par un logiciel.

Les indicateurs de qualité de la spécification individuels sont:

- **Impératifs:** les mots et les phrases qui commandent que quelque chose doit être fourni. (Devez, nécessaire, veuillez, etc.)
- **Continuations :** les phrases qui suivent un impératif et introduisent les spécifications des exigences à un niveau inférieur. (Ci-dessous, comme suit, suivantes, en particulier, etc.) .
- **Les directives:** des mots et des phrases qui illustrent l'information au sein de document des exigences. (Figure, table, par exemple, note)
- **Options:** les mots qui donnent latitude au développeur en satisfaisant les spécifications des exigences qui les contiennent. (Qui peut, mais, optionnel)

- **Faible Phrases:** des clauses qui sont de nature à provoquer l'incertitude et laisser la place aux multiples interprétations. (Adéquate, être capable, normal, tel que minimum, si praticable, mais non limité à, etc.).

*Les indicateurs de la qualité du document d'exigences sont:*

- **Taille:** se rapporte à la taille du document de spécification des exigences par le comptage du nombre total de lignes du texte, les impératifs, les sujets de déclarations des spécifications et les paragraphes.

- **Structure du texte:** indique le nombre d'identificateurs de déclaration constatés à chaque niveau hiérarchique du document.

- **Profondeur de Spécification :** rapporte le nombre d'impératifs trouvés dans chacun des niveaux de la structure du document de texte.

- **Statistiques Lisibilité:** mesurer la facilité d'un adulte qui sait lire et comprendre le document exigences.

Les rapports produits par l'outil sont utilisés pour identifier l'état et la structure des zones du document de la spécification des exigences qui doivent être améliorés.

Semblable à celle de Wilson et al, L'approche Fabbrini et al [74] vérifie également des exigences en langage naturel basé sur un modèle de qualité spéciale pour les exigences logicielles [74].

### **Discussion**

Le traitement du langage naturel est encore moins mur pour qu'il soit utilisé dans la première phase du cycle de vie, il est difficile à évaluer avec la langue naturelle car elle est (Verbeuse, informelle, ambiguë, incomplète...) et donc sa vraie performance vient suite à la création d'une ontologie afin de surmonter le problème de vérification et de validation de l'aspect sémantiques, ce qui reste de la paraphrase est loin d'apporter de l'aide aux analystes ou même aux concepteurs et certainement pas au client.

### **3. Une méthode automatisée avec XML pour la Vérification des exigences**

Duran et al. [76] présentent une approche automatisée pour la vérification d'exigences logicielles. Cette approche est basée sur la représentation des exigences logicielles en langage XML et l'utilisation de XSLT pour vérifier d'une manière automatique les qualités des propriétés par l'utilisation d'un outil expérimental de gestion des exigences appelé REM .L'entrée se compose d'un document d'exigence en langage naturel et une spécification écrite dans un sous-ensemble d'UML. Les exigences sont stockées au format XML et XSLT qui est utilisé comme un langage de vérification d'exigence pour vérifier automatiquement l'ambiguïté, la complétude et la traçabilité des exigences.

### **Discussion**

Le gros inconvénient du XML est sa verbosité. Pour chaque nœud dans la hiérarchie, il y a une balise de début et une balise de fin contenant le nom de la balise. Le fichier est moins compact que s'il avait été écrit en binaire. Cela signifie qu'il serait difficile de mettre à jour les documents (exigences et/ou spécification), où des modifications sont fréquentes dès l'apparition d'un nouveau besoin.

#### **4. Validation des exigences d'une spécification en UML**

Koesters et al. [77] présentent une méthodologie pour la vérification et la validation des spécifications avec UML par la transition à partir de *cas d'utilisation* à un *diagramme de classe* et la vérification de diagramme de classe par rapport au diagramme cas d'utilisation. Depuis le diagramme de classe offre plus de sémantique que le diagramme cas d'utilisation, le diagramme cas d'utilisation est raffiné pour atteindre des spécifications plus précises. Cela se fait grâce au *diagramme d'activité*: le pont entre les diagrammes cas d'utilisation et de Classe. A travers cette approche, il peut être vérifié que l'instance d'une classe dans un diagramme de classe fait formuler les cas d'utilisation.

#### **Discussion**

UML est un langage de modélisation unifiée, ce qui fait de lui un standard plus au moins compréhensible à la fois par les utilisateurs et les analystes, ainsi l'existence des outils CASE qui permettent la construction et la validation de ces diagramme de façon automatique à partir d'une description textuelle valide et une bonne compréhension. Seulement il n'existe aucune information sur la cohérence et les autres caractéristiques d'aspect sémantique qui facilitent une compréhension des sens, Il est obligatoire qu'il y ait un autre cycle pour la sémantique des diagrammes et par conséquent des exigences.

#### **5. Approche empirique pour la Validation des exigences**

Nuseibeh [78] a utilisé l'expression « Il s'agit de comparer le problème de la validation des exigences avec le problème de la validation des connaissances scientifiques », certainement en référence au cycle (Expérience, Observation, Analyses & résultats). Cette approche empirique permet de *procéder à des expériences et à l'analyse des résultats*, où des statistiques sont étudiées après les réunions et des analyses des expériences en découlent. Lulu He et all [79] décrivent une expérience menée dans le module d'ingénierie des exigences en post-graduation dans le but de couvrir le déficit d'apprentissage dans l'enseignement de la validation des exigences. Elle fournit aux étudiants un monde d'expérience réel. L'expérience a fait usage de la méthode d'inspection N-fold, dans laquelle plusieurs équipes d'étudiants inspectent les mêmes cahiers de charge puis se réunissent pour discuter leurs conclusions. Le même auteur a montré dans Lulu He et all [80] l'efficacité de toutes les réunions de l'équipe avec l'inspection N-fold à travers des statistiques des pertes et des gains dus à l'inspection par N groupe avec des compétences

différentes (Aussi bien des étudiants qui n'ont jamais assisté à un vrai projet à ceux ayant un nombre important d'années d'expérience dans le domaine d'industrie). Ces groupes sont divisés en trois sous ensembles de façon aléatoire (testeur, développeur, analyste) et par la suite, des statistiques sont menées pour comparer deux techniques de validation des exigences (check liste, lecture basé-perspective). Leurs objectifs sont la multiplication de l'opération de lecture afin de trouver plus de défauts pour comparer ces techniques de lecture. Par la suite, des statistiques simples sont calculé par moyen variance et écart type. C'est alors à l'équipe d'analyser les résultats des expériences pour en tirer des commentaires et de prendre ensuite des décisions sur ces anomalies.

### **Discussion**

L'expérience et l'analyse des résultats comme démarche scientifique de validation des exigences est la plus fructueuse à notre avis en terme de compétence demandée chez l'équipe d'analyse car il y a un niveau d'apprentissage évolutif chez les membres de l'équipe IE, dû à l'expérience requise dans leurs communications à l'intérieur des entreprises et à celle résultant de la discussion en groupe. Cependant, l'absence de méthode et/ou d'outils pour analyser ces résultats, telles que les statistiques pour en tirer des informations sur la complétude, la cohérence, ...des exigences, fait d'elle une approche incomplète et nécessite donc un autre cycle pour que les exigences soient validées par de client ;

À travers l'utilisation de méthode manuelle, ces techniques qui valident seulement la bonne interprétation de l'information. La vérification d'inconsistances ou la découverte d'information manquante exige des méthodes plus sophistiquées et Ils fournissent seulement une révision partielle de l'information et ces résultats.

## **6. L'Approche système expert pour Validation des exigences**

Cette catégorie d'approches comporte un certain nombre de systèmes automatisés, Des outils (CASE) de prototypes, qui aident à la validation des exigences. Ce qui justifie la qualification de ces outils "systèmes experts", c'est la connaissance de certains aspects qui symbolisent le processus d'ingénierie des exigences. Cela peut être soit la méthode de connaissance, c'est-à-dire la connaissance de la façon d'appliquer une méthode pour l'ingénierie des exigences (par exemple, une analyse structurée voir [82]), ou connaissance du domaine, c'est-à-dire des connaissances sur le domaine que le logiciel est sensé modeler. Les approches Expert sont encore à l'état de prototype, mais ils sont certainement appelés à avoir un impact sur la fonctionnalité de la future génération commerciale des outils CASE.

En général, il existe trois modes dans lesquelles un outil de système expert peut agir dans le processus de validation, à savoir

- **l'expert** : à ce mode, l'outil est supposé assurer lui même le processus de la validation.

- **l'assistant** : à ce mode, il est demandé à l'outil d'aider l'analyste à valider certains aspects des exigences, c'est-à-dire par la vérification de certains types d'incohérences dans le modèle par exemple.
- **l'apprenti** : à ce mode, il est demandé à l'outil d'effectuer le niveau faible de l'activité de validation, c'est-à-dire garder une trace de tous les termes utilisés dans le modèle d'exigences dans le contrôle de définitions multiples, et ainsi de suite.

### Discussion

La génération actuelle des outils CASE fonctionnent au niveau de l'apprenti. Toutefois, la prochaine génération d'outils CASE devrait apporter des performances au niveau de l'assistant qui montre l'interaction entre un analyste humain et l'assistant de validation des exigences logiciel. Ce qui nécessite d'utiliser des outils de prototypage avec des interfaces plus conviviales et interactives afin que le client puisse intervenir lors de la validation finale.

## 7. Les méthodes formelles pour la Vérification et la Validation

Puisque le logiciel est une description formelle, l'analyse de son comportement est justiciable par raisonnement formel. Les méthodes formelles offrent un outil qui automatise la vérification de la complétude et la consistance des spécifications des exigences logiciel. Pendant la décennie passée, beaucoup de méthodes formelles (B [92], LOTOS, Z,...) ont été proposées mais seulement deux approches principales seront expliquées ici.

### a- Model-checking

Désigne une famille de techniques de *vérification automatique* des systèmes dynamiques (souvent d'origine informatique ou électronique). Il s'agit de vérifier algorithmiquement si un modèle donné, le système lui-même ou une abstraction du système, satisfait une spécification, souvent formulée en termes de logique temporelle.

On peut distinguer 2 aspects du *modele checking*: Il peut s'agir de démontrer qu'une certaine classe de propriétés, ou une certaine logique, est décidable, ou que sa décision appartient à une certaine classe de complexité. Il peut s'agir de rechercher des algorithmes efficaces sur des cas intéressants en pratique, de les implémenter selon Schneiderz [83], et de les appliquer à des problèmes réels.

### b- Les réseaux de pétri

José Reinaldo S et all [84], Soulignent que le début des phases de développement est critique pour la réussite des projets, en plus d'être très important pour orienter l'exécution et la décision. Ils ont proposé une validation des exigences avec une méthode appropriée à un cycle de vie représentée par les processus unifiés, et adressée à la spécification et la à la conception des systèmes d'information. La question clé de la proposition est l'établissement de modèle de circulation de l'information formelle, par la capture des

exigences élémentaires en réseau de pétri. La validation des exigences effectuées dans le filet, par simulation ou par l'analyse de propriété (ne conviennent qu'à de grands systèmes).

### Discussion

Les pratiques en ingénierie des exigences n'ont pas donné la place réelle aux méthodes formelles. Deux raisons principales expliquent pourquoi un tel échec s'est produit. Tout d'abord, les méthodes formelles ne favorisent pas le suivi (Cycle de vie du logiciel). La deuxième raison est que les méthodes formelles ont échoué à grande échelle [140]. Les modèles complexes nécessitent souvent une traduction vers des notations plus compréhensibles pour l'utilisateur et donc cela nécessite souvent un autre cycle de validation.

IE doit combler le fossé entre le monde informel des besoins des intervenants, et le monde formel de comportement des logiciels, la question clé de l'utilisation des méthodes formelles n'est pas de savoir quoi formaliser, mais quand formaliser Parnas [91].

### 8. Synthèse des approches

Bien que la littérature mentionne plusieurs approches pour la validation des exigences, elles peuvent toutes être classées selon le niveau de formalité de la spécification avec laquelle démarre le processus de validation en spécification (formelle, Semi-Formelle, Informelle), Aboutissant à une solution (Automatisé, Semi-automatique ou Manuelle) généralisées en deux niveaux d'évaluation : **un niveau interne**, cette activité est assurée par l'équipe d'IE, **niveau externe** faisons appel aux clients, ou Mixte qui fait intervenir le maître d'ouvrage guidé par le maître d'œuvre. Appliqué sur des exemples de systèmes qui peuvent être Critique (Embarqué, temps réel, etc.), Ordinaires de gestion.

Niveaux	Approches						
	Math & Logique	Langage Naturelle	XML	UML	Empirique	Systeme expert	Méthode Formelle
<b>Niveau de Formalité de la spécification</b> Formelle/ Semi-formelle / Informelle	Formel	Informel	Formel	Semi-formel	Informel	Informel	Formel
<b>Niveau D'automatisation</b> Automatisé / Semi-automatique/ Manuelle	Automatisé	Automatisé	Automatisé	Semi-automatique	Manuel	Automatisé	Automatisé
<b>Niveau d'évaluation</b> Vérification Interne / validation Externe / Mixte	Interne	Interne	Mixte	Mixte	Interne	Mixte	Interne
<b>Type de Système</b> Critique/Ordinaire	Critique	Ordinaire	Ordinaire	Ordinaire	Ordinaire	Ordinaire	Critique

TABLE XII. SYNTHÈSE DES APPROCHES DE VÉRIFICATION & VALIDATION

## VII. LA VALIDATION DES EXIGENCES DANS LES SYSTEMES COOPERATIVE

Les deux approches qui guident la validation des exigences de la méthode CREWS, sont à notre connaissance les seuls liens entre la validation des exigences et les systèmes coopératifs.

Après avoir débattu les deux approches CREWS dirigeant l'élicitation voir (chapitre I). La troisième approche fournit **une méthode et un outil logiciel pour générer des scénarios**, ensuite une révision de type Walkthrough<sup>3</sup> aura lieu. La validation se fait manuellement, et non pas de manière automatique, elle suit un processus par étapes pour la vérification qui est fait par le biais de ces scénarios, ainsi une détection semi-automatique des exigences manquantes et/ou incorrectes est faite à travers une analyse informatique de ces scénarios.

La quatrième approche du processus de validation des exigences dans Heymann [70] est réalisée à travers l'animation des scénarios tirés d'une spécification formelle du système. Ce-ci est compatible avec un langage de modélisation des exigences basées agent dit ALBERT, qui permet une inter-liaison facile dans la spécification des exigences. Des scripts de scénarios déclaratifs non-déterministes et des traces d'exécution de scénario déterministes sont la base de cette animation et par conséquent, la validation des exigences.

Maiden et al dans [81] propose une implémentation de la Validation par Frames ou Chaque frame de validation est représentée en cinq parties:

- Un **identificateur** : unique pour le frame;
- Le **pattern(s)** CREWS : est un frame opérationnalisés;
- Une situation** : qui précise les comportements souhaitables dans l'environnement;
- Une ou plusieurs exigence (s) qui spécifient la forme de la valeur sociotechnique du système;
- Les conséquences** des exigences (s) : qui présente le comportement souhaitable.

A chaque fois que l'algorithme de validation applique un frame, l'algorithme de recherche donne le scénario pour une combinaison unique événements / actions, les agents et les objets de types prédéfinis, qui est la signature de la trame, définies la situation de la trame. Lorsque l'algorithme détecte cette combinaison, il cherche un ou plusieurs exigences d'un type prédéfini et le contenu qui, selon le frame, doivent être présents dans le document pour faciliter le comportement souhaitable.

### Discussion

L'un des points forts des quatre approches CREWS, est qu'elles ont été conçues pour se compléter mutuellement. Comme résultat, CREWS propose une méthode cohérente d'ingénierie des exigences basée scénario avec différentes techniques et outils logiciels pour la réalisation de chaque processus. Et le plus

---

<sup>3</sup> Révision structurée

récent c'est l'approche CREWS-L'Ecritoire qui étudie le couplage entre buts et scénarios en proposant d'effectuer des allers et retours entre buts et scénarios. Mais l'aspect coopération comme principe dans CREWS n'a pas suffisamment pris en charge, la distribution des intervenants et leurs hétérogénéités éventuelles. De même et comme conséquence le maintien d'accord avec les différentes parties prenantes, en cas de négociation et / ou collaboration. Au même temps, [71] prédéfinit des scénarios utiles avec "des scripts de test" pour la spécification des exigences, mais il y a une défaillance, car il n'y a pas de processus d'orientation systématique de la génération des scénarios et de cas.

Même si CREWS présente une méthode d'élicitation des exigences coopérative, mais loin d'être une méthode de validation coopérative car les scénarios sont une représentation d'un seul point de vue. Selon Sommerville et al. [61], une seule perspective (point de vue) des exigences du système ne peut pas découvrir l'ensemble des exigences unifiées.

---

## **Conclusion**

---

*La littérature a tendance à considérer la validation des exigences comme un processus hétérogène basé sur l'application d'une grande variété de techniques indépendantes sans pouvoir spécifier le : Quoi, Pourquoi, Qui, Quand et Comment valider et vérifier des besoins/exigences.*

*Bien que la littérature mentionne plusieurs approches pour la validation des exigences, elles peuvent toutes être classées selon le niveau de formalité de la spécification avec la quelle démarre le processus de validation en spécification (formelle, Semi-Formelle, Informelle), Aboutissant a une solution (Automatisé, Semi- automatique ou Manuelle) généralisées en deux niveaux d'évaluation : **un niveau interne**, cette activité est assurée par l'équipe d'IE, **niveau externe** faisons appel aux clients, ou Mixe qui fait intervenir le maitre d'ouvrage guidé par le maitre d'œuvre. Appliqué sur des exemples de systèmes qui peuvent être Critique (Embarqué, temps réel, etc.), Ordinaires de gestion.*

*Et comme notre sujet est autour de l'ingénierie des exigences dans un contexte coopératif. Nous avons jugé utile de présenter dans le chapitre suivant (Chapitre 03) les systèmes d'information Coopératifs, en consacrant aussi une partie sur le travail coopératif assisté par ordinateur et ses outils (CSWS).*

---

---

*CHAPITRE 3 :*

*Les Systèmes D'information Coopératifs  
SIC*

---

# Chapitre III :

## Les

# Systèmes D'Information Coopératifs (SIC)

---

### *Résumé*

---

*Les entreprises s'appuient aujourd'hui de plus en plus sur des organisations de travail qui permettent de rassembler un grand nombre de compétences métiers au sein de projets ou de processus, et de faire coopérer au mieux ces différents individus.*

*Ces nouvelles formes d'organisation du travail se sont largement répandues en exploitant le développement des nouvelles techniques de l'information et de la communication qui permettent de rassembler et faire coopérer un plus grand nombre de participants issus d'horizons et de sites différents.*

*Coopération, collaboration, coordination, communication sont des concepts proches qui ne se recoupent pas toujours. Définir une plate-forme qui favorise la coopération est différent de procéder au développement d'un outil de coordination.*

*Le travail coopératif qui apparaît lorsque plusieurs acteurs sont concernés pour réaliser un travail et sont mutuellement dépendants dans leur travail et doivent coordonner et assembler leurs activités individuelles pour effectivement réaliser le travail qui leur est demandé.*

*Les systèmes d'information coopératifs (SIC), lesquels permettant l'utilisation coordonnée des sources d'information hétérogènes et la résolution des conflits qui en découlent.*

---

## I. INTRODUCTION

Dans tous les types d'industrie, nous observons une pression croissante qui vise à réduire le temps et plus généralement les coûts de développement. Non seulement les entreprises élaborent des produits et des services de plus en plus sophistiqués mais la chaîne logistique se complexifie également. Les évolutions permanentes des marchés, la concurrence intense et le développement rapide des technologies ont considérablement modifié la nature et surtout l'organisation des activités de l'entreprise Nichols G [93]. A l'origine, les organisations de travail étaient construites pour le développement de leurs projets, selon une approche purement séquentielle. C'est la fin d'une activité qui donnait le top de départ de la suivante.

Cette méthodologie analytique de décomposition pouvait être assimilée à la métaphore de la course de relais. En effet la transmission d'informations (bâton de relais) entre deux activités (coureurs) n'était envisageable qu'en fin d'activité (distance parcourue par un relayeur). Si cette approche démontre aujourd'hui ses limites, elle avait le mérite d'être simple à mettre en place et à gérer. C'est pourquoi c'est celle qui était encore récemment appliquée dans de nombreuses entreprises. Le contexte actuel a pour conséquence de forcer les managers à se focaliser sur le processus de développement même et à organiser de manière plus efficace et moins cloisonnée, les personnes et ressources impliquées dans un travail coopératif. Le développement des technologies de communication, notamment par le biais de l'informatique, y a fortement contribué. Ainsi aujourd'hui, les échanges d'information peuvent prendre des dimensions géographiques plus importantes. Et parallèlement, les informations transférées sont de plus en plus complexes (voix, vidéo, etc.) et contraintes (temps réel, etc.).

Ainsi, les entreprises s'appuient aujourd'hui de plus en plus sur des organisations de travail qui permettent de rassembler un grand nombre de compétences métiers au sein de projets ou de processus, et de faire coopérer au mieux ces différents individus. C'est entre autre un des principes de l'ingénierie concurrente (Concurrent Engineering) que de rassembler et pouvoir coordonner les acteurs dans le domaine de l'ingénierie. Cette approche, apparue il y a une quinzaine d'années, prône en effet un rapprochement des compétences métiers. Ce qui favorise à la fois le développement de solutions innovantes et permet à l'ensemble des acteurs d'opérer une synchronisation cognitive, c'est-à-dire un développement mutuel de leurs compétences et connaissances.

Ces nouvelles formes d'organisation du travail se sont largement répandues en exploitant le développement des nouvelles techniques de l'information et de la communication qui permettent de rassembler et faire coopérer un plus grand nombre de participants issus d'horizons et de sites différents. C'est ce qui caractérise le travail coopératif. Le travail coopératif introduit de nouvelles contraintes qui nécessitent des méthodes et des outils adaptés en particulier pour gérer les coopérations souvent complexes qu'elles induisent. En effet, si l'évolution vers des organisations de travail coopératif a pour but d'améliorer la compétitivité des entreprises, elle ne contribue cependant pas à en simplifier la gestion, elle conduit à une division et à une organisation du travail qui augmente

considérablement le couplage entre les différentes activités. Ceci rend l'ensemble du travail plus complexe et donc plus difficilement gérable. Le défi est de parvenir à intégrer un grand nombre d'activités en tenant compte d'innombrables interconnexions entre ces activités. Dans de telles conditions, définir et coordonner les équipes de développement et les activités d'un projet se révèle être une tâche ardue. C'est pourquoi il est nécessaire de disposer de méthodes et d'outils d'aide à la gestion des activités complexes du travail coopératif.

Après avoir présenté la problématique et la motivation derrière les nouvelles formes d'organisation du travail dans la première 1<sup>er</sup> section, Il fait le point sur la définition des différents vocabulaires de ce domaine, Dans la seconde, en suit il s'agira de traiter la mission de travail coopératif, ses éléments, les techniques et les outils logiciels qui le supportent à la section 3, enfin on terminera par décrire les SIC en terme de finalité et Propriétés, typologie et topologie.

## II. PRINCIPALE DÉFINITION

### *Coopération, collaboration, coordination, communication*

Coopération, collaboration, coordination, communication sont des concepts proches qui ne se recoupent pas toujours. Définir une plate-forme qui favorise la coopération est différent de procéder au développement d'un outil de coordination. Nous rappelons ci-dessous les principales définitions du domaine et soulignons les différences majeures.

#### 1. Communication

*Définition : « Echange simple d'informations, concernant par exemple ce qui est à réaliser ou l'état d'avancement du travail » [Petit Robert]*

La différence majeure avec les autres termes réside dans le fait que cet acte peut être réalisé d'une manière complètement indépendante *d'un objectif en commun entre les entités communicantes*. Dans ce cas, communiquer peut être considéré comme une fin en soi. Au contraire, dans le cadre du travail de groupe, la communication est principalement vue comme un moyen.

#### 2. Coordination

*Définition : « Règles et procédures qui assurent le fonctionnement d'un groupe (gestion de flux de tâches) » [Petit Robert]*

On peut également définir la coordination comme : « *l'agencement des parties d'un tout selon un plan logique, pour une fin déterminée* ». Cette définition s'adapte bien à la coordination au sein des organisations. *La coordination est une façon de travailler ensemble de manière harmonieuse*. La mise en place d'une véritable coordination des activités nécessite de subdiviser les objectifs de travail, d'affecter les actions aux groupes ou aux acteurs, de leur allouer des ressources adéquates et partager l'information nécessaire entre eux pour les aider à réaliser leurs buts. La coordination suppose donc :

- ✓ une *décomposition* du processus en activités,
- ✓ une *planification* des activités,

- ✓ une *affectation* des acteurs aux activités.

La coordination est l'ensemble des activités requises pour maintenir la cohérence à l'intérieur d'un processus ou à gérer les interactions entre les agents qui accomplissent ce processus. Elle est rendue plus complexe d'une part parce que les étapes du processus ne sont plus organisées en fonction des étapes d'une méthode linéaire mais en fonction de modules cohérents représentant un résultat en termes de produit (informationnel ou physique); d'autre part parce que ces étapes doivent quand même être traitées en vue d'un objectif commun. Les recouvrements entre activités bien que marginaux dans l'idéal favorisent l'émergence de conflits et traduisent souvent un manque de cohérence. Il faut donc un partage transversal d'informations entre les acteurs et une gestion de la prise de décision partagée. Ceci est surtout important en conception car elle encourage un gel tardif des décisions et une vision élargie au cycle de vie du produit mêlant forcément des acteurs et des points de vue différents. La conception du produit est le moment où ce besoin de coordination et d'assistance à la prise de décision est le plus crucial.

La coordination entre acteurs a pour objectif la mise en œuvre du travail. Elle implique en général de la coopération :

- ✓ lors de l'élaboration du planning,
- ✓ lors de sa mise en œuvre dynamique (pour faire face aux incertitudes et aléas).

### **3. Coopération**

*Définition : « Action de participer à une œuvre commune. Système par lequel des personnes intéressées à un but commun s'associent et partagent le profit. » [Petit Robert]*

Marx [94] avait défini en 1867 la coopération comme l'activité où : « *plusieurs individus travaillent ensemble de façon planifiée dans le même processus de travail, ou dans des processus distincts, mais liés entre eux* » Marx [94]. On peut ajouter à cette définition l'aspect « productif ». Ainsi, un environnement coopératif est un environnement de production dans lequel c'est le groupe qui produit.

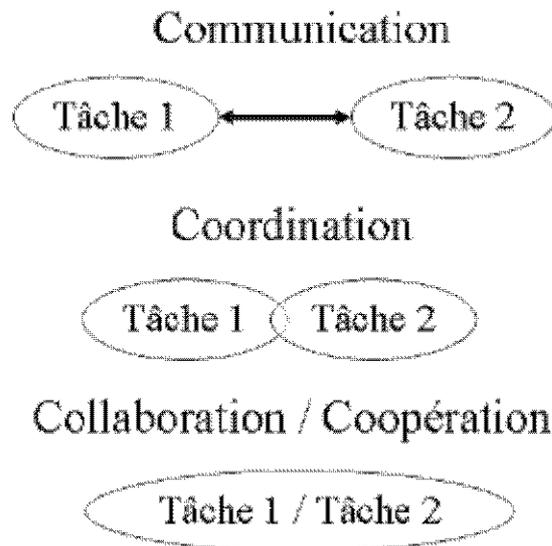
Du point de vue de la gestion des connaissances et compétences, la coopération met en lumière deux types de situations :

- ✓ la résolution de problèmes de manière distribuée, situation dans laquelle le problème est tellement complexe qu'une seule et même personne ne peut en construire qu'une représentation partielle et limitée;
- ✓ la résolution de problème en groupe, situation dans laquelle chacun peut appréhender la structure globale du problème.

### **4. Collaboration**

*Définition : « Création d'une vision commune des questions à traiter et d'un espace commun pour stocker et partager les informations. » [Petit Robert]*

Egalement synonyme de travail en commun, d'aide, de concours, de participation, ..., le terme de collaboration est assez proche de celui de coopération. Nous emploierons ainsi indifféremment les deux termes. Cependant, la collaboration est quelquefois prise dans une acceptation plus large de la définition des interactions entre individus ou entre groupes de travail, lorsque le processus de conversation est peu structuré.



**Fig 25.** Communication, Coordination, Collaboration/Coopération

La figure 27 permet d'appréhender au travers d'une relation entre deux tâches, les différents termes. Ce qui les distingue est la part d'éléments mis en commun : la communication traduit un simple partage de moyens, la coordination implique une intersection entre les domaines définis par les deux tâches et la collaboration/coopération en fait la mise en commun totale. On se rend ainsi aisément compte que la coopération/ collaboration, la coordination et la communication ne sont pas strictement synonymes et les processus impliqués ne peuvent être simplement résolus par la mise en œuvre d'outils de communication, aussi sophistiqués soient-ils. Dans tous les cas, les outils et méthodes de traitement de l'information sont au cœur du débat puisqu'il ne peut y avoir coopération ou coordination que s'il y a échange et partage d'informations et donc communication.

Il ne suffit pas de gérer indépendamment ces trois aspects du travail en groupe que sont la communication, la collaboration et la coordination (les 3 C). Il faut, pour en tirer toute la puissance, les mettre en synergie. Une fois encore, ce qui s'en dégagera sera plus performant que la simple addition des trois éléments.

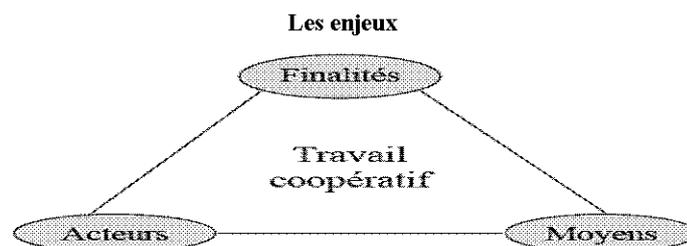
### III. LE TRAVAIL COOPÉRATIF

#### 1. Définition

Le travail est un phénomène social. Des milliers de connexions et de relations sont réunies ensemble dans les processus de travail. Chaque nœud de ce réseau de processus interdépendants constitue en lui-même un petit monde social propre, impliquant un effort concerté de nombreuses personnes. On peut ainsi reprendre la définition proposée par Schmidt [95] :

Si l'on considère la notion profondément sociale du travail, on ne peut que louer les efforts qui ont abouti à la réalisation de systèmes informatiques destinés à faciliter la coopération des travailleurs. En effet, de tels systèmes ont depuis longtemps été implémentés dans les entreprises, les usines, les organisations sociales. Mais leur utilisation pratique dépend de leur capacité à s'adapter à l'organisation du travail dans l'entreprise. Le travail coopératif est donc une thématique de recherche multidisciplinaire, associant des chercheurs des sciences humaines (psychologues, sociologues, économistes) ainsi que des chercheurs en informatique ou télématique, associés aux futurs utilisateurs, acteurs de terrain.

« Le travail coopératif apparaît lorsque plusieurs acteurs sont concernés pour réaliser un travail et sont mutuellement dépendants dans leur travail et doivent coordonner et assembler leurs activités individuelles pour effectivement réaliser le travail qui leur est demandé. » Schmidt [95]



**Fig 26.** Elément clés du travail coopératif

Nous pouvons appréhender le travail coopératif au regard de trois entités figure 28. Ainsi, la coopération équivaut au « partage » au sens d'une mise en commun entre *acteurs* de *moyens* : ressources, informations, connaissances, ... ; et de finalités : avec au minimum la connaissance des objectifs des autres acteurs de la coopération, et au maximum, une responsabilité conjointe solidaire sur un ensemble d'objectifs.

Un partage de ressources sans partage d'objectifs conduit à l'affrontement de logiques d'acteurs, source de conflits potentiels. Dans la plupart des cas, la résolution des conflits implique alors le recours à un tiers : hiérarchie, coordinateur, arbitre, tribunal, ... Mais cette solution est peu praticable et souvent préjudiciable pour une organisation en termes de coût.

Un modèle paraissant idéal serait de responsabiliser un ensemble d'acteurs sur un ensemble d'objectifs (orientations et décisions pertinentes et assumées) mais le risque serait alors de diluer les responsabilités et de perdre beaucoup de temps à fonctionner par décision collective. La question est de trouver un compromis entre des identités fortes d'acteurs et un recouvrement des objectifs de façon à atteindre les principaux enjeux du travail coopératif qui sont de :

- ✓ *favoriser la coordination des participants,*
- ✓ *faciliter la communication entre les participants,*
- ✓ *faciliter le partage des ressources,*
- ✓ *favoriser la cohésion du groupe,*
- ✓ *favoriser l'implication individuelle,*
- ✓ *faciliter l'organisation du groupe.*

## **2. Les méthodologies de management**

Après avoir investi pour optimiser leur système d'information (ERP, CRM, SCM ...), les entreprises s'attachent aujourd'hui à organiser leurs activités autour de leurs différents processus métiers. Elles ont pris conscience de l'enjeu associé à la maîtrise des processus, comme en témoigne l'importance croissante accordée aux certifications, aux réglementations et aux démarches qualité. Il est en effet aujourd'hui établi que les gains de productivité générés par la mise en œuvre généralisée des procédures de management seront du même ordre de grandeur que ceux obtenus lors de l'automatisation de la chaîne de fabrication.

### **2.1. Gestion des processus**

La gestion de processus métiers, appelée aussi *Business Process Management* (BPM), est l'analyse et la modélisation des procédures mises en place par l'entreprise pour réaliser ses activités. Ceci nécessite l'intervention d'une nouvelle catégorie de logiciels unifiant le système d'information et permettant aux opérationnels de reprendre la main sur la définition, la mise en place, le suivi et l'évolution des procédures de l'entreprise. Plus spécifique à la coopération, le *Business Process Reengineering* (BPR) Khoshafian et al [96] consiste à remodeler, rationaliser des équipes de travail mais aussi à identifier les responsabilités des acteurs au sein du processus de travail. Les caractéristiques des logiciels de gestion de processus sont :

- *la désintermédiation* : permettre aux experts et décideurs de définir directement les nouvelles règles, procédures ou applications sans l'intervention d'un informaticien,
- *l'interopérabilité* : ne pas remettre en cause le système d'information existant, s'appuyer sur ses ressources pour faciliter le déploiement des nouvelles procédures,
- *l'évolutivité* : offrir les outils nécessaires pour intégrer les technologies et les composants du système d'information et pour répondre aux évolutions rapides de l'environnement de l'entreprise,
- *la réutilisabilité* : fournir la capacité de partager les "bonnes pratiques" au sein de l'entreprise pour augmenter sa réactivité et son efficacité.

### **2.2. Gestion de la performance**

Le *Corporate Performance Management* ou CPM est une approche de gestion qui propose une amélioration globale de l'organisation. La « gestion globale » de la performance correspond à la promesse, pour un cadre exécutif, d'une part d'avoir accès aux statistiques détaillées de toutes les

activités de son entreprise (pour leur appliquer des traitements, identifier ainsi les améliorations possibles, imaginer des pistes de réorientation stratégique...), et d'autre part d'observer directement les répercussions d'éventuels changements organisationnels [32] ou les répercussions dans la répartition des budgets évaluées par l'outil informatique. Plus encore, une solution de CPM permet dans l'idéal au cadre en question de valider la plus efficace des stratégies envisagées, et de la mettre en application. Il quitte alors le mode "simulation" pour rentrer dans le mode "opérateur". Les procédures sont modifiées, les informations des ERP mises à jour, les objectifs des salariés révisés, etc. Toute la chaîne de commandement et d'organisation est automatiquement informée de ces changements. Bref, le CPM se présente comme une forme complète et ambitieuse d'assistance au management. Si l'on en juge par l'étendue des pré requis, il semble que le CPM soit encore très loin de nous, et réservé aux entreprises les plus volontaristes et les plus dématérialisées.

### 2.3. Gestion des connaissances

Totalement intégrée dans l'amélioration des processus, la gestion des connaissances est la pierre angulaire de la stratégie de management. Dans notre approche, chaque acteur d'un processus coopératif acquiert des connaissances « projet ». C'est cette appréhension collective des informations relatives à un processus de travail qui permet l'évolution du processus.

Baumard [97] a présenté les différentes possibilités de transformation de la connaissance selon deux axes voir figure 29, le nombre d'acteurs (individu ou collectif) et le caractère implicite ou non de la transformation. Sur cette représentation, ce sont essentiellement les transformations horizontales relatives au nombre d'acteurs qui vont nous intéresser.

Pourtant, le Gartner prévoyait en 2003 que 40 % des entreprises adopteront le CPM d'ici 2005. Une chose est sûre : il manque encore un standard universel, qui permette à tous les progiciels de chaque entreprise d'interagir en vue de former un véritable système de CPM.

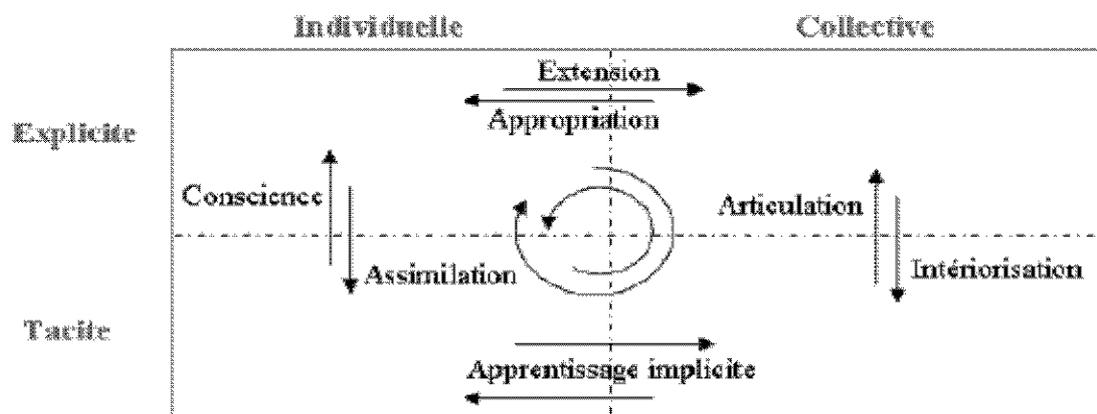


Fig 27. Transformation de la connaissance [97]

Le cadre de développement du travail coopératif doit ainsi permettre :

- d'étendre les connaissances individuelles au groupe,

- de faciliter *l'appropriation* des connaissances collectives par les individus.

Une fois ces pratiques assimilées, le partage et *l'apprentissage* des connaissances pourront être réalisés de manière *implicite*. Comme le concept de *Knowledge Based Engineering* ou KBE le prône, il s'agit donc de favoriser la remontée d'informations en provenance de chaque activité (par des phénomènes de *conscience* et d'*assimilation*). Ceci afin d'alimenter la vision collective, qui est une forme d'intelligence comme le décrit Pierre Levy [98]. Selon le niveau de connaissance, on recueille tout d'abord des informations sur le processus de travail lui-même, puis sur des références ou indicateurs de performance du processus pour enfin parvenir à estimer et à améliorer continuellement sa qualité. Au plus près de la philosophie poursuivie dans cette thèse, DornJ[99] propose une approche itérative de management. Elle consiste à appréhender le développement selon des phases successives de travail (élaboration de la connaissance) et d'échanges (partage de la connaissance).

### 3. Les techniques et outils logiciels support du travail coopératif

Pour réaliser un développement mutuel de connaissances, les managers peuvent entre autre s'appuyer sur des techniques et outils appropriés. On doit le concept de Travail Coopératif Assisté par Ordinateur (TCAO) à Ray Ozzie, créateur de *Lotus Notes* et considéré comme le père du groupware. Aujourd'hui, toutes les entreprises - ou presque - utilisent ce type d'outils. Certaines se contentent d'améliorer leur communication grâce à l'email, d'autres vont beaucoup plus loin en confiant la supervision et la coordination de leurs projets à un ordinateur.

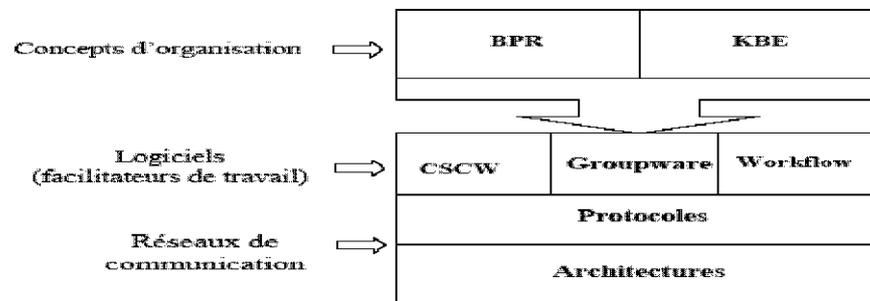
La littérature fournit deux grandes définitions de la TCAO :

- La première que l'on peut qualifier de sociale nous est donnée par Bonnon [100] : « *La TCAO devrait être considérée comme une tentative de comprendre la nature et les caractéristiques du travail coopératif, avec comme objectif la conception d'une technologie informatique adéquate.* »
- La seconde est axée uniquement sur l'aspect informatique Ellice [101] : « La TCAO est un ensemble de systèmes informatiques qui assistent un groupe de personnes engagées dans une tâche commune et qui fournissent une interface à un environnement partagé. »

Ces deux définitions se retrouvent sur un point : l'importance de concevoir un environnement partagé propice au travail de groupe. L'implémentation des organisations de travail se fait par l'intermédiaire d'outils permettant de « reconnecter » entre elles les ressources de l'entreprise distribuée voir figure 30. Les méthodes d'organisation du travail coopératif sont facilitées par l'informatique. Les technologies développées permettent l'établissement de protocoles de communication qui sont mis en œuvre à travers les réseaux pour définir des architectures de travail. Dégageons les trois technologies les plus en vue :

- Le CSCW (Computer Supported Cooperative Work) : activité coordonnée assistée par ordinateur, comme la communication ou la résolution des problèmes effectués par un groupe d'individus qui coopèrent Backer[102].

- Le Groupware : technologie développée dans le but de faciliter les interactions informelles en renforçant les aspects communication, coordination et coopération Bock [103].
- Le Workflow : technologie développée en vue de restructurer les procédures métiers de l'entreprise en automatisant les règles formelles bock [103].



**Fig 28.** Intégration des coopérations

Toutefois, de nombreuses définitions et utilisations de ces termes existent, c'est pourquoi nous allons en expliciter les nuances. Ceci permettra de faire une synthèse des outils de Travail Collaboratif Assisté par Ordinateur (TCAO).

### 3.1. Le CSCW

Le CSCW (traduction anglaise du terme TCAO qui cependant est moins réducteur) étudie les mécanismes individuels et collectifs du travail de groupe et recherche comment les technologies de l'information et de la communication peuvent faciliter ce travail. C'est un domaine de recherche multidisciplinaire intéressant aussi bien pour les sociologues, les psychologues, les ergonomes que les informaticiens et évidemment les managers. Bien que le CSCW comporte le terme "Computer", les outils mis en jeu dans cette discipline dépassent de beaucoup l'ordinateur en incluant la téléphonie, les messageries, la vidéo et les systèmes d'imagerie Gurdin [104].

D'après Wilson [105], « *CSCW est un terme générique qui combine la compréhension du travail des individus dans les groupes avec les technologies habilitantes (disponibles) des réseaux de communication, la communication et la coordination à l'intérieur du groupe partageant le même espace d'information ainsi que les interfaces multiutilisateurs qui sont les composantes clés du CSCW.*

### 3.2. Le groupware

Toutes les applications que l'on met en œuvre dans les intranets collaboratifs font partie d'un domaine de l'informatique qui existe maintenant depuis plus d'une quinzaine d'années sous le vocable de groupware (parfois traduit par collecticiel en français). Les outils appartenant à cette grande famille de solutions permettent d'améliorer grandement le travail coopératif entre les différents membres d'une même équipe ou entre les membres d'équipes différentes ayant des informations à partager. Peu d'entreprises osait franchir le pas jusqu'au milieu des années 90, mais avec l'avènement des technologies Internet, il se trouve que les protocoles de base nécessaires à ce type d'application se sont

standardisés, donnant lieu à une expression de besoins de la part des utilisateurs jamais vu jusqu'à présent dans ces domaines.

On retrouve donc maintenant tout naturellement ces applications recensées comme point central des intranets devant être déployés au sein des entreprises.

Le domaine de la publication d'informations a une couverture plus large que les simples applications de groupware. C'est pourtant un domaine que l'on retrouve pris en compte dans les grands produits de groupware tels que Lotus Notes, par la possibilité de créer des bases de données dédiées à la publication d'information, les bases de documents. C'est bien entendu ce qui a fait le succès des sites Web, grâce à la création du langage HTML et des serveurs HTTP, qui ont permis la diffusion d'information sur n'importe quel réseau TCP/IP de base. *des matériels et logiciels associés, des services et des techniques* ». La prise de conscience de groupe, etc.

D'après Bock [103], le groupware n'est pas seulement un produit logiciel, mais un mode de fonctionnement en équipe qui facilite et améliore la coopération, la communication et la coordination (les 3C). Il implique simultanément trois constituants indissociables :

- ✓ le management, c'est à dire la gestion des hommes, des métiers et des compétences,
- ✓ l'organisation qui détermine le processus de travail,
- ✓ les technologies de l'information, qui apportent les moyens pour faire vivre les processus.

Il existe différentes typologies des outils de groupware. Les outils peuvent être ainsi classés en fonction de leurs apports en matière de communication, coordination et coopération voir figure 31. On peut également proposer une taxinomie selon quatre grandes classes d'outils :

- *Les outils de communication de base (mail, tableau blanc, vidéoconférence ...)*
- Les espaces de travail partagé (partage d'applications, forums ...)
- Les outils d'accès au savoir ou de Knowledge Management (bibliothèques, portails ...)
- Les outils de workflow (outils de gestion des tâches, de synchronisation, agendas partagés ...)

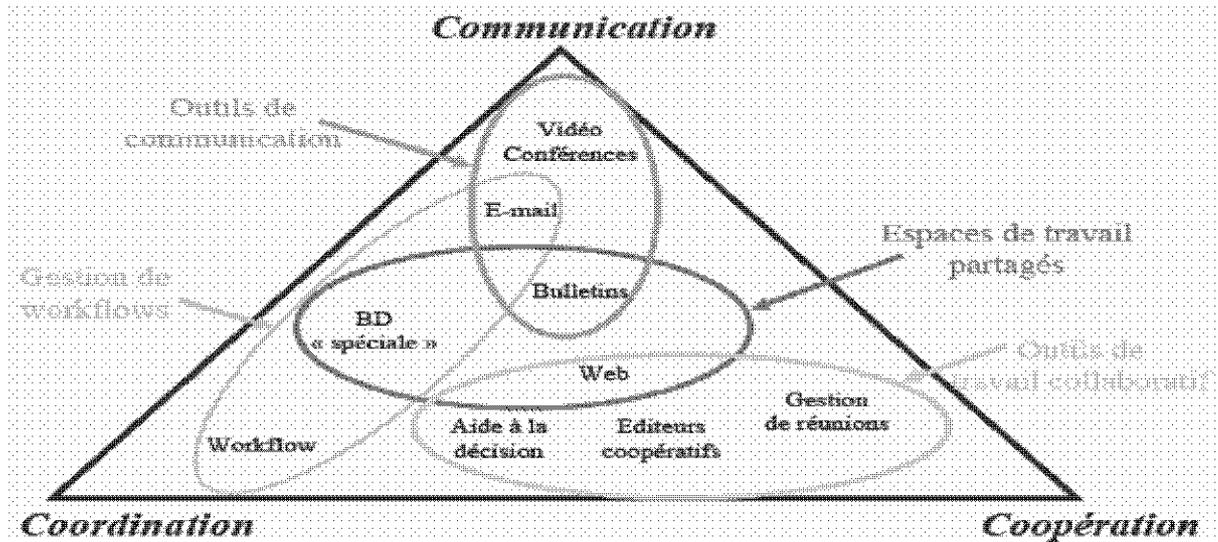


Fig 29. Taxonomie des outils de TACO

### 3.3. Le Workflow

Le workflow, littéralement « flux de travail », peut être défini comme l’outil support d’un travail coopératif impliquant un nombre limité de personnes devant accomplir, en un temps limité, des tâches articulées autour d’une procédure définie et ayant un objectif global. Le workflow est souvent synonyme d’ « automatisation des processus d’entreprise ». C’est un outil qui tente d’automatiser la séquence des actions, des activités ou des tâches que nécessitent un processus de travail depuis sa création jusqu’à sa phase terminale. Il permet de gérer l’ensemble du processus en suivant l’état de chacune de ses instances Marshak [107]. En identifiant et en reliant les participants aux processus, il rend l’information disponible de façon méthodique, tout en améliorant la qualité et la vitesse du processus. Il permet ainsi de fédérer les ressources et les acteurs de l’entreprise, aujourd’hui de plus en plus dispersés.

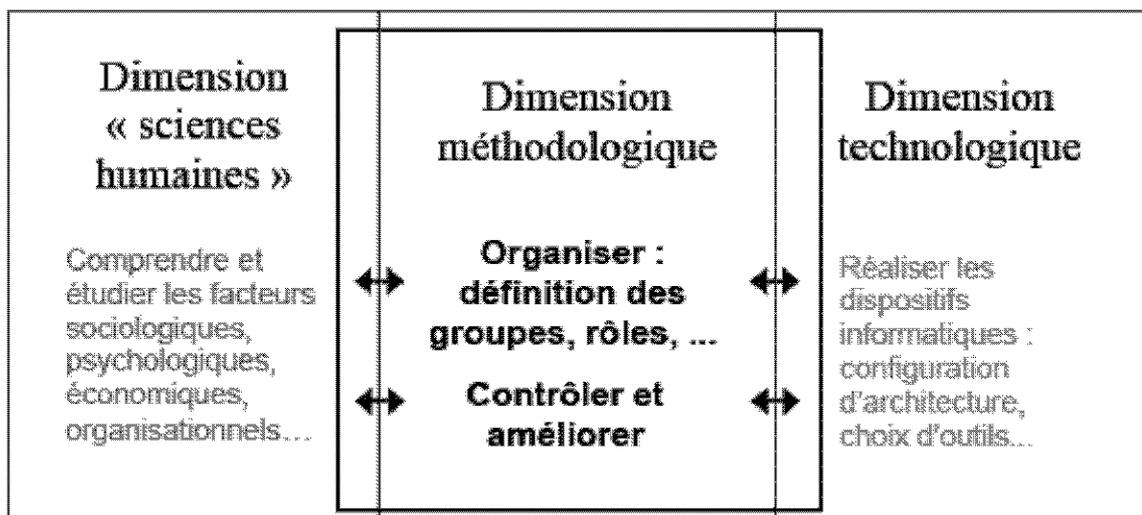
Un logiciel de workflow est un outil qui responsabilise les individus et les groupes, dans des environnements structurés ou non, en vue de gérer automatiquement une série d’événements récurrents ou non pour atteindre les objectifs métiers de l’entreprise Palem et all [108]. On distingue alors deux types de workflow :

- Workflow de *production* (ou transactionnel) : aussi appelé workflow directif, il correspond à des processus métiers connus de l’entreprise qui font l’objet de procédures pré-établies. Le cheminement du workflow de production est donc plus ou moins figé.
- Workflow *ad hoc* : il est basé sur un modèle collaboratif dans lequel les acteurs interviennent dans la décision du cheminement. Le cheminement du workflow ad hoc est donc dynamique.

Le workflow concerne un travail dans lequel plusieurs personnes sont impliquées pour atteindre l’objectif global, mais à des étapes différentes du développement. Les procédures sont alors vues comme un enchaînement pré-établi de traitements découplés. Le problème de l’organisation du travail

à l'intérieur d'une étape du développement reste en suspend. De même, le workflow ne prend que très peu en compte, voire pas du tout, l'organisation du travail des individus, et n'aborde pas les aspects dynamiques.

Les outils de travail coopératif constituent une aide à la mise en place et aux développements des organisations de demain, basées sur des structures horizontales. Leur impact sur l'organisation du travail est important. Cependant, la plupart des méthodes actuelles de conception de tels systèmes coopératifs est trop orientée vers l'organisation des données et l'automatisation des traitements (les outils) et non vers l'organisation du travail des hommes (les méthodes). Il est donc nécessaire de mettre en évidence les spécificités organisationnelles du travail coopératif pour construire un système d'information adéquat. De même, il faut tenir compte le plus tôt possible des outils disponibles de communication et de coopération pour l'élaboration et l'amélioration des organisations de travail coopératif.



**Fig 30.** Les trois dimensions de travail coopératif

Comme nous l'avons montré précédemment, le travail coopératif est un domaine pluridisciplinaire. Trois dimensions le caractérisent figure 32 :

- *sociologique*, où il s'agit d'étudier les aspects psychologiques, socio-économiques du travail en groupe. Ceci permet d'analyser et de comprendre les comportements individuels et ainsi d'anticiper les problèmes éventuels pouvant rejaillir sur une organisation coopérative ;
- *méthodologique*, où l'objectif principal est d'organiser au mieux le travail coopératif. Cela passe par l'intégration des besoins pointés par les Sciences Humaines et Sociales (SHS) pour réussir un développement multi participatif. Il s'agit donc d'analyser les besoins, de structurer l'organisation en groupes de travail, de définir les rôles et les responsabilités mais également d'évaluer les performances de l'organisation en vue de son contrôle et de son amélioration ;

- *technologique*, où l'on met en œuvre les préconisations établies dans l'organisation méthodologique du travail coopératif. Il s'agit entre autre avant le développement, de choisir les outils adéquats, de les configurer et de définir l'architecture du réseau de communication correspondante. Le réseau peut également être instrumenté en vue de retourner des renseignements sur un processus au cours de son développement. Ces retours sont alors utilisés pour le suivi du travail et l'adaptation éventuelle de l'organisation.

Le domaine particulier dont ce mémoire fait l'objet est l'aspect méthodologique. Cette dimension fait le lien ou le tampon entre les deux autres domaines, ses objectifs sont :

- ✓ d'analyser et d'intégrer entre autres les facteurs humains au travers de l'étude des interactions pour organiser « au mieux » le travail coopératif et permettre de configurer les architectures et les divers éléments technologiques, supports du travail coopératif,
- ✓ de prendre en compte les retours délivrés par les dispositifs technologiques pour contrôler et améliorer continuellement l'organisation des processus coopératifs.

## IV. SYSTÈMES D'INFORMATION COOPÉRATIFS (SICs)

### 1. Définitions

Un système d'information coopératif SIC peut être défini comme un « *ensemble de composants plus ou moins autonomes, souvent préexistants qui travaillent de manière synergique en échangeant information, expertise et en coordonnant leurs activités* »Boulangier [109]. Dans le cadre de la coopération de sources d'information hétérogènes, un SIC vise, plus précisément, à permettre l'utilisation conjointe de ces sources et résoudre les différents conflits inhérents.

Zarour [118] considère que les SICs comme « *un ensemble de SIs distribués à travers un grand nombre de réseaux d'ordinateurs et de communication complexes. Ces systèmes supportent toutes les catégories d'applications, des plus simples aux plus complexes. De tels Sis gèrent et accèdent à une grande quantité d'information et de services automatisés, ils supportent également le travail humain individuel ou de groupe. L'exécution des tâches s'effectue de manière concurrente sur des réseaux à l'aide de logiciels spécifiques. Les informations et les services sont disponibles sous plusieurs formes grâce à des référentiels. La communication entre les systèmes compétents peut s'effectuer de façon centralisée ou distribuée grâce à des protocoles de communication* ».

### 2. Typologie des systèmes d'information coopératifs :

La plupart des SIC existants peuvent être classés selon deux familles ou approches : l'approche fédérée et les approches à base de médiation.

#### 2.1. L'approche fédérée

L'approche fédérée (ex. de SIC fédérés : MRDSM [110], Mínd [111],...) repose sur l'intégration. Chaque SI exporte un schéma dans un modèle pivot, les différents schémas d'export étant alors intégrés dans un (ou plusieurs schéma(s) fédéré(s). Un schéma fédéré permet ainsi un accès uniforme

aux données partagées. Cette approche ne fonctionne cependant que si le nombre de SI impliqués est limité.

## 2.2. Les approches à base de médiation

Les approches à base de médiation reposent sur deux composants qui sont le médiateur et le wrapper. Le médiateur simplifie, abstrait, réduit, combine et décrit les données et est chargé des traitements permettant à l'utilisateur d'obtenir des informations extraites des SI locaux. Il permet principalement de résoudre les conflits sémantiques (i.e. conflits de domaine de définition, conflits dans l'interprétation d'une même donnée). Le wrapper fournit une interface d'accès homogène aux sources d'information et résout les conflits de modèle de données et de langage d'interrogation en présentant les données dans le modèle de médiation. On distingue deux types de médiation : la médiation de schémas (ACRIS [112], Tsimis [113],...) qui construit au préalable une base d'informations prenant en compte les SI participants pour permettre au médiateur de faire son travail d'intégrateur et la médiation de contextes (InfoSleuth [114], DILEMMA [115],...) qui repose sur l'intégration dynamique des informations en fonction du contexte de l'application ou de l'utilisateur.

Les systèmes d'information coopératifs nécessitent d'être spécifiés et développés ; la notion d'ingénierie des SIC est présentée en section suivante.

## 3. Propriétés des SIC

Les SIC disposent de différentes propriétés, qui sont évoquées dans Zarour [118], ce paragraphe présente trois propriétés essentielles des SICs, qui sont l'Autonomie, la Distribution et l'Hétérogénéité.

### 3.1. L'Autonomie

Autonomie est relative au contrôle indépendant et séparé des systèmes composants. Le fonctionnement global des SICs ne peut être influé par l'abandon ou l'intégration d'un composant. Scheurman et al [122] identifie trois niveaux d'autonomie :

- *L'autonomie de conception* où chaque système composant des SICs est libre de choisir sa propre conception. Néanmoins il doit respecter certaines contraintes comme le langage de requêtes. Elle désigne le choix du domaine de gestion et la conceptualisation du contexte. En relation avec d'autres systèmes, elle représente la capacité d'un système ou d'un composant de choisir "ses partenaires" pour l'échange des ressources, ainsi que les modes de réalisation de ces échanges.

- *L'autonomie de Communication* se réfère à la capacité des systèmes composants de décider de la façon de communiquer entre eux. Elle représente la capacité, pour un nœud du réseau (système ou composant) de communiquer avec d'autres nœuds.

- *L'autonomie d'exécution* permet à un système d'exécuter ses opérations locales sans interférence avec les opérations soumises par d'autres systèmes composants. Elle concerne la capacité pour un nœud d'exécuter des opérations locales indépendamment des composantes externes.

### 3.2. La Distribution

La Distribution est une propriété remarquable des SICs. Elle identifie les interactions entre différentes composantes; d'une manière plus fine, ces interactions peuvent se décomposer selon

plusieurs axes (qui, pourquoi, quoi, quand, comment, où) Salzano et al [119]. La distribution physique des sources de données est présente à côté de l'autonomie et l'hétérogénéité des systèmes. Les techniques telles que COBRA, RMI/Java permettent la prise en charge de la distribution.

### 3.3. L'Hétérogénéité

L'Hétérogénéité se manifeste sous plusieurs formes. Elle peut être provoquée par les différentes technologies au niveau matériel et/ou logiciel (Systèmes d'exploitation, réseau de communication ...). L'hétérogénéité consécutive des différences des composants SI. Ces différences se situent en général au niveau des modèles de données pour *l'expression des structures et des contraintes* et aussi au niveau des *langages des requêtes*. Les différentes formes d'hétérogénéité engendrent plusieurs types de conflits lors de la coopération des SIs. L'hétérogénéité se manifeste à plusieurs niveaux ([120], [121]) et peut concerner *les systèmes* (plateformes techniques, systèmes logiciels), comme *l'information* elle-même, pour des problématiques relevant de la syntaxe, de la structure et de la sémantique.

### 4. Types de connaissances dans les SIC :

Nous pouvons affirmer que les Systèmes d'Information prennent de nos jours une place centrale dans les organisations. En effet, on peut considérer *deux types de populations dans les organisations* pour les années à venir.

Dans le premier cas, *les populations nomades* de l'entreprise doivent bénéficier du même niveau d'information que le collaborateur travaillant sur son poste fixe, pourvu que techniquement, la chose soit permise par l'exploitation des technologies sans fil, et sécurisée par un système de contrôle d'accès.

Dans le second cas, il s'agit non seulement de permettre des échanges, mais aussi de les inclure dans des ensembles plus vastes, cohérents et correspondant à la spécificité opérationnelle de l'entreprise - les fameux processus métier incluant non seulement la population sédentaire de l'entreprise mais aussi les clients, les fournisseurs et tous les acteurs étant partie prenante de ce processus.

Il s'agit donc pour les architectes de Systèmes d'Information de concevoir des systèmes offrant les mêmes fonctionnalités que les systèmes d'information classiques mais devant par ailleurs supporter le travail collaboratif et coopératif au sens large. Le développement des tels systèmes, Systèmes d'Information Coopératifs, fait appel à plusieurs types de connaissances selon Pascal [123] aussi bien informatiques qu'organisationnelles :

- Entreprise Etendue
- Conception centrée sur l'utilisateur (User Centred Design)
- Interfaces Adaptatives Intelligentes
- Collecticiel

#### 4.1. *Entreprise Etendue (Extended Enterprise)*

D'un point de vue technologique, "l'Entreprise étendue" désigne l'adaptation du système d'information à deux types d'échelle:

- au sein de l'entreprise, la multiplicité des canaux d'accès aux applications et aux données
- au delà de l'entreprise, le réseau des clients, fournisseurs, partenaires (et bien sûr, filiales éventuellement hétérogènes - suite, par exemple, à une fusion acquisition) qui interviennent dans les processus organisationnels.

D'un point de vue fonctionnel, l'Entreprise étendue se traduit par une plus grande collaboration et une meilleure gestion des connaissances. Elle permet également de répondre aux problématiques de gestion de cycle produit, en agrégeant de manière satisfaisante les données de conception, de gestion financière et logistique, les données relatives aux fournisseurs et aux clients.

Egalement, l'Entreprise étendue vise à réduire les cycles d'approvisionnement - places de marché privées par exemple - et de décision.

#### ***4.2. Conception centrée sur l'utilisateur (User Centred Design)***

La conception de tels systèmes, Systèmes d'Information Coopératifs, suppose une prise en compte accrue de l'utilisateur. Afin d'assurer une meilleure prise en compte des besoins d'utilisation, les méthodologies et technologies utilisées doivent placer les utilisateurs au cœur du processus de conception soit par des méthodes de conception telles que UML par exemple soit par une validation importante de tels systèmes par des groupes d'acteurs divers.

#### ***4.3. Collecticiel***

Il paraît évident que les systèmes développés devront supporter le travail collaboratif et / ou coopératif. Les membres du groupe peuvent collaborer à distance, soit au même moment (activité synchrone), soit à des moments différents (activité asynchrone). L'objectif principal de tels outils étant de favoriser la production, dans le contexte du travail collaboratif à distance et de l'usage d'un environnement virtuel partagé par un groupe d'utilisateurs, le concept d'awareness prend une place importante. Ce terme désigne en fait la perception que possède chacun de la présence, de la localisation, de l'identité, de la disponibilité de l'autre à un moment donné, lors de la connexion

#### ***4.4. Outils d'Aide à la Décision de Groupe***

Un des aspects stratégiques des systèmes d'information dans l'entreprise est de pouvoir supporter la prise de décision au travers des Systèmes Interactifs d'Aide à la Décision. L'élargissement des organisations à leurs partenaires, contribue, entre autres phénomènes, à une évolution de la prise de décision. Les décideurs participent à un processus global de prise de décision sans en avoir forcément la maîtrise complète. On assiste alors à une plus grande autonomie de ces acteurs particuliers de l'entreprise en même temps que leurs responsabilités se trouvent de plus en plus diffuses et de plus en plus entrecroisées. Les systèmes d'aide à la décision de groupe doivent alors intégrer des contraintes d'utilisation telles qu'il peut ne pas exister ni unité de lieu ni unité de temps entre tous les décideurs inclus dans un processus. On se place alors dans un contexte de prise de décision asynchrone et distribuée.

#### 4.5. *Interfaces Adaptatives Intelligentes*

Par ailleurs par systèmes d'information coopératifs nous entendons systèmes coopératifs au sens large, c'est à dire des systèmes devant assurer une bonne Coopération entre l'homme et la machine. Nous nous intéressons particulièrement à toutes méthodologies ou technologies pouvant améliorer les performances du couple homme machine.

#### 5. **Types (Formes) des systèmes d'information coopératifs (SIC).**

Les SICs sont Issues principalement de deux courants de recherche, l'Intelligence Artificielle Distribuée (IAD) et les bases de données, la coopération de systèmes d'information et ses solutions, les Systèmes d'Information Coopératifs, reposent sur plusieurs acceptions.

Dans l'approche des bases de données distribuées, un SIC « *intègre des sources d'informations distribuées - bases de données ou systèmes à bases de connaissances – pouvant utiliser des représentations de connaissances et de données hétérogènes* » Brodie [125].

La communauté de *l'Intelligence Artificielle Distribuée* définit un SIC comme « un ensemble d'agents "computationnels" qui partagent continuellement des objectifs avec d'autres systèmes d'information, des agents humains aussi bien qu'avec l'organisation au sein de leur environnement opérationnel » Herin [127], boulangier [109] proposent une définition plus fédératrice dans laquelle un SIC est perçu comme « un ensemble de composants plus ou moins autonomes, souvent préexistants qui travaillent de manière synergique en échangeant information, expertise et en coordonnant leurs activités ».

D'autres définitions sont données par des auteurs comme :

Dans *le domaine du travail coopératif assisté par ordinateur* (TCAO ou CSCW), un SIC est "Un ensemble de composants plus au moins autonomes, souvent préexistants travaillant ensemble de façon synergique en échangeant des informations, expertise et en coordonnant leurs activités" boulangier [109].

Huhns [129] considère, que le développement des systèmes d'information coopératifs est passé par quatre générations, systèmes d'information centralisé, architecture Client/Serveur architecture Peer to Peer, et finalement les systèmes coopératifs.

Pour Barret et all [128], *le Système d'Information Interorganisations* (SIIO) est défini en tant que « ..Système qui implique le partage entre deux ou plusieurs organisations, de ressources d'information telles que le hardware, software, moyens de transmission, règles et procédures, données, bases de données et expertise ».

Pour Boughla et all [126] « ...Nous définissons donc le Système d'Information Coopératif Interentreprises (SICI) comme un système qui met en relation des groupes d'acteurs, restreints ou étendus, appartenant à des organisations juridiquement indépendantes et prenant trois formes de SI :

- Un SI de communication, qui permet de faciliter les dialogues (l'interaction communicative) entre entreprises;

- Un SI de coordination, qui permet de faciliter la coordination entre les entreprises en s'appuyant sur des concepts de rôles, règles, routes, ressources, etc. ;
- Un SI orienté Résolution Collective de Problèmes (RCP), qui permet de faciliter la collaboration entre des entreprises travaillant sur des tâches communes afin d'atteindre un objectif commun. ».

## **6. Ingénierie des systèmes d'information coopératifs :**

L'ingénierie des systèmes d'information coopératifs (de type sources d'information) est un « domaine de recherche qui à trait au développement systématique de solutions interopérables pour les systèmes hétérogènes et autonomes comprenant à la fois des bases de données mais aussi d'autres types de sources d'information provenant de domaines d'application variés » Conrad et all [116]. Elle vise ainsi à construire des solutions réutilisables dédiées à la coopération de sources d'information. Pour Fernandez et all [117], cette tâche d'ingénierie est délicate car elle repose sur une double complexité : (1) les SIC sont des systèmes très complexes à modéliser (connaissances) et (2) l'ingénierie de tels systèmes est elle-même complexe. En outre, ils remarquent que leur ingénierie ne s'appuie pas, pour la plupart des cas, sur des techniques de réutilisation qui ont fait leur preuve (framework, pattern,...) et qui permettraient de faciliter celle-ci. Il nous semble nécessaire d'utiliser des techniques de réutilisation éprouvées favorisant l'analyse, la conception et l'implémentation des SIC. Nous proposons ainsi de concevoir et de réutiliser des patterns pour prendre en compte la complexité des SIC et de leur développement. Le concept de pattern est présenté en section suivante.

---

## **Conclusion**

---

*L'objectif de ce chapitre a été de créer une compréhension claire et précise des concepts de coopération, Travail coopérative et Systèmes d'information coopérative. Cette étude a mis en évidence un certain nombre de caractéristiques liées au domaine des SICs, afin de surmonter l'absence d'une théorie de coopération. En présentant une analyse des concepts théoriques des travaux se rapportant à la plate forme de notre contribution « **Méthodologie de validation collaborative des exigences dans les organisations distribuées** » qui sera présentée dans le prochain chapitre.*

---

---

*CHAPITRE 4 :*

*Vers une Méthodologie de Validation  
Des Exigences Collaborative  
Dans  
Les Organisations Distribuées  
-VECOD –*

---

# Chapitre IV :

## vers une Méthodologie de Validation des Exigences Collaborative dans les Organisations Distribuées

### -VECOD -

---

#### Résumé

---

*Notre contribution propose une méthodologie qui présente une démarche collaborative d'un processus de validation basé compétences multi points de vue. Le but de cette méthodologie est double, il s'agit à la fois d'accroître la qualité (c'est-à-dire aboutir à une meilleure adéquation aux besoins de l'utilisateur) et de diminuer la complexité de la tâche. La méthodologie propose un processus subdivisé en trois phases :*

- **Phase d'organisation** : où se passe l'aménagement de travail coopératif à travers trois étapes : (i) **étape de planification** du travail collaboratif, (ii) **étape de qualification** des exigences, suivie de la (iii) **étape d'inspection** des documents et conformité au standard.

- **Phase de vérification des exigences** : au niveau de chaque organisation, s'opèrent des vérifications de la partie qui la concerne, et dont elle a été origine. Ici le travail se passe à travers quatre activités (i) **Activité de Vérification des critères fonctionnels d'aspect non sémantiques** : pour juger la Traçabilité ; la Non redondance ; la Modificabilité ; la Vérifiabilité et la Conformité au standard des exigences. (ii) **Activité de Validation des critères fonctionnels d'aspect sémantiques** : pour évaluer la Complétude ; la Consistance ; la Non ambiguïté ; la Non conflictualité, l'Exactitude ; la Faisabilité et la nécessité des exigences. (iii) **Activité de Validation des critères non fonctionnels** des exigences : cette phase s'achève par (iv) **l'Activité de génération de la QFD** dans laquelle s'effectue l'affectation des poids selon chaque point de vues concernant la partie du travail concerné par toute organisation à part.

- **Phase de validation des besoins** : cette phase fait intervenir la partie prenante utilisatrice guidée par celle qui est réalisatrice. La méthode est multi point de vue et s'effectue comme suit : (i) phase d'**Evaluation des conflits** qui provient des différentes

## I. INTRODUCTION

La complexité croissante des systèmes et l'interdisciplinarité nécessaire à leur réalisation poussent les organisations à recourir aux équipes, car elle semble constituer un moyen pour améliorer la qualité et le temps de réalisation des produits industriels.

De plus en plus, la coopération fait l'objet de beaucoup de travaux de recherche et le nombre de conférences qui y sont consacrées témoigne de l'importance que ce sujet revêt actuellement. Pourtant, elle n'est pas une pratique récente car les gens coopèrent depuis toujours. Cependant, la coopération, telle qu'elle réapparaît de nos jours, n'est pas simplement un travail d'équipe, elle se veut structurée et organisée afin de produire de meilleurs résultats.

Selon Zarour N [118], Le développement d'applications aujourd'hui se caractérise souvent par l'existence de plusieurs domaines d'expertise différents faisant intervenir la coopération de plusieurs organisations supportées par un Système d'Information Coopératif.

Le plus grand nombre de travaux dans le domaine des SIC est consacré à *la conception* Saidane [130] Jacqueline [131] Darses [132] et à la phase d'implémentation, justifiée par le grand nombre d'outils logiciels de communication et d'échange d'information Voir Baudin [133] mais encore très peu d'attention à *l'ingénierie des exigences/besoins*.

Le travail que nous présentons dans cette contribution vise *précisément la validation des exigences* qui n'est souvent pas suffisamment couverte par les méthodes d'IE d'après lulu He et all [79] Tandis qu'Artem Katasonov [136] considère que la plupart des livres présentent la validation comme une liste de "*bonnes pratiques*" et présente la validation des exigences comme un processus hétérogène basé sur l'application d'une grande variété de techniques indépendantes. A son tour Nuseibeh, [78] compare le problème de la validation des exigences au problème de la validation des connaissances scientifiques en référence au cycle (Expérience, Observation, Analyses & Conclusion).

## II. MOTIVATION

Notre principale motivation est de proposer une *méthodologie de validation des exigences collaborative dans les organisations distribuées*. La proposition est basée point de vue selon les compétences cognitives des parties prenantes en guidant leur collaboration tout au long du processus. Ce processus est divisé en plusieurs activités, afin de supporter la plate forme coopérative tout en précisant les rôles et les résultats ainsi que les outils de chaque sous processus.

Nous avons rencontré lors de la fusion des domaines (SIC, IE, validation) les problèmes suivant : des problèmes liés à la plate forme coopérative, des problèmes liés à la validation durant le cycle de vie logiciel, des problèmes liés à la validation durant le processus d'ingénierie des exigences et des problèmes liés à l'ambiguïté du terme validation dans la littérature.

Afin de répondre à ces insuffisances. Nous proposons *VECOD*, avec une démarche méthodologique générale qui englobe l'ensemble des activités requises pour vérifier, valider et faire

évoluer un système apportant une solution économique et performante aux besoins d'un client tout en satisfaisant l'ensemble des parties prenantes qui s'articule autour d'un processus *continu, progressif, collaboratif, distribué, multi point de vue*.

Pour cela nous proposons ce qui suit :

- Une définition de la validation et de la vérification des exigences, prenant en compte la séparation des aspects selon leurs aptitudes sémantiques, quelques critères de qualité à vérifier et à valider, enrichis par d'autres critères résultants du travail coopératif distribué.
- L'utilisation des outils groupware (collecticiels), pour assurer le travail coopératif, selon une dimension spatio-temporelle, la plus adéquate par rapport à la fonctionnalité assurée par chaque étape d'IE, avec quelques recommandations de leurs adaptations avec notre démarche.
- Proposer une démarche plus générale que celle trouvée dans la littérature, et qui traite la validation en tant qu'activité et non en tant que phase, qui présente la validation comme un processus continu, incrémentale, collaboratif, qui se déroule tout au long du processus IE, basé point de vue.
- Déterminer, de façon détaillée, les rôles et les compétences des parties prenantes, les résultats de chaque sous activité du processus de validation, et les moyens permettant sa mise en œuvre.

### III. PRÉSENTATION DE LA SOLUTION

Pour qu'un système puisse se définir comme un processus coopératif et interdisciplinaire (sciences de l'ingénieur, sciences humaines, sciences cognitives, génie logiciel, etc.) de résolution de problème, il est nécessaire :

- ✓ Qu'il s'appuie sur les connaissances, méthodes et techniques issues de la science et de l'expérience ;
- ✓ Qu'il mette en œuvre, définisse, fasse évoluer et vérifie la définition d'un système (ensemble organisé de matériels, logiciels, compétences humaines et processus en interaction) ;
- ✓ Qu'il apporte une solution à un besoin opérationnel identifié, conformément à des critères de qualités ;
- ✓ Qu'il satisfasse les attentes et contraintes de l'ensemble de ses parties prenantes et soit acceptable pour l'environnement ;
- ✓ Qu'il cherche à équilibrer et optimiser l'économie globale de la solution sur l'ensemble du processus de validation.

Nous raisonnons sur la solution des problèmes cités dans la section II comme suit :

#### 1. Les Problèmes Liés à l'ambiguïté du terme dans la littérature

Nous pensons surmonter ce problème par une définition claire et non ambiguë des activités de vérification et de validation des exigences tout en déterminant, ce qui va être *vérifié* et ce qui va être *validé* par rapport au critère de qualité (voir chapitre 2 section 4) et en précisant les rôles de chaque intervenant.

### 1.1. Le rapport entre les activités de vérification & Validation Par rapport aux propriétés à contrôler

Fabbrini et al [135] a proposé quatre types de propriétés linguistiques des exigences (syntaxique, structurelle, pragmatique et sémantique), comme les éléments-clés pour classer les exigences fonctionnelles voir figure 31. La propriété sémantique doit être validée par le biais de l'interaction humaine, mais la non-sémantique (syntaxique, structurelle et pragmatique) peut être vérifiée manuellement ou automatiquement.

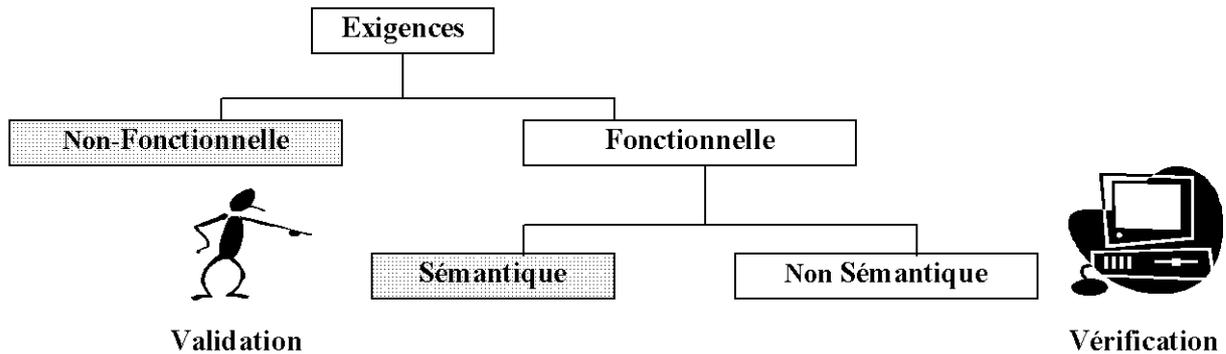


Fig 31. Taxonomie des exigences [134]

Et d'autre par Artem Katasonov [136] a présenté un lien entre les propriétés (voir chapitre2 section 4) et les processus de vérification & validation des exigences comme suit.

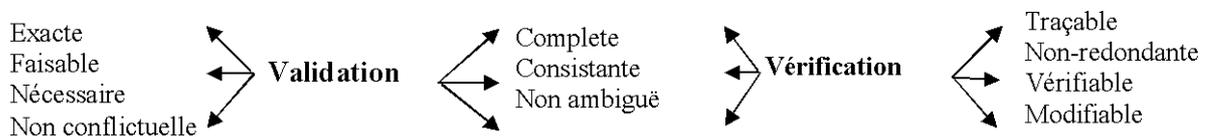


Fig 32. Lien des propriétés par rapport aux VV [136]

### 1.2. Définition (Le quoi ?)

Par enchaînement logique qui provient de ces deux explications de Fabbrini et Artem Katasonov, nous redéfinissons la vérification des exigences comme « l'examen *syntaxique, structurel et pragmatique* des exigences fonctionnelles d'orientation non sémantiques pour garantir ces critères de qualité pour la « Traçabilité, la Non redondance, la Modificabilité et la Vérifiabilité » de façon interne avec des moyens plus au moins composés selon l'aptitude du maitre d'œuvre.

Par distinction à la validation des besoins « qui se charge de qualifier l'aspect sémantique et non fonctionnel, pour certifier la « Complétude ; Consistance ; Non ambiguïté ; Non conflictuelle, Exactitude ; Faisabilité et Nécessité » des exigences où sa validation est externe non définitive qu'après accord par le maitre d'ouvrage d'où la nécessité d'utiliser des moyens plus simples, conviviales et agréables.

### 1.3. Propriétés émergentes du travail coopératif :

En plus des propriétés qui assurent la qualité des exigences et des livrables (voir chapitre2 section 4), il révèle d'autres propriétés et de nouvelles définitions liées à la plate forme coopérative distribuée, telle que :

- **L'identification** : Indiquer l'unicité et l'importance et / ou la stabilité de chaque exigences !, qui est généralement présentée par un indice  $i$  ; ce numéro de l'exigence est un numéro unique, mais dans un environnement distribué, on doit indiquer la source de cette information. Donc L'identification utilisera deux indice  $[i,j]$  où  $i$  présente un numéro de l'exigence par rapport à l'ensemble des exigences et  $j$  identifiera celui qui a émis cette exigence, qui est sensé intervenir plus tard dans sa validation et la vérification.

- **Complémentarité** : dans un contexte où l'expertise se complète par l'intervention d'autres experts, où une exigence reste incomplète, ambiguë jusqu'à l'intervention d'autres informations qui découlent d'autres domaines ou experts. A un niveau d'itération, l'exigence n'est pas valide jusqu'à l'apparition d'autres détails qui justifient le besoin de collaboration afin de compléter le sens des choses.

La valeur estimée de la **complémentarité** qui sera recensée par (la différence entre le nombre des exigences après la fusion et la sommes des différentes exigences selon chaque point de vue) elle mesure le taux de motivation et engagement entre les parties prenantes, et de puis la qualité du travail en groupe.

#### Métrique de la Complémentarité

$$\text{Complémentarité} = \frac{\sum_{i=1}^N \text{Nombre Totale des lignes des QFDs} - \text{le nombre de lignes de QFD après Fusion.}}{N}$$

/  $N$  est le nombre d'organisation

- **Cohérences Globales** : Elles ne doivent pas contredire d'autres exigences établies, après l'intégration globale des exigences.

La valeur estimée de la **cohérence globale** qui sera recensée par le nombre de déficit arrangé par règlement de conflit.

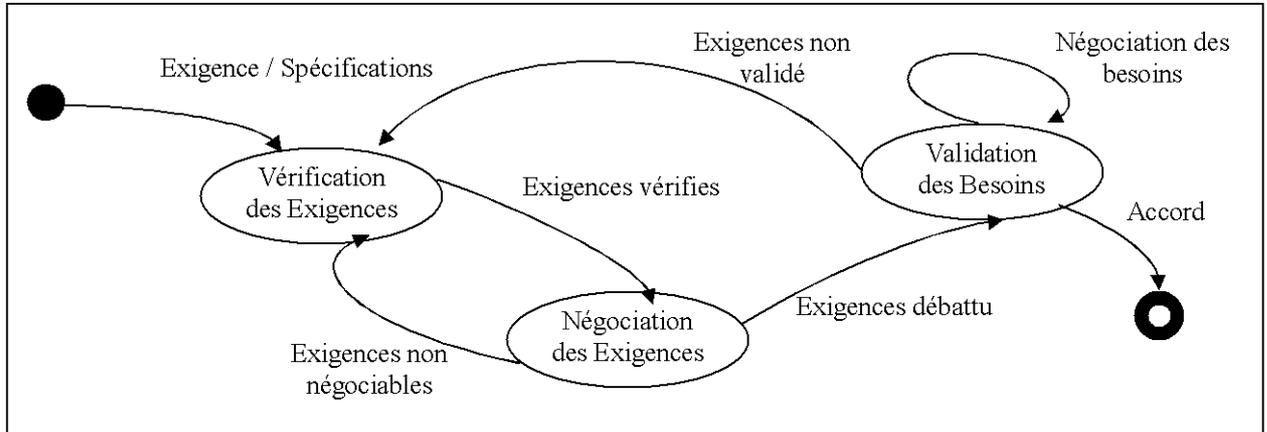
#### Métrique de la Cohérence globale

$$\text{Cohérence Globale} = \frac{\text{Nombre de conflit réglé}}{\text{Nombre de conflit totale}}$$

## 2. Les Problèmes liés à la Validation dans le processus d'ingénierie des exigences

Pour passer du besoin exprimé en langage métier à un modèle complet et cohérent support au développement et partager une compréhension commune de l'architecture du système, il est nécessaire d'utiliser un langage connu permettant de mener une succession de transformations des modèles aux niveaux requis. UML (Unified Modeling Langage) est un langage de modélisation largement répandu et répondant à cet objectif.

Notre vue de la validation des besoins est plus générale que celle trouvée dans la littérature parce qu'elle traite la validation tel qu'un processus de vérification des exigences systèmes par les experts suivie d'une négociation des exigences jusqu'à entente entre ces experts, qui vont passé à la validation des besoins par les clients *figure 33*.



**Fig 33.** Processus de validation en Ingénierie des exigences

Le processus de validation des exigences de la méthodologie VECOD sera présenté comme un processus progressif où chaque sous activité est affectée de façon argumentée à un ou plusieurs intervenants, qui fonctionne selon le diagramme de cas d'utilisation voir figure 34 :

La sélection des participants se fait d'après certains critères qui diffèrent selon les phases On s'attachera en particulier à avoir différents **points de vue** et à faire appel à des **compétences** multiples complémentaires. De manière générale, il est préférable de choisir les participants en dehors de l'équipe de développement. Ils seront ainsi plus objectifs car moins impliqués dans le projet. Il convient ensuite d'attribuer un rôle à chacun. Il est bien évident qu'un participant peut éventuellement tenir plusieurs rôles.

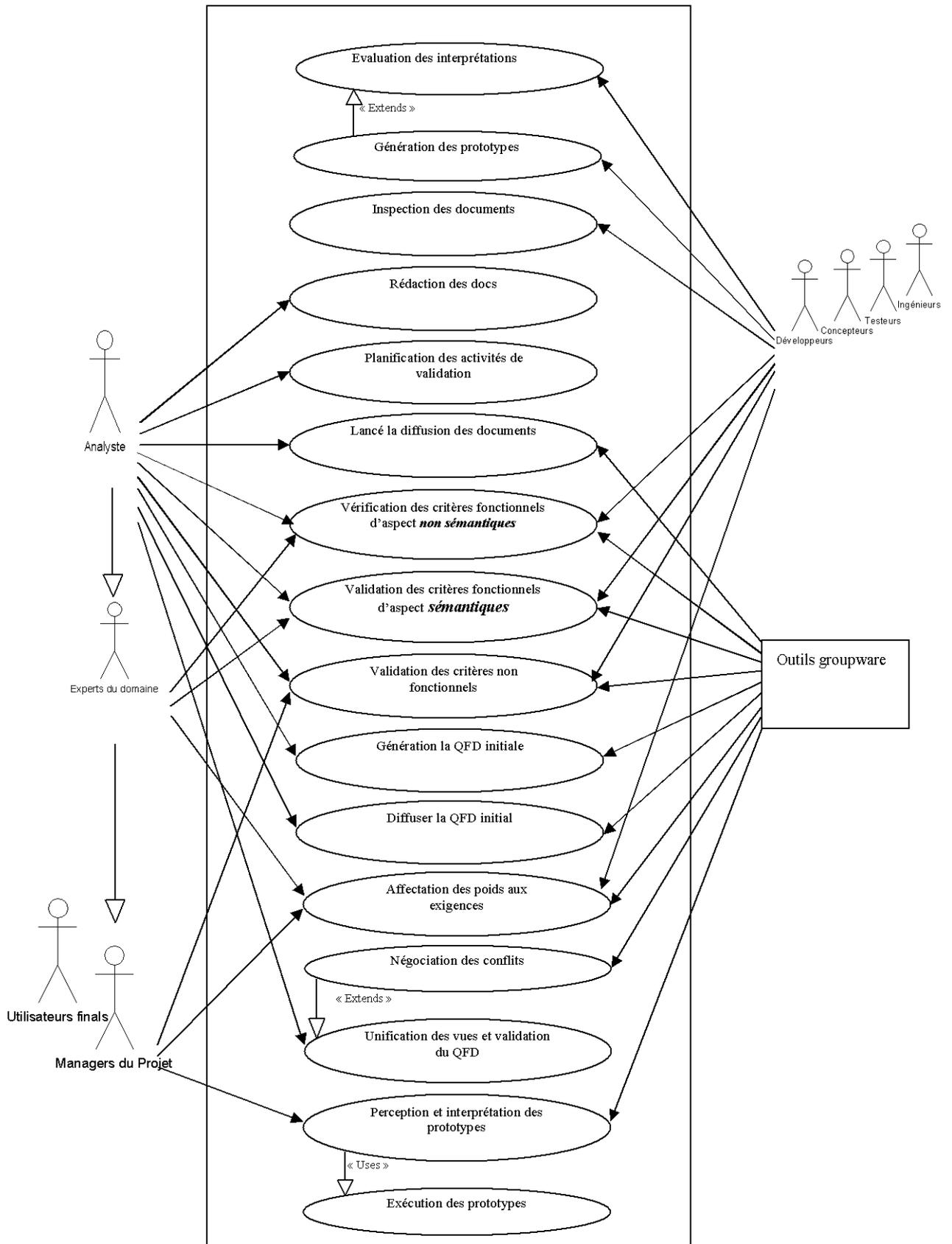


Fig 34. Diagramme cas d'utilisation de la démarche

Ce tableau présente les rôles des parties prenantes, leurs compétences et quand ils interviennent dans le processus de validation.

Partie Prenante	Intervention	Rôles	Compétences & Expertises
<b>Analyste</b>	Processus Complets	C'est un modérateur, il dirige les débats. Il prépare les réunions et veille à l'enchaînement des étapes. Il veille pendant la conduite des débats au maintien des objectifs et s'attache à ne pas négliger les facteurs humains. Il préside à la prise de décision.	Analyse des SI Animation & Communication Ordonnance, Décision, Négociation
<b>clients</b>	Validation	Préciser les besoins Lire les exigences pour vérifier la correspondance avec les besoins.	Communication
<b>Managers du Projet</b>	Inspection	Utiliser le cahier de charge pour planifier l'offre et le processus de développement du système	Domaine Du Problème Gestion cout, délai, technique Communication
<b>Experts du Domaine</b>	Validation	Préciser les exigences Fonctionnelles	Domaine des Problèmes & Solutions Communication
<b>Utilisateur Final</b>	Validation	Étaler les exigences Fonctionnelle & Non Fonctionnelle Organisation, Contexte, Contrainte	Domaine des Problèmes & Solutions Connaissance de l'outil informatique (Systèmes d'exploitation, software, hardware)
<b>Ingénieurs Système &amp; Développeur</b>	Vérification	Utiliser les exigences pour comprendre le système en cours d'élaboration	Communication IHM
<b>Ingénieurs test Système</b>	Vérification	Utiliser les exigences pour développer des tests de validation pour le système	Test de validation Communication
<b>Ingénieurs de maintenance Système</b>	Maintien de la validation	Utiliser les exigences pour aider à comprendre le système	Communication
<b>les concepteurs (y compris de versions précédentes)</b>	Vérification	Détailler et compléter les exigences	Communication Domaine de Solution

TABLE XIII. ROLES ET COMPETENCES DES PARTIES PRENANTES

### 3. Les problèmes liés à la plate forme coopérative

L'orientation de la démarche que nous proposons est multipoint de vue, visant à établir une démarche d'ingénierie des exigences des Systèmes d'information Coopératifs. Elle vise à établir une démarche d'un processus pour une organisation virtuelle, composée de plusieurs organisations géographiquement dispersées où chaque organisation est supportée par son SI voir figure 35. Elle utilisera les moyens apportés par la nouvelle technologie de l'Information et de la Communication NTIC (Surtout les outils d'Internet et d'extranet), à ce niveau elle utilisera les outils d'un Groupeware, qui peuvent soutenir le processus d'ingénierie (du point de vue moyens de communication), que l'approche vise à développer. Pour permettre à des utilisateurs géographiquement éloignés de travailler en équipe. Le travail en équipe peut se concrétiser par le partage d'information, ou bien la création et l'échange de données informatisées.



Fig 35. Processus coopératif

#### **IV. UN META-MODELE POUR LA METHODOLOGIE VECOD**

Après la modélisation de notre démarche, Présentant sa partie fonctionnelle statique par le diagramme cas d'utilisation *figure 33*. Nous passons à la modélisation de la partie structurelle par un méta modèle UML *figure 36*.

Dans l'ingénierie des systèmes IS, diverses parties prenantes coopèrent à des stades différents, dans le processus global et celui de l'IE, en utilisant des artefacts variés sous formes de documents en papier électroniques ou des modèles plus au moins formelles, selon le niveau de détail atteints.

Les parties prenantes qui interviennent dans le processus d'IE font partie de deux classes. L'une est réalisatrice (maître d'œuvre), présentée par les analystes, les concepteurs, les ingénieurs systèmes, etc. Qu'ils ont des aspects communs avec les développeurs (expériences dans le développement, travail de groupe, etc.), Leurs rôles est la réalisation du produit logiciel. L'autre classe est utilisatrice (maître d'ouvrage), présentée par l'utilisateur final du produit logiciel, le manager du projet, l'expert du domaine, etc.

Toute au long du processus d'IS, des documents sont utilisés pour guidé et/ou marqué le niveau atteints du projet. Commençons par le document qui contiens les besoins du client jusqu'au document qui guide la mise en service et la maintenance du produit. Au stade exigences, les documents utilisés et dérivés peuvent se classé en deux catégories, document des besoins décrie par le maître d'ouvrage contenant les besoins fonctionnelles et non fonctionnelles, et un référentiel d'exigences rédigé par l'analyste qui se compose des exigences systèmes fonctionnelles et non fonctionnelles. Les exigences fonctionnelles sont évaluées par la partie réalisatrice qui vérifie les critères non sémantique et valide les critères sémantique.

La matrice QFD<sup>4</sup> généré par l'analyste à partir du référentiel vérifié, et dont le maitre d'œuvre et le maitre d'ouvrage lui affectes des poids. Le développeur, en utilisant les résultats de la matrice et un langage de prototypage, réalise le prototype qui sera exécuté et évalué par la partie utilisatrice à travers des outils groupeware. Ensuite, une interprétation guidera la validation afin d'agrée, modifier ou refuser le prototype selon des critères sémantiques et non sémantiques.

---

<sup>4</sup> Quality Function Deployment

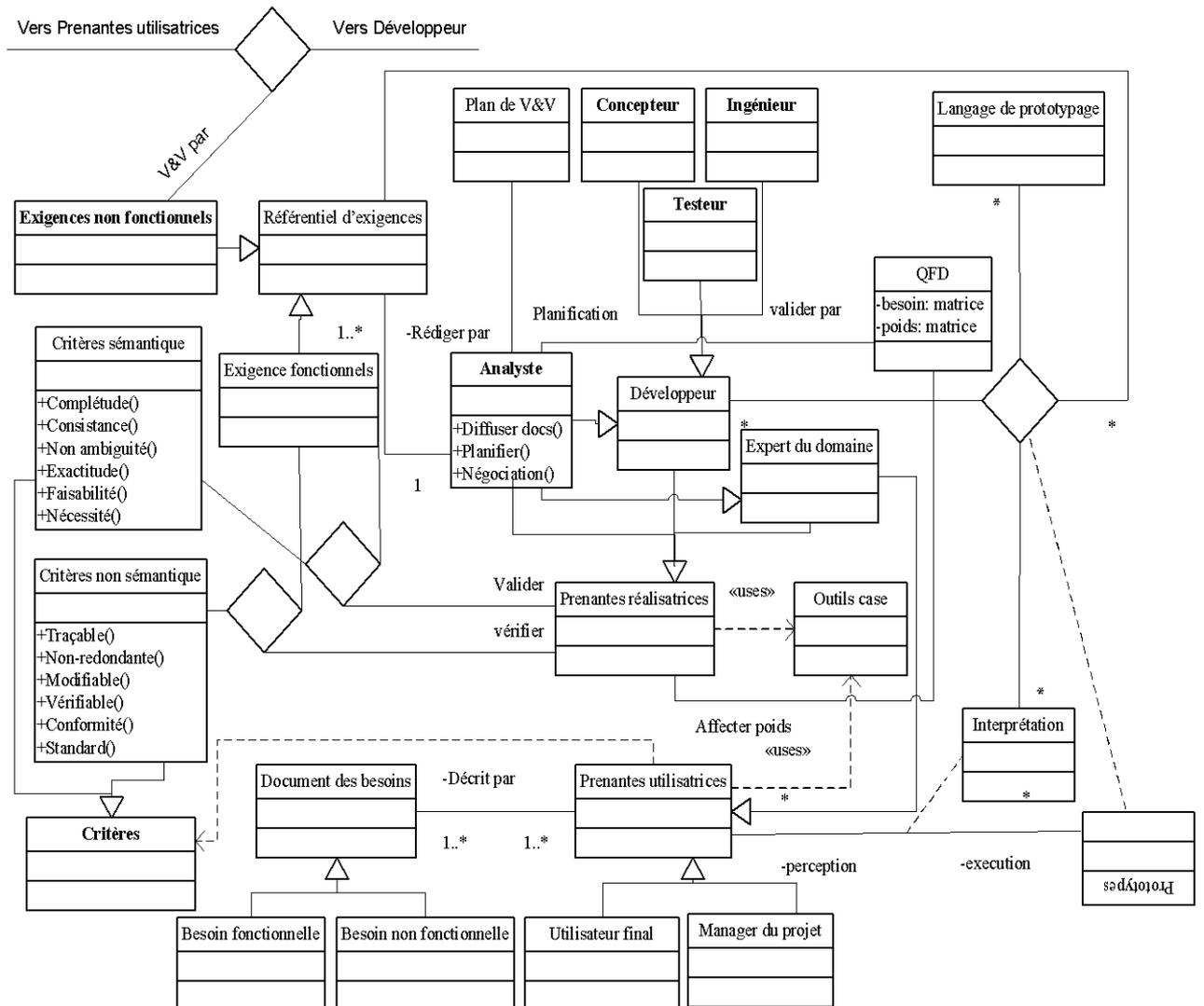


Fig 36. Meta-Modèle pour la Méthodologie VECOD

## V. PRESENTATION DES DIFFERENTES PHASES DU PROCESSUS DE LA VALIDATION COLLABORATIVE

*Après avoir résolu les problèmes liés à la Validation dans les systèmes d'information, nous présentons la démarche de façon détaillée afin de répondre à la question (Qui fait Quoi Quand Comment Où avec quel moyen).*

L'idée de ce travail se focalise sur les étapes que va prendre un ensemble de documents (qui renferment des exigences) pour qu'elles soient validées par l'utilisateur final et /ou une partie ou l'ensemble des parties prenantes qui sont réparties géographiquement.

*Son fonctionnement dynamique* est schématisé dans le diagramme d'activité de la figure 37, le processus globale passe par plusieurs étapes faisant intervenir diverses parties prenantes ; Pour cela,

nous décomposons l'activité de validation par les étapes suivantes en vue de l'adapter à un contexte coopératif distribué.

*Notre démarche décompose la validation en trois phases figure 37 qui sont :*

- **Phase d'organisation** : où se passe l'aménagement de travail coopératif à travers trois étapes : (i) étape de **planification** du travail collaboratif, (ii) étape de **qualification** des exigences, suivie de la (iii) étape d'**inspection** des documents et conformité au standard.

- **Phase de vérification des exigences** : au niveau de chaque organisation, s'opèrent des vérifications de la partie qui la concerne, et dont elle a été origine. Ici le travail se passe à travers quatre activités (i) **Activité de Vérification des critères fonctionnels d'aspect non sémantiques** : pour juger la Traçabilité ; la Non redondance ; la Modificabilité ; la Vérifiabilité et la Conformité au standard des exigences. (ii) **Activité de Validation des critères fonctionnels d'aspect sémantiques** : pour évaluer la Complétude ; la Consistance ; la Non ambiguïté ; la Non conflictualité, l'Exactitude ; la Faisabilité et la nécessité des exigences. (iii) **Activité de Validation des critères non fonctionnels** des exigences : cette phase s'achève par (iv) l'**Activité de génération de la QFD** dans laquelle s'effectue l'affectation des poids selon chaque point de vues concernant la partie du travail concerné par toute organisation à part.

- **Phase de validation des besoins** : cette phase fait intervenir la partie prenante utilisatrice guidée par celle qui est réalisatrice. La méthode est multi point de vue et s'effectue comme suit : (i) phase d'**Evaluation des conflits** qui provient des différentes QFD, où une discussion est réalisée lors de la phase de **négociation des conflits** en (i<sub>1</sub>), suivie d'un débat d'entente dans la phase d'**unification des vue** en (i<sub>2</sub>) après satisfaction totale des différents collaborateurs, et pour finir une projection de toutes les informations contenues dans la QFD vers un prototype Logiciel exécutable par le maitre d'ouvrage, présenté dans (ii) la phase de **génération de prototype**.

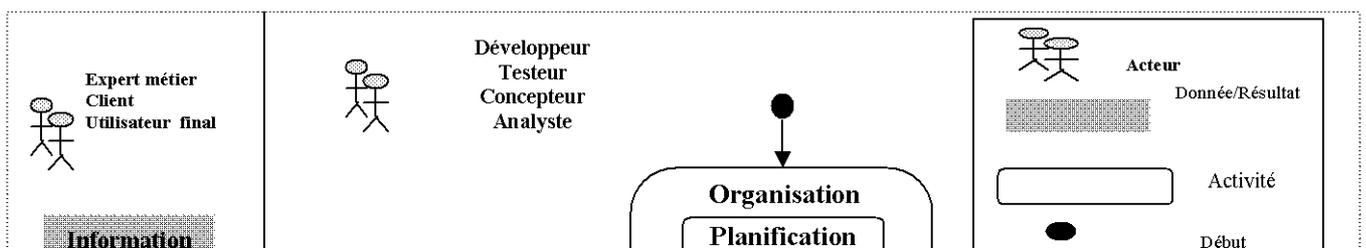


Fig 37. : Diagramme d'activité du fonctionnement de la méthodologie VECOD.

### **1. Phase d'organisation**

Comme plusieurs organisations unissent leurs compétences et ressources pour répondre à une opportunité qu'elles n'auraient pu prendre en charge seules, il est nécessaire à ces organisations de mettre en place une infrastructure leur permettant de partager des documents, travailler et communiquer facilement tout comme le feraient des employés d'une même organisation et ce malgré des contraintes géographiques.

L'objectif essentiel de cette phase est de :

- (1). Préparer l'environnement de travail, fixer le calendrier, Affecter les rôles
- (2). Procéder à la collecte des documents
- (3). Avoir les artefacts appropriés pour le suivi du processus de validation.
- (4). La diffusion des documents : Après que l'organisation virtuelle ait été créée, les rôles affectés et les documents préparés, il ne reste qu'à les diffuser.

### 1.1 La Planification

Ces organisations virtuelles s'appuient grandement sur les technologies de l'information et de la communication qui offrent la possibilité d'établir rapidement une plate-forme reliant chaque entreprise et permettant ainsi, la coordination et la collaboration de chaque partenaire indépendamment de son lieu de travail, supervisée par l'analyste qui affecte les rôles aux parties prenantes.

- Pour les parties prenantes utilisatrices et exploitantes du système, elles ont pour objectif d'assurer l'adéquation de la solution aux besoins sous tous les aspects (fonctionnalités, performances, économie, sécurité) pour toutes les situations d'exploitation du système.
- Pour les parties prenantes réalisatrices, elles ont pour objectif de conduire à un bon compromis entre les enjeux et contraintes concernant tant la solution que le projet (besoins, attentes, performances, contraintes techniques et industrielles, coûts, délais et risques).

L'analyste est la personne clé dans le processus de **Planification** dont le rôle de **Modérateur** fait de lui le responsable de la sélection de l'équipe qui veille à ce qu'elle effectue ses tâches conformément à chacun et à toutes les activités du processus.

Ce processus détermine les rendez-vous des réunions à travers l'envoi de Mail ou l'utilisation d'agenda partagé par tous les membres de l'équipe d'ingénierie (membres de l'organisation virtuelle).

### 1.2 La Qualification

C'est ici que commence le travail de l'analyste, qui effectue :

- La collection des documents et rend les documents produits des activités précédentes, compréhensibles par toutes les parties prenantes.
- -La préparation des artefacts d'évaluations des exigences

## Fiche technique de chaque exigence : inspiré du modèle VOLER<sup>5</sup>

<p>Numéro de l'exigence : <b>identifiant unique</b></p> <p>Spécification : <b>objectif de l'exigence, en une phrase, généralement sous la forme « le produit devrait faire ceci pour telle personne »</b></p> <p>Type d'exigence : <b>référence au modèle</b></p> <p>Critère de satisfaction : <b>une mesure de l'exigence qui permet de tester si la solution proposée remplit l'exigence initiale.</b></p> <p>Contentement du maître d'ouvrage : <b>degré de contentement du maître d'ouvrage si le produit final satisfait cette exigence. De 1 (pas intéressé) à 5 (très content).</b></p> <p>Exigences dépendantes : <b>listes d'exigences dont l'implémentation dépend de l'implémentation de celle-ci.</b></p> <p>Documents relatifs : <b>référence à des documents qui illustrent et expliquent cette exigence.</b></p> <p>Historique : <b>date de création, modification, destruction.</b></p>	<p>Origine : <b>qui a émis cette exigence ?</b></p>
---	---

### Exemple

<p>Numéro de l'exigence : <b>26</b></p> <p>Spécification : <b>nous devrions enregistrer l'heure à laquelle la panne est survenue.</b></p> <p>Type d'exigence : <b>5</b></p> <p>Critère de satisfaction : <b>on peut connaître l'heure de survenue de la panne.</b></p> <p>Contentement du maître d'ouvrage : <b>3</b></p> <p>Exigences dépendantes : <b>(25,30)</b></p> <p>Documents relatifs : <b>Fiche de suivi de panne</b></p> <p>Historique : <b>date de création : 01-01-2006 ; date de modification : nul ; date de destruction : nul</b></p>	<p>Origine : <b>service technique</b></p>
--	---

→**Check-list** : tel que

Questions Check-list	objectif
Chaque exigence est facilement identifiée?	La traçabilité, la conformité aux normes
Est-ce que les termes spécifiques sont définis dans le glossaire	Compréhensibilité, claire
Est-ce que les exigences des individus utilisent le même terme de différentes façons?	Ambiguïté
Si une exigence fait référence à d'autres, sont-elles décrites dans le document?	Complémentarité
Est-ce que les exigences liées sont regroupées?	Organisation
Existe-il des contradictions dans les exigences?	Redondance
Est-ce que vous devez examiner d'autres exigences pour comprendre ce qu'elles signifient?	Complète

→**Grille d'évaluation** : nous proposons une grille avec la forme suivante pour aider à marquer les critères satisfaits des non satisfaits pour chaque exigence à part.

Identifiant de l'exigence	vérifiable	modifiable	Traçable	Non-redondante	Complète	Consistance	Non ambigu	Non conflictuelle,
[id, org]	y/n	y/n	y/n	y/n	y/n	y/n	y/n	y/n

<sup>5</sup> <http://www.volere.co.uk/template.htm>

### 1.3 L'inspection

L'équipe d'ingénierie des exigences présentée par *la partie prenante réalisatrice* se réunit en ligne pour inspecter les documents. Chacun collabore selon son expertise et compétence (voir tableau 1) ; la réunion se termine alors par la formulation d'une *Check-list* qui guidera le reste du processus.

L'examen critique porte sur tout document livrable interne ou non, prévu dans le cycle de vie et le plan qualité. Il vise à fournir un point de vue indépendant de l'auteur du document. Il permet de réduire le coût du produit et fournit des mesures précises pour la gestion de projet.

L'examen critique de document est utile en validation et en vérification. Les propriétés sont contrôlées par rapport à des critères objectifs de :

- ✓ **la forme**: précision, satisfaction des normes définies dans le plan qualité et le plan projet
- ✓ **le fond**: complétude, cohérence, (références imprécises ou inexistantes),

L'examineur doit juger en fonction des bonnes pratiques, des usages et de l'état de l'art du domaine.

En vue de la mise en évidence d'erreurs, l'examineur est amené à :

Reconstituer le raisonnement de l'auteur,

S'assurer de sa cohérence,

Identifier:

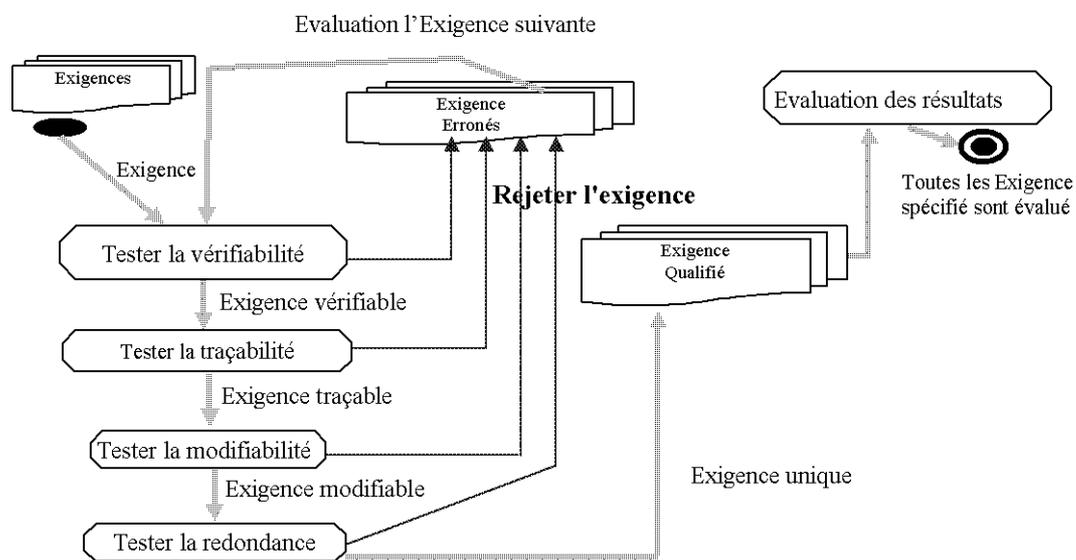
- les mauvaises interprétations de documents de référence,
- les critères de qualité non observés,
- les hypothèses erronées.

## 2. Phase de vérification des exigences

Au niveau de chaque site et hors ligne, commence une inspection spécialisée (selon la partie des exigences qui les concerne, en terme d'expertise et origine de l'exigence). Le but est **d'assurer l'aptitude des exigences à leur spécification** sur tous les sites (membre de l'organisation virtuelle). Cette méthode est Nommée *N\_Fold* de Martin et all [72] où différentes équipes inspectent les mêmes documents. **Notre contribution** va utiliser cette méthode avec des équipes hétérogènes éloignées géographiquement. Ce qui fait d'elle une nouvelle utilisation de cette méthode, qui n'a été à notre connaissance utilisée que dans des situations où les membres ont les mêmes compétences ou des compétences proches (voir chapitre 2 V.2,b). Ici nous l'utilisons pour une collaboration et/ou négociation entre membres d'équipe avec des compétences complémentaires pour bénéficier des expertises de chaque organisation à part sans qu'elles soient obligées de changer et/ou d'apprendre d'autres techniques et artefact (utilisé dans le reste des organisations). Pour le problème de la langue, nous suggérons des documents rédigés en anglais avec la possibilité d'utiliser des glossaires et des traducteurs automatiques. C'est l'expérience des membres qui joue un grand rôle et l'intervention des managers du projet est intéressante pour signaler seulement quelques grandes lignes d'ordre général.

Ici le travail se passe à travers quatre processus. (i) **Processus de Vérification des critères fonctionnels d'aspect non sémantiques**, (ii) **Processus de Validation des critères fonctionnels d'aspect sémantiques** (iii) **Processus de Validation des critères non fonctionnels des exigences** : cette phase s'achève par (iv) **Processus de génération de la QFD** dans laquelle s'effectue l'affectation des poids selon chaque point de vues concernant la partie du travail concerné par toute organisation à part.

**2.1 Processus de Vérification des critères fonctionnel non sémantiques** : ces critères qui se résument dans l'examen *syntaxique, structurel et pragmatique* des exigences fonctionnelles, qui synthétise les propriétés Traçable ; Non-redondante ; Modifiable ; Vérifiable et Conformité au standard



**Fig 38.** Processus de Vérification de quelques critères non sémantiques

Notre contribution va plus loin pour toucher quelques conseils de bonne formulation des exigences afin qu'ils puissent être testables. La Totalité des approches et techniques s'arrête à un stade de détection.

- **Tester la vérifiabilité** : Une exigence sera finalement vérifiée par tests d'exécution et analyse des résultats. Cette méthode implique que cette exigence doit manifester un comportement qui peut être vérifié par des tests. Des valeurs spécifiques sont nécessaires pour les tests et les exigences doivent contenir ces valeurs spécifiques. Des exigences qualitatives doivent spécifier des valeurs données par le client ou en accord avec lui en référence à la formulation d'objectifs selon bloom [151]. Définir des objectifs permet de structurer le concept.

### Exemple

**Expression testable:** "Le système doit répondre à l'entrée des variations de température en 1 seconde."

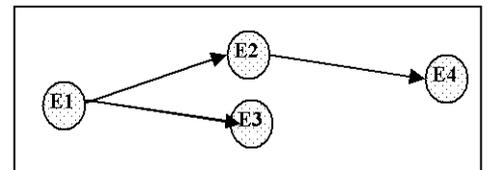
**Expression non testable:** "Le système doit avoir une haute disponibilité."

Commentaire: Cette exigence est insuffisante. Comment cela peut-il être testé? Le client doit spécifier une valeur comme suit. 99,999% de disponibilité seront nécessaires.

- **Tester la traçabilité :** les Exigences Traçables sont particulièrement marquées et sont rédigées d'une manière structurée et à grain fin, par opposition aux grands paragraphes narratifs ou listes à puces. Pour régler ce problème, l'utilisation d'un graphe pour relier chaque exigence par ses pré successeurs et ses successeurs. On devait être en mesure de relier chaque exigence à sa source.

**Exemple**

Exigence	Successeurs	Pré-successeurs
E1 : Etablir Facture	E1, E2	
E2 : Calcule Prix Totale	E4	E1
E3 : Calcule TVA		E1
E4 : Quantité de produit		E2



- **Tester la modifiabilité :** Une exigence sera en mesure d'être révisé en cas de besoin et de maintenir un historique des changements apportés à chaque besoin. Cela signifie que chaque exigence soit **uniquement identifiée** et exprimée séparément des autres exigences de sorte qu'on puisse y faire référence sans ambiguïté. Pour régler ce problème, *on peut la rendre plus modifiable en l'organisant de telle sorte que les exigences connexes soient regroupées ensemble, et en créant une table des matières, index, et des listings croisés de référence.*

- **Tester la redondance :** chaque exigence se rapportant des objectifs communs à une autre doit être éliminée ou fusionnée afin de la compléter.

**Exemple**

Exigence = Vérification du stock.  
 Exigence = Contrôle du stock.  
 « Vérification du stock » et « contrôle du stock » et la même fonction, puisque après discussions avec la partie utilisatrice, on a trouvé qu'ils ont la même spécification.

**2.2 Processus de Validation des critères fonctionnels d'aspect sémantiques**

Ces critères se résument dans l'examen sémantique des exigences fonctionnelles qui synthétise les propriétés Complétude ; Consistance ; Non ambiguïté ; Exactitude ; Faisabilité et Nécessité.

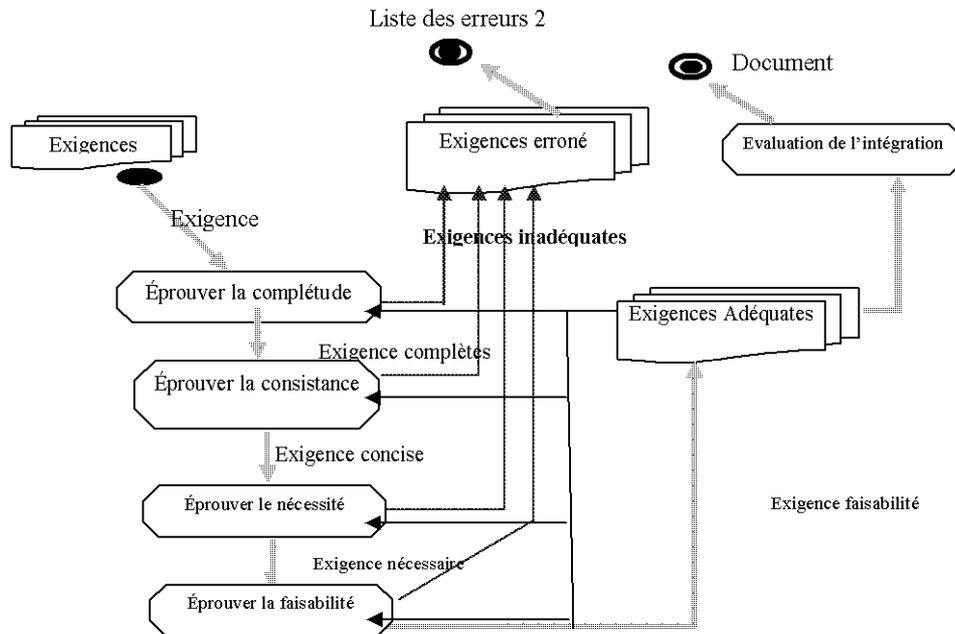


Fig 39. Processus de validation de quelque critère sémantique

- **Evaluer la complétude** : Aucune exigence ou information nécessaire ne doit manquer. L'exhaustivité est aussi une caractéristique désirée d'une exigence individuelle. Il est difficile de repérer des exigences lorsqu'elles viennent à manquer et/ou sont absentes. Organiser les exigences hiérarchiquement pour aider les vérificateurs à comprendre la structure de la fonctionnalité décrite, de sorte qu'il sera plus facile pour eux de dire si quelque chose manque.

Chaque exigences est formulé par une fonction  $f(x) \rightarrow y$  en spécifiant les entrées et les sorties de cette dernière.

#### Exemple

Calcul de prix totale est une fonction tel que :  $f(\text{Prix unitaire}, \text{Quantité}) = \text{Prix Totale}$ .

**Prix Totale = Prix unitaire \* Quantité**

Les entrées sont : *Prix unitaire et Quantité*

La sortis est le *prix totale*

- **Evaluer la consistance** : Des exigences uniformes ne sont pas incompatibles avec les exigences d'autres logiciels ou avec les exigences de niveau supérieur (système ou l'activité). Les désaccords entre les exigences doivent être résolus avant que le développement ne puisse continuer. Il est difficile de savoir ce qui est correct tant que des recherches ne sont pas effectuées. Il faut cependant être prudent lorsqu'on modifie des exigences car des incohérences peuvent glisser sans que l'on ne s'aperçoive surtout si on examine seulement les changements particuliers sans nous pencher sur les exigences connexes.

On ne peut pas savoir qui (le cas échéant) est correct jusqu'à ce que nous fassions quelques recherches. On doit être prudent lors de la modification des exigences, comme des incohérences peuvent se glisser si nous vérifiant seulement les particuliers du changement et non les autres exigences connexes.

- **Evaluer la Nécessité** : Elles doivent porter sur des éléments nécessaires, c'est-à-dire des éléments importants du système que d'autres composants du système ne pourraient pas compenser.

Exigence = « Approvisionnement du stock »  
 Condition atteindre le niveau = stock minimum.  
 La valeur du stock minimum est nécessaire pour déclencher l'opération d'approvisionnement

- **Evaluer la Faisabilité** : réalisables avec le budget, les savoir-faire, les technologies matérielle et logicielle existante.

Exigence = « Achat online »  
 Connexion au système bancaire  
 Un service de sécurité des transactions existant  
 Un système distribué

} → Achat online (faisable)

### 2.3 Processus de Validation des critères non fonctionnels

Désormais, les propriétés non fonctionnelles ne sont pas détaillées dans ce travail.

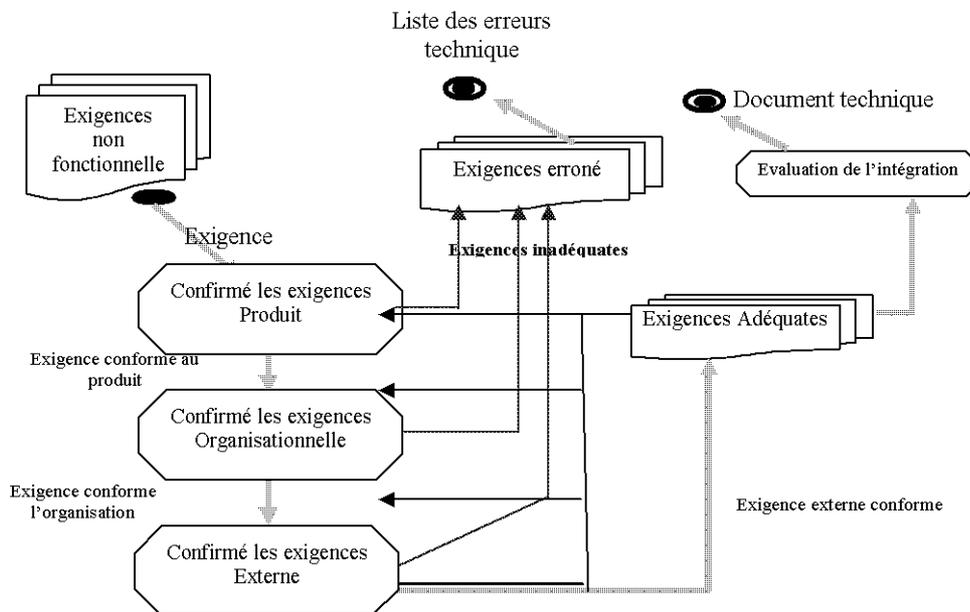


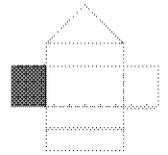
Fig 40. Validation des critères non fonctionnels

## 2.4 Processus de génération de la QFD

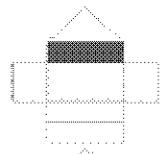
Le QFD<sup>6</sup>, s'agit d'une représentation matricielle des voix du client (les attentes formulées par des personnes sondées) et des fonctions techniques. On présente le QFD sous la forme d'une matrice, dont chaque ligne correspond à une voix du client, et chaque colonne à une solution technique. Les valeurs de cette matrice représentent la valeur d'incidence d'une solution par rapport à une voix.

La méthode du QFD se déroule alors en 6 étapes :

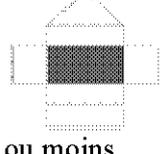
### 1. Identifier les besoins des clients (le quoi)



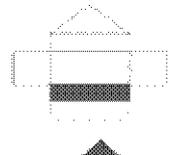
### 2. Définir les spécifications du produit à offrir (le comment)



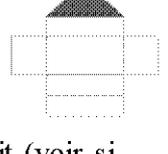
3. Établir la relation entre les spécifications et les besoins des clients (le comment par rapport au quoi) Dans cette étape, l'équipe ***met en rapport*** les attentes des clients avec les moyens de réponse proposés, en considérant la capacité de chaque solution à satisfaire plus ou moins la clientèle. Il faut alors remplir la partie relation matricielle à l'aide de chiffres, ou de symboles légendés ( par ex : 9 : satisfaction élevée, 3 : satisfaction moyenne, 1 : satisfaction faible ). On peut alors juger qualitativement de la contribution de chaque moyen et les prioriser.



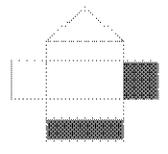
### 4. Cibler le niveau de performance technique des caractéristiques (le combien)



5. Déterminer les relations entre les caractéristiques (le comment par rapport au comment) L'équipe évalue le degré d'interrelation entre les caractéristiques du produit (voir si deux caractéristiques entrent en conflit ou sont redondantes). Puis elle analyse les diverses corrélations et signale les communications à réaliser en interne, entre les services de l'entreprise attachés au développement du produit.



6. Comparer le produit avec ceux des concurrents (la comparaison) Enfin, il ne reste plus qu'à évaluer la satisfaction des clients, ainsi que les solutions technologiques élues par ***rapport à la concurrence***.



<sup>6</sup> Quality Function Deployment est né dans les années 50, au sein de la firme japonaise Toyota

Ce processus de réalisation de la QFD est effectué de façon automatique par le biais d'une feuille de calcul Microsoft Excel ou par des logiciels spécialisés (SQFD<sup>7</sup> ou QFDcapture<sup>8</sup>, etc.).

Le compte tenu d'une cellule  $C$  avec une paire (exigence, spécification), le modèle spécifie la valeur de corrélation correspondant,  $R \times S \rightarrow \{0, 1, 3, 9\}$ , où  $R$  est l'ensemble des exigences,  $S$  est l'ensemble des spécifications et  $\{0, 1, 3, 9\}$ , est l'ensemble des valeurs de corrélation possible (une valeur de zéro correspond à une cellule vide). Les valeurs  $\{0, 1, 3, 9\}$ , sont utilisés à l'équivalence adoptée « nulle », « faible », « moyenne » et « fort » corrélations.

Au début,  $C = 0, \forall r \in R; \forall s \in S$ ; ce qui signifie que toutes les cellules sont vides.

La première offre est la première des parties prenantes en spécifiant une valeur non nulle  $C$ , tandis que la valeur précisée est l'offre initiale. Notre modèle associe l'offre initiale à une question, à définir comme,  $\text{Choix} = (\text{poids}; \text{origine})$ ; où l'offre initiale  $\in \{1; 3; 9\}$ , origine  $\in ST$ , et  $ST$  est l'ensemble des parties prenantes.

L'offre initiale est publique et son instanciation ouvre la possibilité pour d'autres parties prenantes d'exprimer leurs propres préférences. Toutes les parties prenantes qui attribuent une valeur à la cellule seront traitées en tant que partisans ou les opposants à l'offre initiale. Seules les parties prenantes participant à la définition auront la possibilité d'appartenir à  $ST$ .

Notre motivation pour le choix de **La méthode QFD (Déploiement des Fonctions Qualité) ou maison de qualité** .c'est une technique pour confirmer le mouvement de la qualité. Elle considère le client comme *l'acteur essentiel qui permet le Travail d'équipes*, Basé sur *L'intention du client*, Supporte le travail *simultané*, Amélioration de la communication/ documentation.

Enfin, sa finalité, c'est la **corrélation** voir *figure 41* entre plusieurs valeurs qui définissent différents point de vues, ce qui est une amélioration en comparant avec les calculs statistiques (moyenne, variance, écart type), où la corrélation fait rapprocher les points de vues pour en prendre seulement les plus proches avec une participation positive.

---

<sup>7</sup> Software Quality Function Deployment.

<sup>8</sup> <http://www.qfdcapture.com>

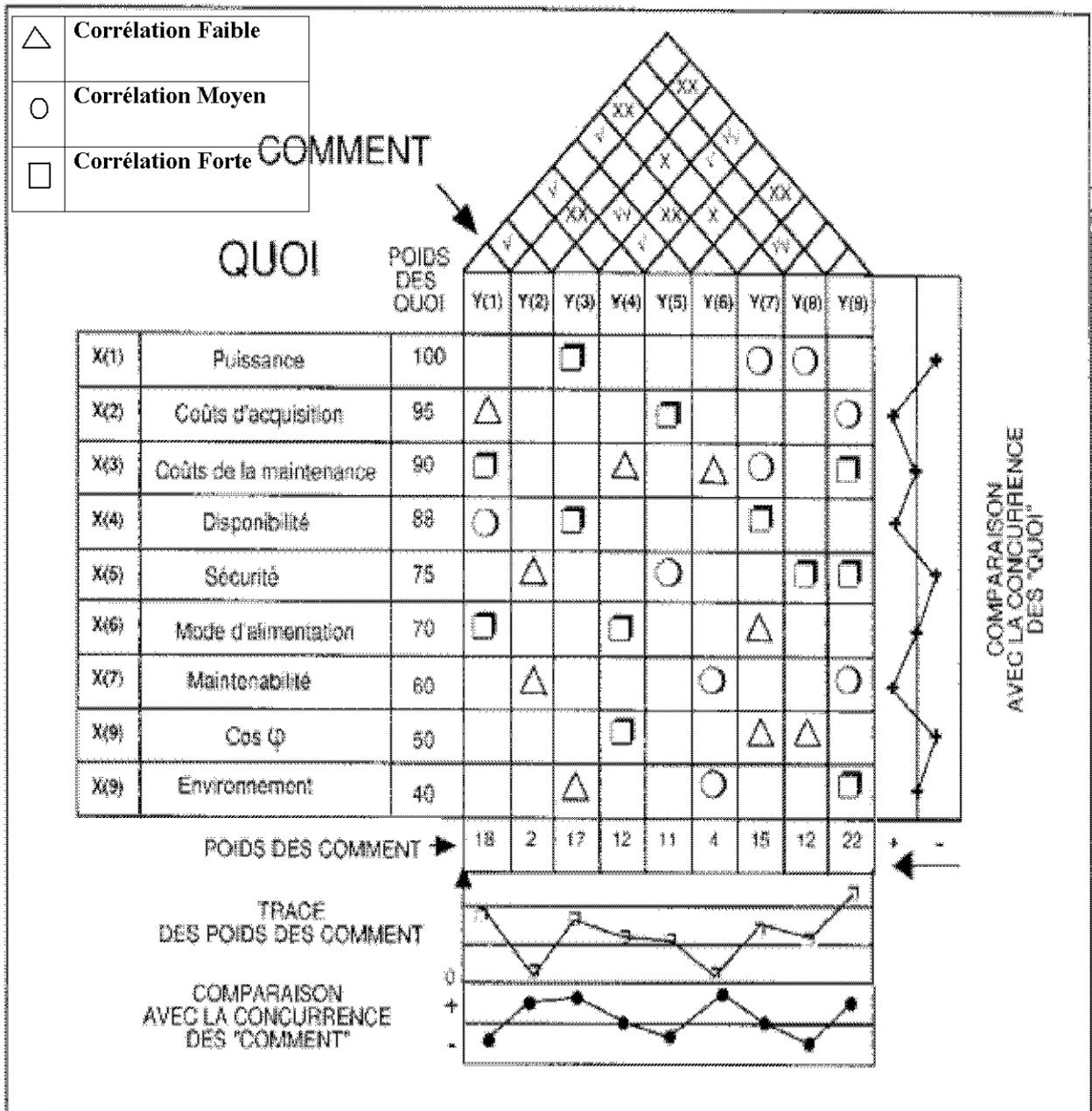


Fig 41. Exemple de QFD<sup>9</sup>

### 3. Processus de validation des besoins

Après la connexion de tous les sites participants (membres de l'organisation virtuelle) dans le projet suite au message reçu : une table ronde virtuelle est conçue à l'aide d'un Groupware de type synchrone, fermés, invisibles. Cette plate-forme de travail collaboratif permet aux groupes de commencer l'unification des QFD afin d'en résulter une seule unifier.

#### 3.1 Evaluation des conflits

<sup>9</sup> <http://www.cyber.uhp-nancy.fr/demos/MAIN-017/2lanalysed/2-3-2analy.html>

Suivant les étapes de négociations qui peuvent se résumer en :

- Premièrement, la *détecter* quand les exigences utilisateurs sont conflictuelles.

L'entrée de cette phase sera un nombre de QFDs chaque QFD concerne la partie du problème vérifié par une organisation.

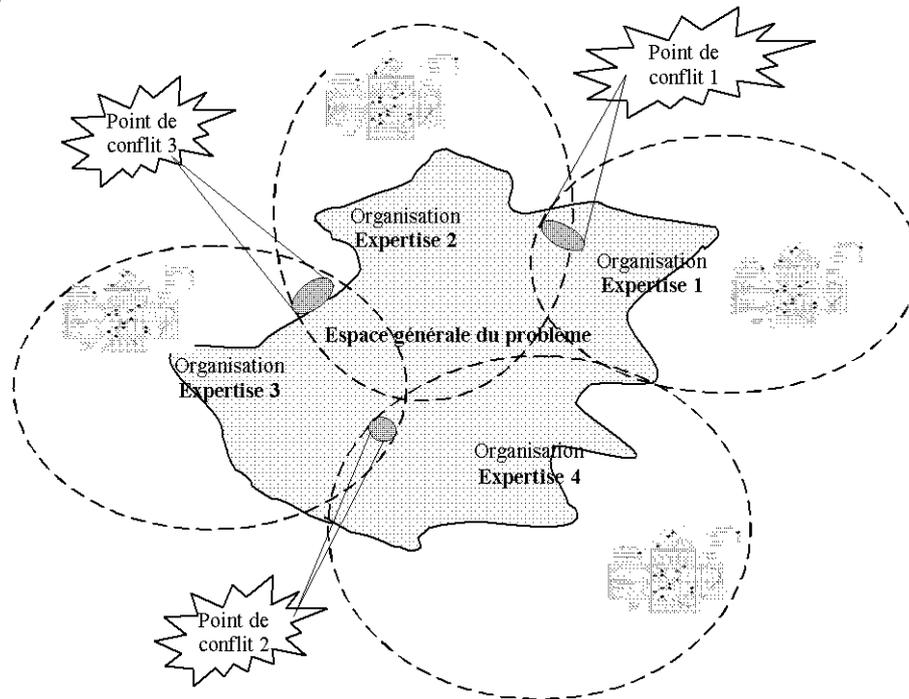


Fig 42. Points des conflits par rapport à l'expertise

Comme schématisé dans la figure 42, les points communs entre expertises des organisations créent des points de conflit où se déclarent des valeurs de poids affectés par plusieurs origines. Il se trouve que dans notre cas (SIC), le choix (poids, origine) peut attribuer des valeurs de poids différentes selon les points de vue de l'origine pour le même couple (exigences, spécification). Ce qui nécessite une négociation entre les différents participants (origines).

- Ensuite, *convaincre* toutes les parties prenantes afin qu'elles comprennent les exigences essentielles de leur point de vue.
- Enfin, arriver à une *entente* sur un ensemble d'exigences cohérentes qui satisfont le plus possible les parties prenantes.

Notre contribution va utiliser le GROUPWARE afin de permettre le travail de groupe

### 3.2 Négociation des conflits

La phase de négociation peut apparaître comme une activité séparée dans quelques approches d'ingénierie des exigences (voir chapitre I).

Notre démarche fait participer la négociation comme une tâche inévitable et essentiel dans le processus de validation des besoins pour trouver un compromis entre les différentes parties prenantes. Alors et en plus notre plate forme est coopératif et l'une de ces caractéristiques est

l'hétérogénéité conséquente des différences des composants SI, engendre plusieurs types de conflits qui fait de la prise en charge de la négociation un principe. Ce qui n'est pas bien prit en charge par la méthode *CREWS*. Chaque information extraite des SI locaux, génère surement des conflits sémantiques (conflits de domaine de définition, conflit dans l'interprétation, etc.), Lors de la coopération des SIs une interface d'accès homogène doit toucher la plate forme technique, systèmes logiciels, etc.

Ici on cherche à parvenir à un consensus afin de négocier les poids donnés sur les spécifications des exigences. L'obtention de consensus est mesuré par le taux de **complémentarité** entre exigences a prés fusion des différentes points de vus. Si ce taux est positif l'entente est Forte sinon elle est moyenne ou négatif.

Le choix d'une corrélation qualitative est une tâche, dont l'objectif est d'identifier le lien entre la plus appropriée \ ce qui "sera mis en œuvre et / comment" la mise en œuvre correspond à l'attente des intervenants. Comme il existe de nombreuses parties prenantes, il est naturel que les différentes valeurs puissent être proposées, selon des perspectives différentes sur le système, les interprétations de ce qui est impliqué dans le développement du système, les ordres du jour cachés, etc.

Trois alternatives pour l'obtention de corrélations dans le QFD ont été documentées dans la littérature: (1) demandent des réponses individuelles et de la moyenne des résultats, éventuellement en utilisant un facteur de modération, tels que l'importance relative attribuée à chacune des parties prenantes Stylianou et all [138]; (2) en utilisant une analyse multicritères, de préférence dans les préférences de combiner une fonction d'utilité Olson et all [139], et (3) à une réunion, où les parties prenantes doivent négocier leurs différents points de vue jusqu'à ce qu'un consensus soit atteint sibertin [141].

Bien qu'il existe des différences entre les deux premières approches, de leur accent est mis sur les individus, Cette dernière approche est considérée comme bénéfique pour la constitution d'équipes, l'augmentation de la participation dans le développement de produits, l'obtention d'un consensus général sur le "quoi faire", et la préservation de l'élan quand le groupe change. Un problème avec la dernière approche est que, en se fondant sur les réunions dé la définition des besoins à la négociation et d'obtention de consensus, le processus de développement prendra plus de temps. Cependant, ce retard sera bien récupéré dans les prochaines phases du processus de développement.

### **3.3 Unification des vues : et génération d'une unique QFD négocié**

La matrice QFD a été mise en œuvre avec une feuille de calcul Excel.

Cette matrice QFD fournit une perspective globale de la multiplicité en cours de négociation, chacune liée à une cellule QFD, pour que les acteurs bien parvenir à un accord.

Notre contribution pour l'unification des vues passe par deux étapes :

1 : Bien que les parties prenantes puissent réagir réciproquement avec la feuille de calcul QFD a travers le groupwere, pour analyser les négociations en cours, ils ne peuvent pas fixer ou

modifier les valeurs des cellules directement dans le tableau. Ici viens le rôle de la réunion comme solution à parvenir à un consensus.

Ce qui donne la séquence d'intervention suivant :

Dans le temps t1 l'organisation S1 affecte la valeur 1(faible) à la case C comme poids de son importance pour elle, Dans l'autre coté avec l'organisation 2 ils analysent le couple (exigence, spécification) et affecte la valeur 3 (moyen) sans savoir la valeur donnée par les autres organisations. Et de même pour l'organisation3 voir *Tableau V*

Temps	S1	S2	S3	Action
t1	1			S1 selects C = 1
t2				S2 analyses cellule
t3		3		S2 selects C = 3
t4				S3 analyses cellule
t5			3	S3 selects C = 3

TABLE XIV. EXEMPLE DE CONFLIT DANS LA QFD

2 : Dans le but de les convaincre sans pour autant favoriser l'un par rapport à l'autre.

Notre contribution consiste à **calculer la médiane** des valeurs de C Tableau VI. Comme amélioration au calcul de moyenne dans la méthode winwin Bohem [150], [142]. Comme la moyenne fait apparaître d'autres valeurs qui n'appartiennent pas a l'ensemble (0, 1,3, 9) qui n'est pas supporté par la QFD.

Temps	S1	S2	S3	Action
t6				Notre proposition C = 3

TABLE XV. EXEMPLE DE GESTION DE CONFLIT

3 : Ici il ne reste qu'a fusionnée les différentes QFD et vérifier **La complémentarité** ainsi que **la cohérence globale** entres ces derniers afin de ne trouver qu'une seul.

- **Tester la Complémentarité**

S'il est elle est **positif et importante** il se justifie soit par : (1) le bon choix des organisations participant à ce travail coopératives et (2) les rôles affecté a chacun des participants son conforme a leurs compétences. Ce ci augmente la relation entres ces organisation et crée un niveau de confiances.

### Exemple

L'organisation 1 : spécifie l'exigence **mode de Paiement** seulement par **{Carte bancaire}**  
 L'organisation 2 : le spécifie par **{Sur internet, Sur place, En espèce}**.  
**Lors de la fusion** l'exigence **mode de Paiement** englobera les techniques **{Carte bancaire, Sur internet, Sur place, En espèce}**.  
 Donc c'est spécification sont **complémentaire**

- **Tester la cohérence globale**

Ensuite nous passons à la métrique de *la cohérence globale* du système. Ici il apparaît d'autres problèmes avec la qualité du tout unie.

### Exemple

L'organisation 1 affecte le poids **9** au couple (mode de paiement, carte bancaire) ;  
L'organisation 2 affecte le poids **3** au couple (mode de paiement, carte bancaire) ;  
L'organisation n affecte le poids **3** au couple (mode de paiement, carte bancaire) ;  
La gestion de conflit par médiation donne la valeur **3**. **Donc la Cohérence globale = 1/1=1**. Car un conflit trouvé un réglé ce qui n'est pas le cas toujours.

### 3.4 Génération de prototype

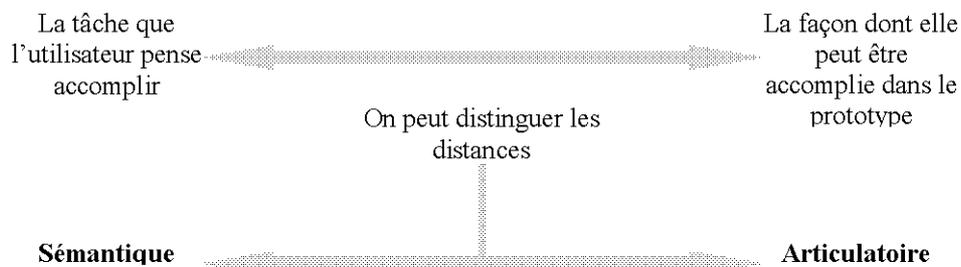
La validation est faite par le client ou ses représentants. Elle est faite suite à la vérification. Elle vise à assurer l'adéquation entre ce que l'analyste a écrit et ce que le client a dans la tête. Pour cela, les modèles élaborés doivent être communiqués aux utilisateurs afin d'être acceptés, modifiés ou refusés sous forme de prototype.

Ici le groupe d'IE va passer à l'automatisation d'un prototype rapide exécutable pour simuler le futur système, et ce ci en transformant le QFD (ensemble de spécifications vérifié et négocié) a un ensemble d'interfaces logicielles exécutables en line.

Les questions qui se posent ici sont : c'est comment être sûre de ce qu'ont attend du système réellement (problématique des sciences cognitives).

La seule façon de réduire l'écart entre système perçu (notre prototype) et système réel (ce que le client a dans la tête) est de confronter le modèle mental de maitre d'œuvre à celle du maitre d'ouvrage.

Cette difficulté pourrait être qualifié d' « articulatoires », en référence à la théorie de l'action de Norman [137]. Celle ci associe la réalisation d'une tâche au parcours d'une distance – figure 42. Les distances articulatoires traduisent les difficultés à adapter l'intention de l'utilisateur aux commandes disponibles, et à interpréter l'état du système à partir de l'état de l'interface.



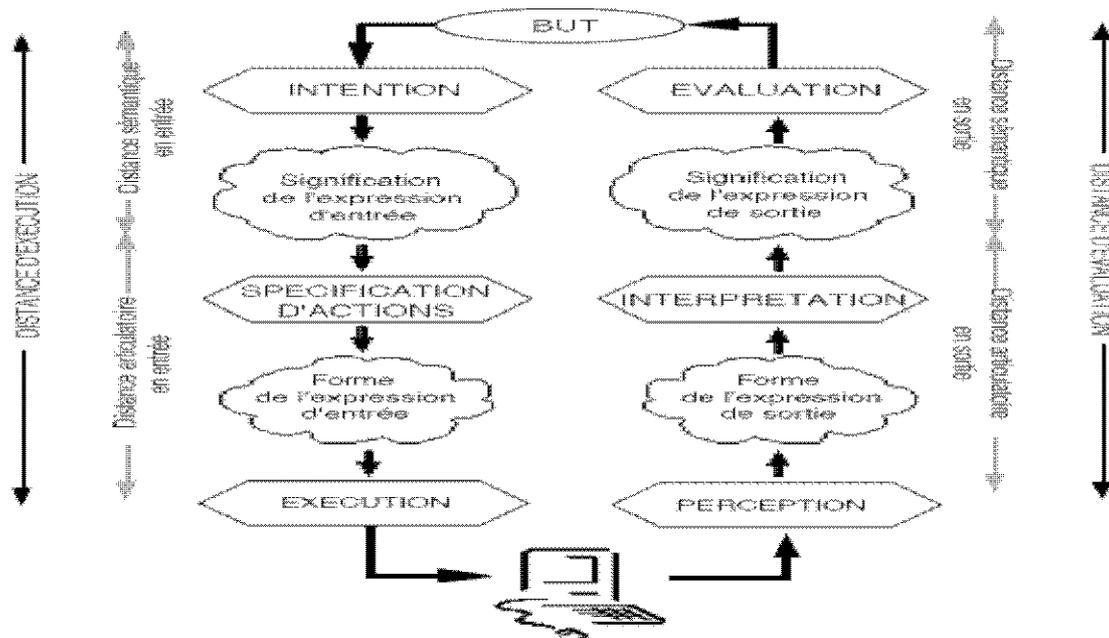


Fig 43. Distances sémantiques et distances articutoires [137].

La théorie de NORMAN [137], Décompose la réalisation d'une tâche :

- ✓ *établissement d'un but* : représentation mentale de l'état à atteindre.
- ✓ *formation d'une intention* : analyse de la différence entre état actuel et But, et décision d'agir pour atteindre le but.
- ✓ *spécification d'une suite d'actions physiques* (liens entre variables psychologiques (but) et variables physiques (système physique réel) et, correspondances entre variables physiques et dispositifs de commandes
- ✓ *exécution des actions*
- ✓ *perception de l'état du système* (représentation mentale)
- ✓ *interprétation* de l'état du système
- ✓ *évaluation* (comparaison au but recherché et ... retour)

**Motivation** : Aucune méthode, approche ou même théorie vise à guider les étapes de construction de prototype proposé dans la littérature, malgré le nombre énorme de travaux qui utilisent des prototypes pour assurer la validation des exigences.

L'effort de validation sera d'autant plus faible quand les distances sémantique et articutoire seront réduites.

La distance sémantique est fiable si l'on peut facilement :

- ✓ exprimer dans le langage de spécification ce que l'on veut réaliser,
- ✓ sans faire appel à des notions étrangères et
- ✓ comprendre ensuite le résultat.

La distance articulatoire est faible si l'on peut déduire facilement de la forme d'une expression sa signification. La validation passe par les étapes suivantes:

- ✓ Exprime les objectifs et les intentions en des termes significatifs pour l'utilisateur, en des termes liés à la représentation mentale qu'elle se fait de son problème, c'est-à-dire des termes psychologiques.
- ✓ Traduit ses intentions dans des actions à exécuter à l'aide des mécanismes de prototypage qu'on l'exécute sur machine.
- ✓ Perçoit physiquement le résultat des actions
- ✓ interprète mentalement la perception physique
- ✓ Évalue la réalisation de ses objectifs à partir des variables d'état du prototype qui sont des variables physiques.
- ✓ Discuter les résultats et négocier les spécifications avec les utilisateurs du système coopératif.

## VI. TRAVAUX VOISIN ET POSITIONNEMENT PAR RAPPORT A QUELQUES TRAVAUX D'ACTUALITE

La littérature mentionne deux niveaux de travaux: un **externe** qui représente la validation au sens propre du mot, avec *l'utilisateur, le client et / ou d'autres parties prenantes*. Basés sur des techniques telles que *les interactions, le prototypage, des animations ou des simulations*. D'autre part **interne** qui représente généralement la **vérification**, cette activité est assurée par *l'équipe d'IE* qui est le plus souvent basée sur des *techniques non conviviales* telles que : *Formulation mathématique, Logique, Formel, traitement de langage naturelle, etc.* utilisant des outils complexes ; Ce niveau a toujours besoin d'être complété par une seconde itération de transformation afin qu'il puisse être validé par le maître d'ouvrage, sinon sa crédibilité reste douteuse

Dans les travaux l'actualité ils y a mixages, ils travaillent à plusieurs niveaux interne et externe avec ces différents modes (formel, semi-formel, informel), utilisation des méthodes manuel et/ou automatisé *tableau 8*.

## VII. TRAVAUX VOISIN ET POSITIONNEMENT PAR RAPPORT A QUELQUES TRAVAUX D'ACTUALITE

Travaux	Démarche	Niveau de Validation	Technique de Validation	Outils utilisé	Animateur de VV	Principe
'Visual Requirements Validation: Case Study in a Corba-Supported Environment' 2002 [143]	/	Externe	Visuel Graphique	CORBAT	Multiutilisateur	
'Validating functional system requirements with scenarios' 2002 [144]	/	Externe	Scenario	Réseaux Bayesian	opérateurs humains	
'Supporting Requirements Verification Using	/	Externe	/	XSLT,XML REM	/	Produire des cahiers de charge automatiquement,

XSLT' 2002 [145]						fournir une vérification orientée métrique
'Applying use cases for the requirements validation of component-based real-time software' 1999[146]	/	Externe & Interne	Cas d'utilisation		/	Un modèle pour aides à la structuration des cas d'utilisation décrivent le comportement des logiciels basé composants.
<b>José Reinaldo</b> 'Applying petri nets to requirements validation'2004[147]	processus unifié	Externe & Interne	Cas d'utilisation, réseau de pétri	Outil Case	Utilisateur /analyste	Transformation de cas d'utilisation réalisé par les outils case en réseau de pétri
'Instrumentation automatisée des contrats et des scénarios de validation des exigences en. Net '2008[148]	/	Externe & Interne	scenario	.Net	Multiutilisateur	la validation a besoin d'un modèle testable capable de générer automatiquement des instruments de contrôle en vue de la réutilisation de ce modèle
'A Navigational Web Requirements Validation Through Animation'2008[149]	/	Externe & Interne	Animation Et Simulation	Web	les participants	valident les séquences d'animation produite traduites en un graphique de navigation
'A rigorous approach to requirements validation'2006[140]	/	Externe & Interne	prototypage rapide, scénario	Model-checking senarios	Utilisateur Analyste	Model-checking est utilisé dans l'analyse formelle identifier les incohérences internes dans le cahier des charges, et de générer des «Scénarios» des spécifications qui peuvent aider à identifier des incomplétudes ou des anormales.
'Using Inspections to Teach Requirements Validation'2008[79]	Empirique	Interne	Etude statistiques	PBR, checklist N_fold	Des Groupes d'Etudiants	Expérience, observation & analyse discussions, et résultats
<b>Méthode VECOD</b>	<b>Progressive Multipoint de vue</b>	<b>Interne/Externe</b>	<b>prototypage</b>	<b>QFD , check-list, groupware</b>	<b>Groupe distribué hétérogène</b>	Processus continu, collaborative, de vérification négociation et validation distribué,

TABLE XVI. COMPARAISON ET POSITIONNEMENT DE VECOD

---

## **Conclusion**

---

*L'exercice de la validation est long, coûteux et apporte parfois de mauvaises nouvelles : anomalies, non conformités et au final retard du projet, voire son annulation pure et simple.*

*Nous avons voulu apporter notre contribution au processus de validation des exigences dans un contexte distribué en proposant une Méthodologie de validation des exigences collaboratives dans les organisations distribuées baptisé VECOD est qui se résume à:*

- Surmonter le problème de la validation dans SI en précisant qui fait quoi, quand, où, comment et par quel moyen.*
  - Dominer le problème de l'ambiguïté du terme en choisissant une seule définition et en différenciant la vérification de la validation.*
  - Proposer pour le problème d'IE une démarche (vérification, négociation, validation), modélisé par des digrammes fonctionnels et un méta modèle d'UML.*
  - Enfin pour finir avec la plate forme coopérative, l'utilisation de groupware qui a permis de traiter les organisations distribuées.*
- 

---

## **Perspective de la méthodologie**

---

- Nous suggérons d'appliquer la méthode WINWIN au niveau de la phase de négociation.*
  - Utiliser un modèle incrémental, de réalisation de prototype testable à chaque niveau d'incrément.*
  - La prise en charge de la VV des exigences non fonctionnelles.*
-

---

*CHAPITRE 5 :*

*Déploiement Et Etude de Cas*

# Chapitre V :

# Déploiement & Etude de Cas

---

## *Résumé*

---

*La méthodologie VECOD sera déployée pas à pas à travers l'outil groupeware NetMeeting de Microsoft, permettant le travail collaboratif (d'échanger, de communiquer des documents et de coordonner un processus très réglé) et son tableau blanc interactif virtuel partagé permettant la construction du prototype à travers un travail collaboratif en temps réel.*

*L'étude de cas relatée dans ce travail porte sur la réforme organisationnelle du processus de la validation d'un cahier de charge pour une application de commerce électronique. C'est un projet qui fait appel à la coopération de plusieurs organisations spécialisées, chacune dans un domaine bien spécifié tel que la gestion, le commerce, les applications en line, etc.*

---

## I. INTRODUCTION

Pour que notre méthodologie soit concrétisée, nous essayerons de présenter la plate forme logiciel la plus appropriée, afin d'assurer la coopération. Notre choix a tombé sur Le logiciel NetMeeting<sup>10</sup> est une application de Microsoft qui permet la communication en temps réel sur un réseau local ou sur Internet, en utilisant le protocole TCP/IP. On peut ainsi faire de la visioconférence, échanger des fichiers ou partager des applications entre différents postes. Un tableau blanc Interactive Virtuel une fonction de ce groupware permet aux participants à une « conférence » de travailler en même temps dans le même environnement. Tous peuvent ainsi écrire et dessiner simultanément sur le tableau et voir immédiatement le résultat sur leur propre écran. On peut également dialoguer en mode texte (faire du chat) ou en mode vocal.

Dans le chapitre précédent de ce mémoire, nous avons proposé une méthodologie pour la validation des exigences dans les organisations distribuées, Afin d'illustrer les idées et les concepts inclus dans la contribution proposé, nous allons utiliser cette méthodologie comme base d'une étude de cas dans un environnement réel. Cette étude va nous permettre de prétendre le fonctionnement des différentes phases proposées appliqué sur quelques propriétés de qualité standard ainsi que celle émergentes de la plate forme coopératif. Le but de cette étude de cas est de montrer la faisabilité de la méthodologie basée point de vue comme enrichissement et nécessité dans des situations réels.

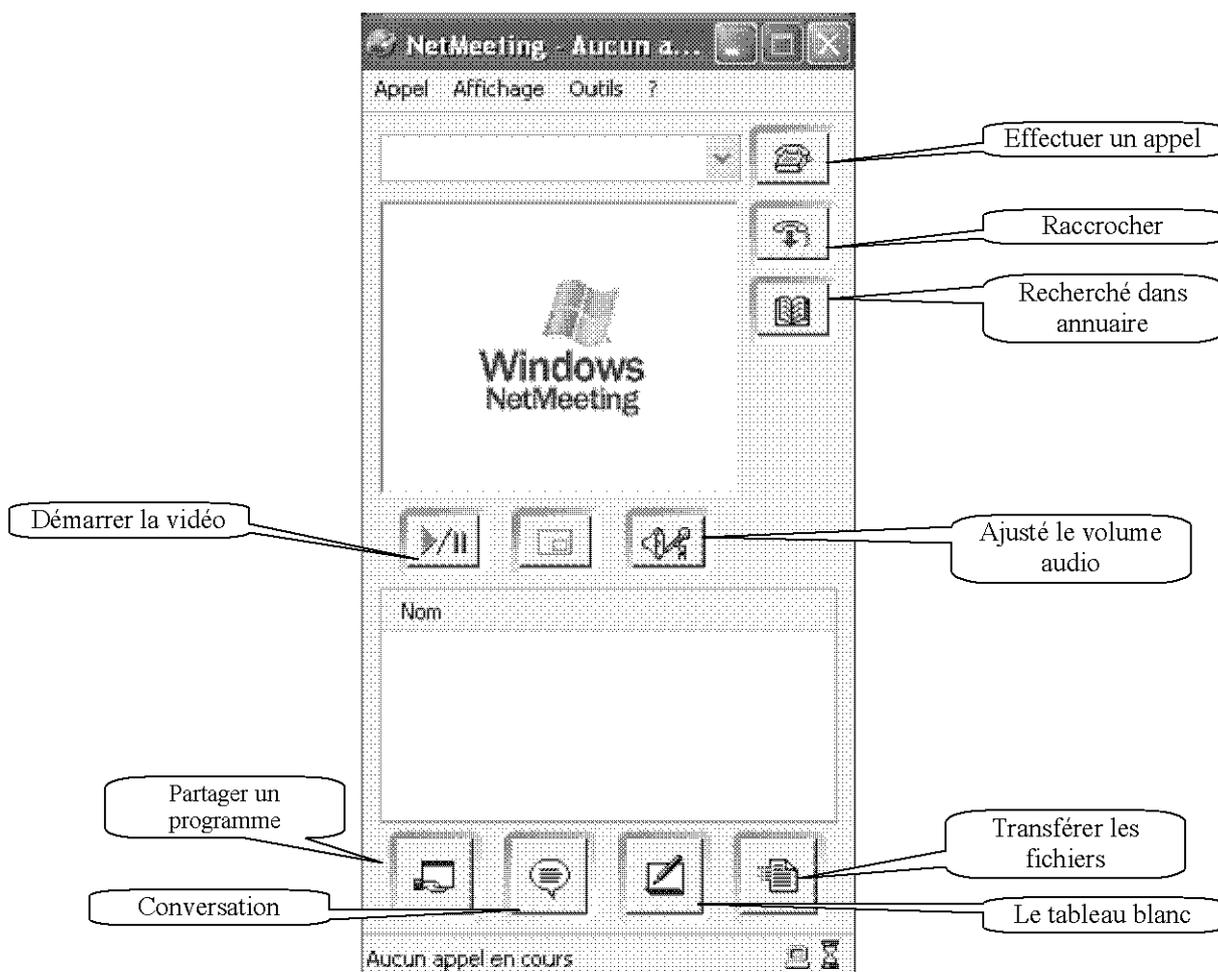
Afin de montrer la faisabilité de notre méthodologie VECOD, nous l'avons illustrée sur la gestion de la facturation dans un domaine de commerce électronique. Un exemple que l'on veut typique, car il n'y a pas assez d'exemples de 'bonne exigences', disponible (à apprendre d'eux). En partie parce que peu de projets ont de bons à partager, et de l'autre part que ceux d'entreprises sont prêts à mettre leurs spécifications des produits dans le domaine public, « Un produit mal conçu est une idée livrée gratuitement à la concurrence, accompagnée de son test en vraie grandeur sur le marché ».

## II. DEPLOIEMENT DES DIFFERENTES PHASES DE VECOD

Les fonctions de NetMeeting permettent notamment de collaborer en temps réel lors de l'utilisation de toute application Windows, de transférer des fichiers et d'utiliser un tableau blanc de dessin commun *figure 44*.

---

<sup>10</sup> Logiciel de visioconférence et de chat de Microsoft.



**Fig 44.** Interface de NetMeeting

Pour effectuer une communication à l'aide de "NetMeeting", les personnes utilisant des ordinateurs d'un réseau local doivent exécuter cet utilitaire.

Pour afficher la fenêtre principale de NetMeeting:

Cliquer sur le bouton "Démarrer";

Cliquer sur "Exécuter";

Saisir la commande "conf";

Valider par "OK".

#### **SUR INTERNET**

Une adresse IP permet d'identifier une machine sur un réseau TCP/IP de manière unique.

L'adressage peut être statique ou dynamique. Dans le premier cas, l'adresse IP est fixe ; dans le second cas, une nouvelle adresse est attribuée à chaque allumage de l'ordinateur.

Pour connaître l'adresse IP d'un poste, il suffit de cliquer sur le « ? » de NetMeeting, puis de choisir la rubrique A propos de Windows NetMeeting. Apparaît alors la phrase « Adresse IP » suivie d'une série de 4 groupes de 1 à 3 chiffres séparés par un point. Par exemple : 192.168.0.100

On peut également exécuter la commande winipcfg (par le bouton Démarrer – Exécuter) qui donne tous les paramètres de la configuration IP de la machine.

## 1. Phase d'organisation

L'objectif essentiel de cette phase est de :

Préparer l'environnement de travail, Procéder à la collecte des documents, Préparer les artefacts appropriés pour le suivi du processus et diffusé les documents.

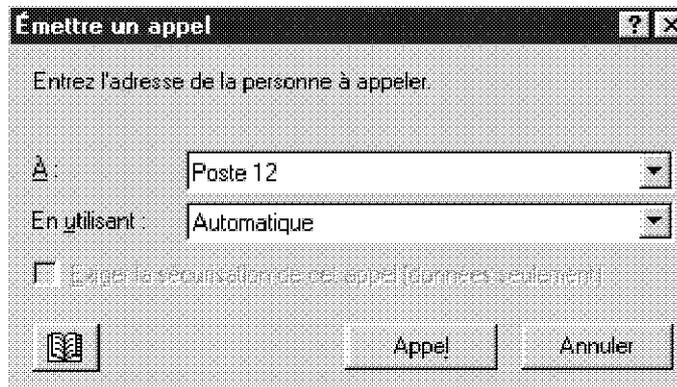
Pour cela il établit une communication. Un des utilisateurs doit envoyer un appel et son correspondant doit l'accepter.

### 1.1 La Planification

#### a- Envoi d'un appel

Saisir le nom de l'ordinateur du correspondant dans la zone A :

Lancer l'appel en cliquant sur le bouton "Effectuer un appel" *Figure 45*.



**Fig 45.** Interface Emettre un appel

#### b- Réception d'un appel

La personne utilisant le poste appelé est avertie par une sonnerie. Pour commencer la communication il doit cliquer sur le bouton "Accepter", et pour refuser il doit cliquer sur "Ignorer".

Ce processus permet de fixer les rendez-vous des réunions à travers l'envoi de d'appel pour tous les membres de l'équipe d'ingénierie (membres de l'organisation virtuelle).

### 1.2 La Qualification

Après la préparation des artefacts *figure 46*, présentés par la fiche technique de l'exigence, la checklist et la grille d'évaluation. Qui seront utilisés réciproquement par tous les postes. Et cela grâce au "partager un programme" de NetMeeting.



**Fig 46.** Fiche technique de l'exigence, check-list, grille d'évaluation

### 1.3 Inspection

L'équipe d'ingénierie des exigences s'appelle pour inspecter les documents. La réunion se termine alors par la formulation d'une *Check-list* qui guidera le reste du processus.

La mise à jour fait sur les documents est enregistré et plusieurs types de fichier peuvent être utilisés (pdf, doc, xls, bmp, etc.), à travers l'option transfert de fichier de NetMeeting *figure 47*.



**Fig 47.** La diffusion des documents

## 2. Phase de vérification des exigences

Au niveau de chaque site et hors réunion, peut commencer une inspection spécialisée (selon la partie des exigences qui les concerne). Le but est d'assurer l'aptitude des exigences à leur spécification.

Après les vérifications des documents d'exigences, précédemment diffusé par l'analyse chaque exigences est vérifier a part pour ces critères : fonctionnelles d'aspect sémantique, fonctionnelles d'aspect non sémantique et non fonctionnelles.

```

Procédure vérification (document : chaîne de caractère) ;

Début
DOC :=vide ; Liste_Erreur :=vide
  Répéter pour le document
    Répéter pour chaque exigence
      Vérification pour chaque exigence les critères non sémantiques
      Vérification pour chaque exigence les critères sémantiques
      Vérification pour chaque exigence les critères non fonctionnels
      Correction des fautes
      si désaccord alors Liste_Erreur := Liste_Erreur + exigences ; (* crée liste des erreurs*)
      jusqu'à EOF : (* fin document *)
    DOC :=DOC + exigences ; (* Intégration l'exigence dans le document *)
  Jusqu'à traiter tous le document.

Fin.

```

Après la vérification qui s'achève par le filtrage des documents, une analyse statistique est prête a commencé. Cette analyse commence par la génération de la matrice QFD.

```

Procédure affecté la QFD
  Début
    Four i :=1 a tous les exigences faire
      Four j :=1 a toutes les spécifications faire
        Mat [exigence, spécification] := Poids ;
      Fin pour ;
    Fin pour ;
  Fin

```

## 2.1 Processus de génération de la QFD

Un modèle précédemment généré de classeur EXCEL *figure 48*, générons un QFD est établie et enregistré. Ce modèle peut être utilisé par les membres de la réunion à part et en groupe. Ce ci est parelle a l'utilisation pédagogique de NetMeeting dans une salle informatique en réseau, une séquence d'apprentissage du tableur dans laquelle le professeur montre aux élèves, à partir de son poste, comment entrer une formule et la recopier. Ceux-ci interviennent pour proposer leurs solutions.

*L'analyste ici va jouer le rôle de l'enseignant et le reste de l'équipe les étudiants.* Ce partage ce fait à travers l'option partage des applications *figure 49*.

Choisissez *Outils* – 'Partage'

Dans la liste des applications, sélectionnez Microsoft Excel.

Cliquez sur le bouton 'Partager'.

Pour que les participants puissent agir sur votre programme, cliquez sur le bouton 'Permettre le contrôle'

Pour éviter que les élèves ne prennent le contrôle sans votre autorisation, ne cochez pas l'option 'Accepter automatiquement les demandes de contrôle'

Cliquez sur le bouton 'Fermer'.

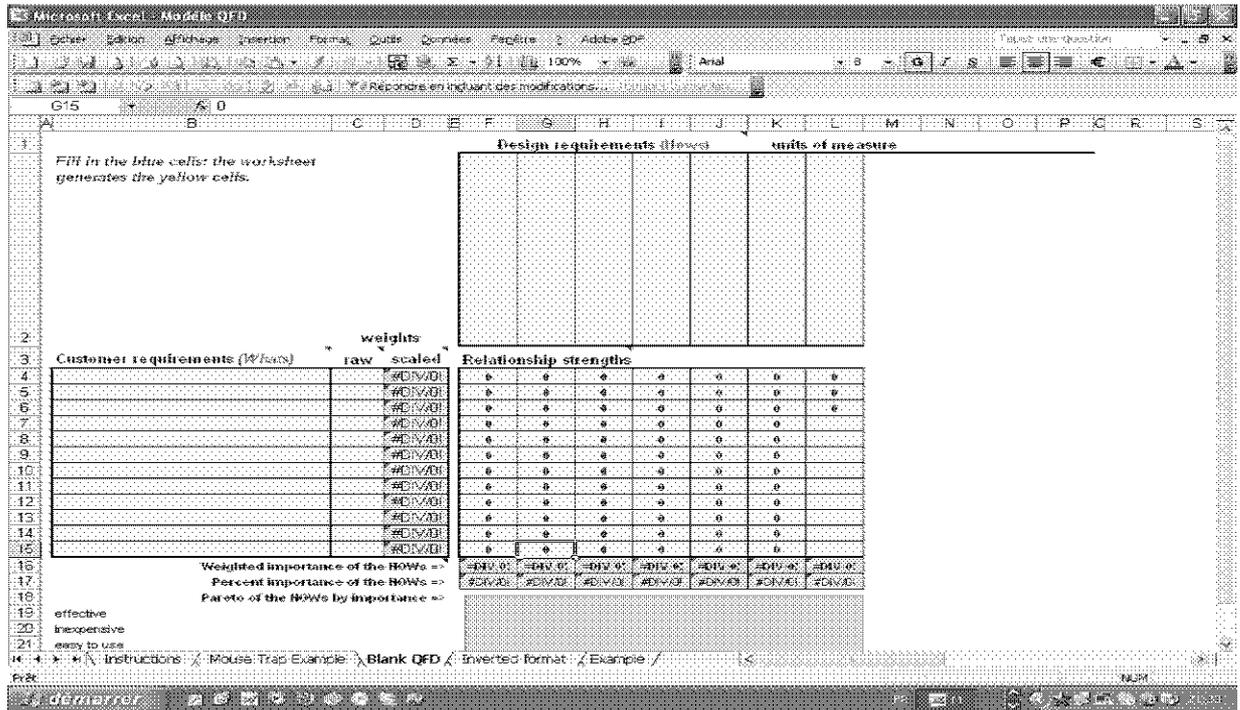


Fig 48. Modèle de QFD sous EXCEL

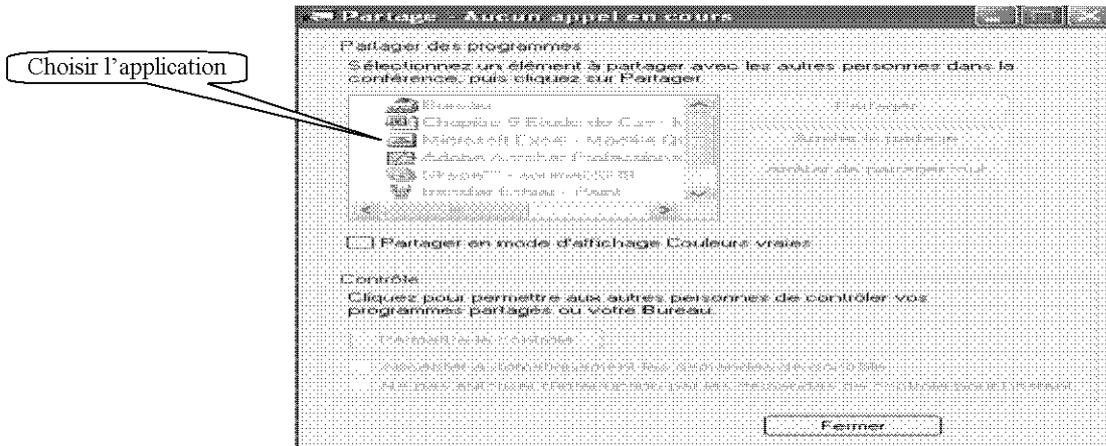


Fig 49. Partagé des Application

### 3. Processus de validation des besoins

A travers le partage de programme de NetMeeting un travail en groupe est effectué, supervisé par l'analyste afin de fusionnés les QFD et les négociés.

L'analyste a le contrôle de l'application et par conséquent, voit toutes les propositions des partie prenantes, ils peuvent l discuté si nécessaire ?

### 3.1. Détection de conflits

*Fonction Détection conflit (QFD1, QFD2) : Logique ;*

Début

*Id := 1 ;*

*repete*

**Si** (pour le même couple (id, spécification) de QFD1 et QFD2  $\exists$  plusieurs poids)

**Alors** *return (vrai) ;*

*id := id + 1 ;*

*Jusqu'à id > taille QFD1*

Fin

### 3.2. Gestion de conflits

*Fonction gestion conflit (QFD1, QFD2, id : position du conflit) : poids ;*

Début

*Trié les différentes poids de id. QFD1 et id. QFD2 ;*

*Choisir le médiane ; (\* Notre contribution \*)*

*Return(Médiane) ;*

Fin

### 3.3. Fusion des QFD

*Fonction fusion de conflit (QFD1, QFD2) : QFD ;*

Début

*Id := 1 ; complémentarité := 0 ;*

*repete*

**Si** (id de l'exigence de QFD 1 = id de l'exigence de QFD 2)

*Et (spécification 2 not in spécification QFD1)*

**Alors début** *spécification 1 := spécification 1 +  $\sum$  spécification 2 pour la même id ;*

*complémentarité := complémentarité + 1 ;*

**fin ;**

*id := id + 1 ;*

*Jusqu'à id > taille QFD1 ;*

*Taux de complémentarité = complémentarité / taille QFD ; (\* pour mesurer le taux de la complémentarité entre les différentes organisations participons \*)*

Fin.

#### a. Tester la complémentarité

*Fonction Calcule Complémentarité (QFD<sub>1</sub>, QFD<sub>2</sub>, ..., QFD<sub>n</sub>) : Complément ;*

Début

*Moyen :=  $NLQFD_1 + NLQFD_2 + \dots + NLQFD_n / N$  ;*

*Nombre :=  $NLQFD_1 + NLQFD_2 + \dots + NLQFD_n$  (\*Nombre de Lignes \*)*

*Complément = Nombre - NL QFD<sub>final</sub> ; (\*Nombre de Lignes après fusion\*)*

**Si** *complément = 0 alors ('aucun complémentarité')*

*Si* *complément > Moyen taille des QDF alors ('Bonne complémentarité')*

**Sinon** *('mauvaise complémentarité') ;*

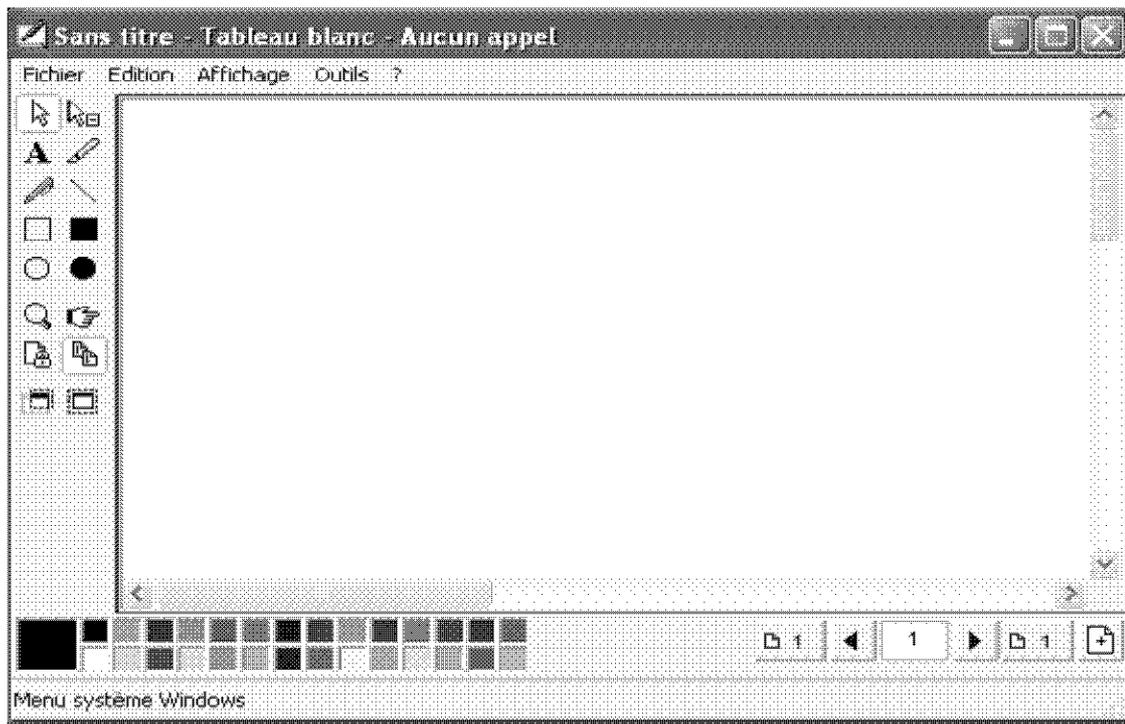
Fin.

## b. Tester la cohérence globale

*Fonction Cohérence Globale (QFD<sub>1</sub>, QFD<sub>2</sub>) : Complément :*  
*Début*  
*Cohérence Globale := Nombre de conflits réglé / Nombre de conflits négocié.*  
*Fin.*

### 3.4. Génération de prototype

Le Tableau blanc de NetMeeting de la *figure 50* permet de dessiner et d'écrire sur une surface commune visible par tous les participants à une conférence. Chacun peut, également, y coller des textes, des images... (via le Presse-papiers de Windows) en provenance de logiciels qu'il utilise sur son poste. Il les met ainsi à la disposition des autres conférenciers qui, à leur tour, peuvent les exploiter dans leurs propres applications. Chacun peut imprimer, sauvegarder le contenu du Tableau blanc, depuis son poste de travail. Ce logiciel offre des outils, un environnement et des fonctionnalités qui rappellent le logiciel de dessin Paint, accessoire de Windows, généralement bien connu des enfants.



**Fig 50.** nterface Tableau blanc de NetMeeting

L'analyste ouvre une conférence sur un poste et lance le Tableau blanc. Les participants, en appelant le poste maître y accèdent depuis leur ordinateur et peuvent y « déposer » le résultat de leurs travaux par simple copier/coller.

Ici commence la génération du prototype, qui sera créée à fur et à mesure du modèle réalisé par le groupe. Lorsque tous les groupes ont terminé leur spécimen, on procède à la correction collective en consultant l'ensemble des pages. On sauvegarde et on imprime le contenu du Tableau blanc. Afin de garder trace des résultats d'analyse qui sera utilisé plus tard dans la conception et l'implémentation.

### III. DESCRIPTION DU L'ETUDE DE CAS : COMMERCE ELECTRONIQUE

Pour voir comment peut on appliquer la méthodologie proposée, pour la validation des besoins utilisateur et vérifier les exigences systèmes, nous concéderons que l'exemple du cas d'étude est de regrouper des organisations voir entreprises travaillant ensemble dans le but espéré est un gain de compétitivité, en offrant la possibilité à des ingénieurs et techniciens qui doivent coopérer d'enrichir et de multiplier le nombre de leurs interactions.

ASPECT FONCTIONNEL NON SEMANTIQUE
Exemple : Besoins calculé le prix total La facture est une pièce justificatif pour l'acheteur, qui à une structure presque unique pour tous les domaines. Elle contient <i>les prix unitaires</i> et les quantités achetées de chaque produit.
ASPECT FONCTIONNEL SEMANTIQUE
Lors de la vérification des propriétés comportemental du besoin « Besoins calculé le prix total », on trouve que le calcul de la facture dépend de la TVA. Alors que la TVA peut être une valeur constante ou calculé ? on est obligé de prendre en considération cette ambiguïté, et on peut proposer le développement d'une fonction facturation avec la TVA variable et paramétrable selon le cas. Et même chose pour la langue de la facture.
ASPECT NON FONCTIONNEL
l'adaptation de la valeur du montant au règlement du payé (avec arrondissement, sans arrondissement, etc.) ; la forme du montant (ex. : 1000, 1000,00, 100.000 DA) de la facture a chaque magasin. De même l'utilisation des lettres alphabétiques pour décrire le montant de la facture est un aspect non fonctionnel, il ne dépend pas de la facture, mais il sert à vérifier et empêché les erreurs de calcul lors de la lecture des factures.

## 4. Déroulement de l'exécution

### 1.1. L'organisation 1 : Spécialiste en application Site Web et de Sécurité

On fait intervenir des spécialistes en création de site web et de sécurité afin de participer à la validation du cahier de charge selon la partie du problème qui les concerne voir *figure 51*.



### 1.3. Fusion des matrices et gestion des conflits

Après la fusion de deux QFDs et la gestion des conflits dans le point A de la figure 50 et dans le point B de la figure 52, le QFD générique sera tel que la figure 53 point C.

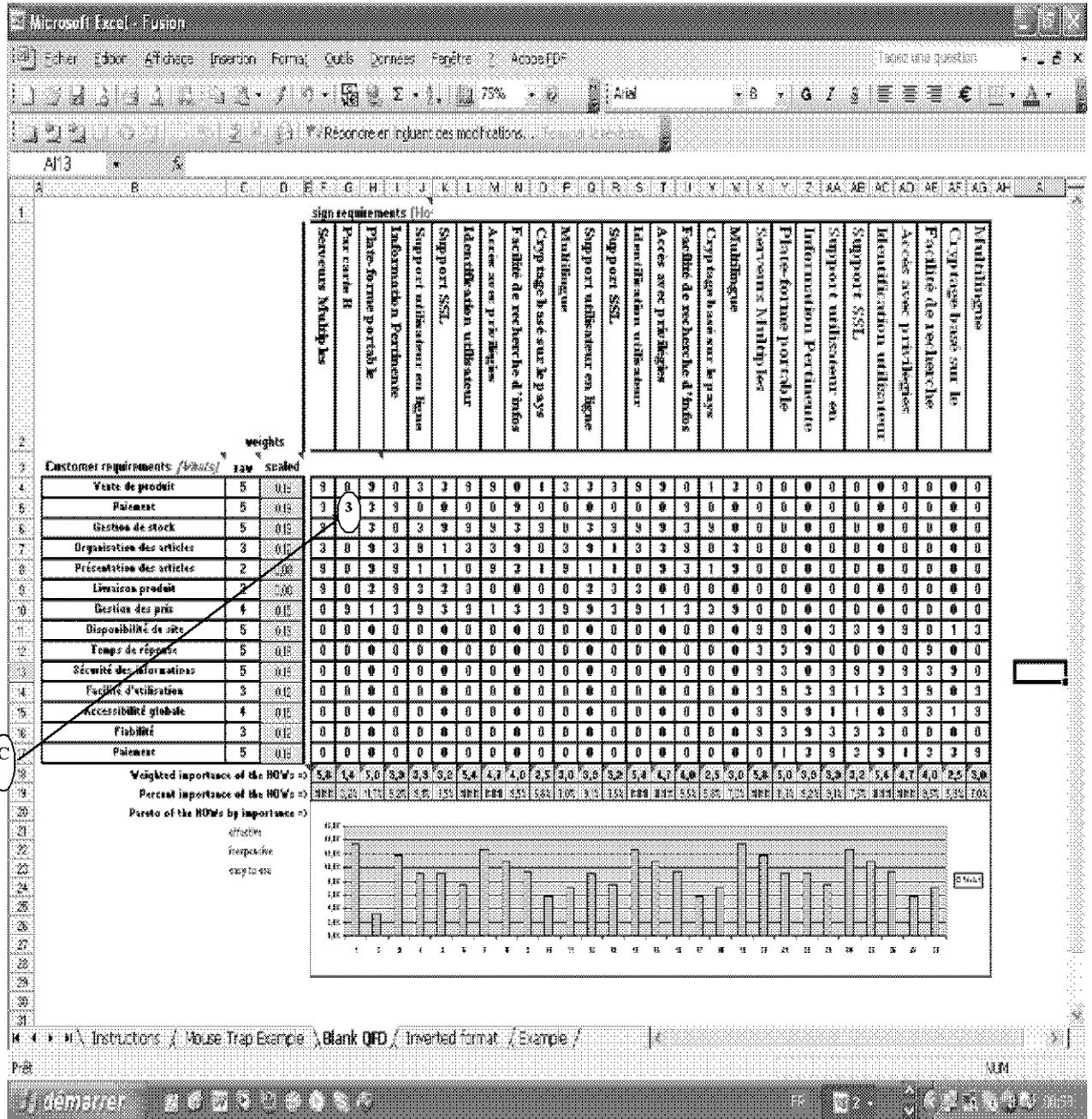


Fig 53. QFD générale du problème

L'histogramme généré sera pour autant un conseiller qui assistera par la suite à la décision de la validation. Les étapes qui restent du processus de génération de la QFD n'ont aucun intérêt ici, ils seront appelés plus tard dans le processus de gestion des exigences est exactement lors de la gestion du changement.

## 5. Génération de prototype

« Le tableau blanc interactif partagé (TBIP) », une fonction (application) qui fut longtemps utilisée à travers les enseignants dans les sales TP. Il s'agit en fait d'ouvrir un TB Virtuel sur un PC et le partagé avec des étudiants collaborateurs lors d'un travail à distance ou conférence.

La réalisation de la facture comme étude de cas *figure 54* fait intervenir de façon collaborative toutes les participions à la conférence en utilisant le tableau blanc qui garde trace de toutes les étapes de ça réalisation. Chaque participant peut copier, coller, sélectionner, etc., jusqu'à arrivé a une entant globale pareil a la construction d'une expression faisons intervenir toutes la classe. Par la suite ce prototype peut être enregistré, modifier, imprimé, etc.

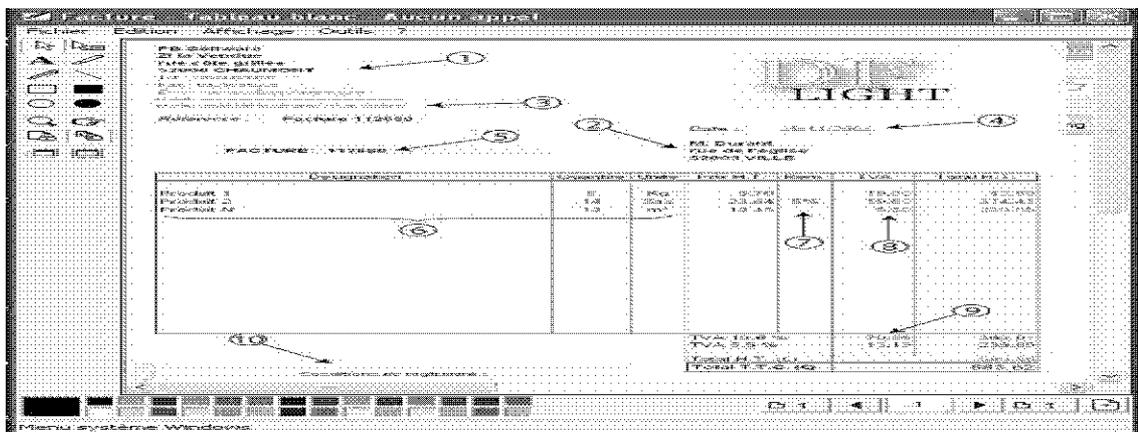


Fig 54. Exemple de création de facture en collaboration

## 6. Evaluation de la coopération

Afin d'évalué le taux de coopération entre les différentes organisations participions a notre étude de cas. On a calculé le **taux de la complémentarité** entre les QFD qui ont découle, et la figure 55 montre qu'il est assez positif et le choix des organisations été donc bénéfique.

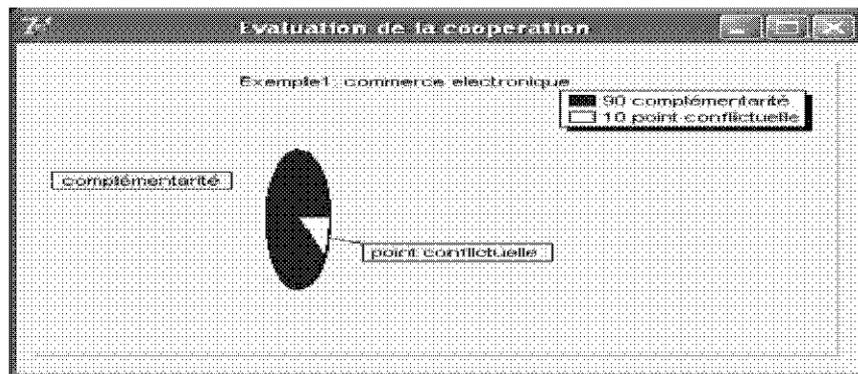


Fig 55. Taux de complémentarité, 'exemple commerce électronique'

---

## **Conclusion**

---

*Nous avons montré comment peu t-on utilisé un groupware comme l'infrastructure utilisée par les organisations membres du projet de coopération, en l'occurrence, nous avons choisi pour le déploiement de la méthodologie VECOD le groupware NetMeeting de Microsoft. Commençons de : l'organisation du travail collaborative, les taches individuelles, les réunions de travail jusqu'à la validation.*

*Suivie d'une étude de cas pour monter la faisabilité de la méthodologie avec un cas réel. Le cas présenté est la validation d'un cahier de charge pour la une application de commerce électronique. Sa prise en charge nécessite l'existence de plusieurs expertises différentes. Nous avant par le billé de l'exemple montré l'obligation d'appelé des expert en création de site web et d'autres en commerce. Ceux même qui ont participé lors de l'elicitation et la spécification des besoins, afin d'ont participé de même lors de la validation des livrables qui en résulte.*

*Pour finir nous avons évalué le taux de coopération par le taux de complémentarité entres les différentes points de vue.*

*Mais comme dit la règle statistique que l'ensemble doit être statistiquement suffisante pour que le résultat soit pertinent.*

*Comme l'étude de cas ne peut visualiser qu'une petite partie des cas qui peuvent survenir, le calcule de la propriété de cohérence global en ajoutera rien, comme on a mentionné un seul point de conflit et on là réglé.*

---

---

*CONCLUSION GÉNÉALE*

*&*

*PERSPECTIVES*

---

# Conclusion Générale

---

Dans ce travail, nous nous sommes focalisés sur l'étude de plusieurs domaines différents à savoir : L'ingénierie des exigences (IE), La Validation des Exigences (VE), Les systèmes d'informations coopératifs (SIC) et les groupwares.

Le but principal du présent travail consistait à contribuer à la validation dans le processus d'ingénierie des exigences, appliqué aux environnements coopératifs. La motivation qui sous-tend ce travail réside principalement dans le fait qu'une *Méthodologie de validation collaborative* offre une valeur ajoutée certaine *dans les organisations distribuées* et *l'approche multi points de vue* reflétant les compétences cognitives des parties prenantes est sans doute la mieux adapté.

Son processus est divisé en plusieurs activités, afin de supporter la plate forme coopérative tout en précisant les rôles et les résultats ainsi que les outils de chaque sous processus. Cette démarche méthodologique générale englobe l'ensemble des activités requises pour vérifier, négocier, valider et faire évoluer un système apportant une solution économique et performante aux besoins d'un client tout en satisfaisant l'ensemble des parties prenantes qui s'articule autour d'un processus continu, progressif, collaboratif, distribué, multi point de vue.

Dans notre travail nous avons concentré nos efforts sur quatre axes :

- *Dû au problème de l'ambiguïté du terme validation dans la littérature*, on est parvenu, à montrer que la validation des besoins utilisateurs est subjective et sans l'intervention du maître d'ouvrage ou ces représentants, sa crédibilité reste douteuse elle touche l'aspect sémantique!, Elle vient suite à la vérification des exigences systèmes par les maîtres d'œuvres présentés généralement par une équipe dont les compétences sont complémentaires et/ou conflictuelles. L'éminente gestion des conflits émergents est surmontée par leurs négociations, dont le but est d'unifier les différents points de vues. Une définition de la validation et de la vérification des exigences, prenant en compte la séparation en aspects et selon leurs aptitudes sémantiques, précisons en suite les critères de qualité à vérifier et à valider, enrichis par d'autres critères émergents du travail coopératif distribué.

- *Dû au problème de la validation en IE*, nous avons proposé une démarche plus générale que celle trouvée dans la littérature, et qui traite la validation en tant qu'activité et non en tant que phase, dont le processus est continu, incrémental, collaboratif, qui se déroule tout au long du processus IE.

- *Dû au problème de la validation dans le cycle de vie logiciel*, une réplique claire répondant aux questions : *Quoi Valider et Vérifier*, *Qui assure la tâche dans chaque étape de la démarche*, *Quand intervient la Validation et la Vérification par rapport au cycle de l'IE*, *Comment valider et vérifier*, etc. Sont tous modélisés par des diagrammes statique et dynamique de UML, et pour finir une modélisation en Meta-Modèle englobe notre vue pour la validation dans le processus d'IE.

- *Dû au problème de la plate forme coopérative*, le déploiement de notre méthodologie par un outil groupware (collecticiel), est présenté en détail. Notre choix est tombé sur NetMeeting de Microsoft et son outil Tableau Blanc Interactif Virtuel. On a évalué pour finir le taux de coopération à travers deux métriques qui mesurent la complémentarité et la cohérence globale dû au travail collaboratif.

On est encore loin du processus d'automatisation totale de la validation dans l'IE, car elle est toujours subjective à cause de la sémantique des exigences à l'égard des fameuses interprétations apparents chez les êtres humains. Toutefois, l'aide à la Validation et à la Vérification présente des avantages certains. Nous pensons que les applications logicielles dans la Validation quelque soit la plateforme se limiteront pendant quelques temps encore et resteront vraisemblable à des systèmes d'aide à la décision de complexité différentes.

# Perspectives

---

- Proposition d'une méthode de validation progressive des exigences vers des modèles dont la sémantique est connue, à partir d'exigences en langage naturel contrôlé.
- Etudier la vérification des besoins en se référant à des métriques et mesure bien définis.
- Etudier la négociation des besoins dans le processus d'ingénierie des besoins et l'appliquer en détail dans les environnements coopératif au tant qu'une activité séparée.
- La validation des exigences devrait englober à la fois des éléments quantitatifs et qualitatifs.
- Le Maintien de la validation au cours du processus de gestion des exigences qui nécessite un examen fréquent de tous les documents, qui garantissent la cohérence s'il y a eu de modification, d'écart, de défaillance ou de modification du procédé de production, y compris les procédures de contrôle des changements et versions.

---

# Références Bibliographiques

---

# Références Bibliographiques

---

- [1] Gerald Kotonya and Ian Sommerville, "Requirements Engineering Processes and Techniques", John Wiley & Sons, England, 1998
- [2] Julio Cesar Sampaio do Prado Leite and Peter A. Freeman, "Requirements Validation Through Viewpoint Resolution", IEEE transactions on Software Engineering, Vol. 17, No. 12, December 1991.
- [3] Ian Sommerville, "Integrated Requirements Engineering: A Tutorial", IEEE Computer Society, 2005.
- [4] AFIS, « Ingénierie des Exigences » <http://www.afis.fr>
- [5] D.Amyot uotawa 'Fondement de l'ingénierie des besoins 'Module 1 SEG 3501 .
- [6] C. Sibertin-Blanc, Université Toulouse 1 cour GL chapitre : La spécification
- [7] D. Essame' La méthode B et l'ingénierie système : Répondre à un appel d'offre'. 2007.
- [8] Bashar Nuseibeh, Steve Easterbrook "Requirements Engineering: A Roadmap" 35-46. Finkelstein, A. C. W. The Future of Software Engineering. New York, NY: ACM Press, 2000.
- [9] Klaus Pohl "The Three Dimensions of Requirements Engineering" 1993 - Springer
- [10] P. Loucopoulos & V. Karakostas 'System Requirements Engineering : Chapter 2 - Processes in R.E. !' page 4.
- [11] Pohl, K. 'Process-Centered Requirements Engineering' Research Studies Press Ltd., Taunton, Somerset, England, 1996.
- [12] Posner, M. I. (Ed.). 'Foundations of Cognitive Science. MIT Press'. (1993).
- [13] Goguen, J. & Jirotko, M. (Ed.)'Requirements Engineering: Social and Technical Issues'. London: Academic Press. (1994).
- [14] Goguen, J. & Linde, C. 'Techniques for Requirements Elicitation'. 1st IEEE International Symposium on Requirements Engineering (RE'93), San Diego, USA, 4-6th January 1993, pp. 152-164. (1993).
- [15] Lehman, M. M. Programs, Life Cycles, and Laws of Software Evolution. Proceedings of the IEEE, 68(9): 1060-1076. (1980).
- [16] Holtzblatt, K. & Beyer, H. R. Requirements Gathering: The Human Factor. Communications of the ACM, 38(5): 31-32. (1995).
- [17] Burg, J. F. MLinguistic Instruments in Requirements Engineering. Amsterdam: IOS Press. . (1997).
- [18] D.Amyot uOttawa' SEG 3601- Module 4 Analyse et validation d'exigences'
- [19] Larry Boldt, 'Trends in Requirements Engineering People-Process-Technology' Technology Builders, Inc., 2001.
- [20] Päivi Parviainen, Hanna Hulkko, Jukka, Kääriäinen, Juha Takalo & Maarit Tihinen' Requirements engineering Inventory of technologies' VTT PUBLICATIONS 508 ISSN 1455-0849 (URL: <http://www.vtt.fi/inf/pdf/>)
- [21] Vierimaa, M, Ronkainen, J, Salo, O, Sandelin, T, Tihinen, M, Freimut, B & Parviainen, 'Comprehensive collection and utilisation of software measurement data'. P2001. MIKKO Handbook Pages 96-100. Technical Research Centre of Finland. VTT Publications 445.
- [22] Beyer, H & Holtzblatt, K' Contextual Design: Defining Customer- Centered Systems' Morgan Kaufmann Publishers, Inc1998.
- [23] Sawyer, P.& Kotonya, G ' Software Requirements ' In the Trial Version (0.95) of SWEBOK, Guide to the Software Engineering Body of Knowledge. Chapter 2, pages 1-26. <http://www.swebok.org/> (available 26.3.2002) 2001.
- [24] Gerald Kotonya , Ian Sommerville' Requirements Engineering', John Wiley and Sons, 2004.
- [25] Davis A. M,," Software requirements: analysis and specification"1990 Prentice Hall.
- [26] Leffingwell D, Widrig D,' Managing Software Requirements – A Unified Approach',2000 Addison-wesley.
- [27] Sailor J. D,' System Engineering: An Introduction' In: Thayer, R.H. & Dorfman M. IEEE System and Software Requirements Engineering. IEEE Software Computer Society Press Tutorial. IEEE Software Society Press 1990.
- [28] Sommerville. I, Sawyer.P,'Requirements Engineering: A Good Practice Guide', 1997John Wiley & Sons.
- [29] <http://crinfo.univ-paris1.fr/CREWS/Process>.
- [30] C. Rolland, C. Ben Achour, 'Guiding the construction of textual use case specifications'. Data and Knowledge Engineering Journal, 1997.
- [31] Madu, Christian M. 'House of Quality in a Minute' . Fairfield, CT: Chi Publishers. (1999).
- [32] Akao, Yoji 'QFD: Past, Present and Future ' . [http://www.qfdi.org/QFD\\_History.pdf](http://www.qfdi.org/QFD_History.pdf) (1997).
- [33] Bocquet J.C, Duffau B, "La gestion du changement: de la gestion de production à la gestion de conception." L'art du management. 1997, pp.626-631, V Mondiale
- [34] Chuen-Lung C, Stanley F, Bullington "Development of a strategic plan for an academic department through the use of QFD" Computers& industrial engineering, 1993, pp 49-52
- [35] J. SAADI , A. ADIL, FAKHREDDINE' Application de l'approche QFD à la conception de programmes dans l'université marocaine CP 2005 – Casablanca, Morocco .
- [36] Estelle FREY, Samuel GOMES, Jean-Claude SAGOT 'Application de la méthode QFD comme outil d'extraction des connaissances métier en conception intégrée 'Copyright 2007 – Projet CoDeKF.
- [37] SWEBOK, "Guide to the Software Engineering Body of Knowledge", IEEE, 2004.
- [38] Wikipedia, "Entropy", October 04, 2008, 13:45, (<http://en.wikipedia.org/wiki/Entropy>)
- [39] Siew Hock Ow & Mashkuri Hj. Yaacob, "A Study on the Requirements Review Process in Software Development: Problems and Solutions", IEEE, 1997.
- [40] Ian Sommerville, "Software Engineering Seventh edition", Pearson Education Limited, USA, 2004.
- [41] Karl Wiegers, "Software Requirements, 2nd Edition", Microsoft Press, 2003; ISBN 0-7356-1879-8
- [42] Michael Fagan, "Design and Code Inspections to Reduce Errors in Program Development", IBM Systems Journal 15, 3 (1976): 182-211.
- [43] O. Laitenberger, T. Beil, T. Schwinn, "An industrial case study to examine a non-traditional inspection implementation for requirements specifications", presented at Proceedings of the Eighth IEEE Symposium on Software Metrics, Los Alamitos CA, 2002.

- [44] M. Halling, P. Grünbacher, S. Biffl, "Groupware Support for Software Requirements Inspection", ([www.cas.mcmaster.ca/wise/wise01/HallingGrunbacherBiffl.pdf](http://www.cas.mcmaster.ca/wise/wise01/HallingGrunbacherBiffl.pdf)), November 11, 2007, 18:00 PM,
- [45] Tony Gorschek, Nina Dzamashvili – Fogelström, "Test-case Driven Inspection of Pre-project Requirements-Process Proposal and Industry Experience Report", in proceedings of the Requirements Engineering Decision Support Workshop held in conjunction with the 13th IEEE International Conference on Requirements Engineering, 2005.
- [46] Nina D. Fogelström and Tony Gorschek, "Test-case Driven versus Checklist-based Inspections of Software Requirements – An Experimental Evaluation", WER07 - Workshop em Engenharia de Requisitos, Toronto, Canada, May 17-18, 2007, pp 116 – 126
- [47] Gilb T. and Graham D., "Software Inspection", Addison-Wesley Publishing Company, 1993.
- [48] Oliver Laitenberger, "A Survey of Software Inspection Technologies", Handbook on Software Engineering and Knowledge Engineering, Fraunhofer IESE, 2002
- [49] Laitenberger, O. and DeBaud, J-M., "An Encompassing Life Cycle Centric Survey of Software Inspection", Journal of Systems and Software, 50(1):5-31, 2000.
- [50] O. Laitenberger, C. Atkinson, M. Schlich, K. El Emam, "An experimental comparison of reading techniques for defect detection in UML design documents", The Journal of Systems and Software, vol. 53, 2000, pp. 183-204.
- [51] T.Thelin, "Empirical Evaluations of Usage-Based Reading and Fault Content Estimation for Software Inspections," in Lund Institute of Technology -Department of Communication Systems. Lund: Lund University, 2002.
- [52] Jeffrey Carver Ph.D., "Impact of Background and Experience on Software Inspections", Department of Computer Science, University of Maryland, 2003
- [53] Paulo Costa, Forrest Shull and Walcelio Melo, "Getting Requirements right: The Perspective-Based Reading technique and the Rational Unified Process", IBM, 2006. [http://www128.ibm.com/developerworks/rational/library/sep06/melo\\_costa\\_shull/index.html](http://www128.ibm.com/developerworks/rational/library/sep06/melo_costa_shull/index.html)
- [54] A. M. Davis, "Software Requirements: Objects, Functions and States", Prentice-Hall, USA, 1993. 70
- [55] J. Siddiqi, I. Morrey, R. Hibberd, G. Buckberry, "Towards a System for the Construction, Clarification, Discovery and Formalisation of Requirements", proceedings of first international conference on Requirements Engineering, IEEE, 1994, pp.230-238
- [56] V. Scott Gordon James M. Bieman, "Rapid Prototyping: Lessons Learned", IEEE, 1997
- [57] Leszek A. Maciaszek, Bruce Lee Liong and Stephen Bills, "Practical Software Engineering a Case Study Approach", England, 2005.
- [58] CQU Course web page, "Developing Entity Relationship Diagram (ERDs)", August 13, 2008, 03:40AM, Course taught in 2000, ([http://webfuse.cqu.edu.au/Courses/2006/T2/COIS20025/Assessment/Item\\_2/Part\\_A\\_Resources/erd.pdf](http://webfuse.cqu.edu.au/Courses/2006/T2/COIS20025/Assessment/Item_2/Part_A_Resources/erd.pdf))
- [59] Anu Maria, "Introduction to Modeling and Simulation", Proceedings of the 1997 Winter Simulation Conference ed. S. Andradóttir, K. J. Healy, D. H. Withers, and B. L. Nelson.
- [60] Gang Liu, Shaobin Huang and Xiufeng Piao, "Study on Requirements Testing Method Based on Alpha-Beta Cut-off Procedure", IEEE Computer Society, 2008.
- [61] Ian Sommerville and Pete Sawyer, "Viewpoints: Principles, Problems and a Practical Approach to Requirements Engineering", Technical Report Ref. CSEG/15/1997, Lancaster University, (<http://www.cs.st-andrews.ac.uk/~ifs/Research/Publications/Papers-PDF/1995-99/VPsAnnalsOfSE.pdf>).
- [62] Artem Katasonov, 'Requirements Management and Systems Engineering' Lecture 6: Requirements Validation and Verification, (ITKS451), Autumn 2008, University of Jyväskylä.
- [63] A.M. Davis, "Software Requirements: Objects, Functions and States", Prentice Hall, 1993.
- [64] E. Bryne, "IEEE Standard 830: Recommended Practice for Software Requirements Specification," presented at IEEE International Conference on Requirements Engineering, 1994.
- [65] L. Rosenberg, T.F. Hammer, and L.L. Huffman, "Requirements, testing and metrics," presented at 15th Annual Pacific Northwest Software Quality Conference, 1998.
- [66] Alun Preece, "Evaluating and Validation Methods in Knowledge Engineering", In R Roy (ed), Micro-Level Knowledge Management, Morgan-Kaufman, pages 123-145, 2001.
- [67] A. Terry Bahill & Steven J. Henderson, "Requirements Development, Verification, and Validation Exhibited in Famous Failures".
- [68] G. Kotonya and I. Sommerville, Requirements Engineering: Processes and Techniques, John Wiley & Sons, 2000.
- [69] M. Jackson, 'The Machine and it's World' 1995
- [70] P. Heymans, Some thoughts about the animation of formal specifications written in the ALBERT language, Proceedings of the Doctoral Consortium of the 3rd IEEE International Symposium on Requirements Engineering (RE'97), Annapolis, MD, USA, January 6-10, 1997.
- [71] N.A.M. Maiden, S. Minocha, K. Manning, M. Ryan, "CREWS-SAVRE : systematic Scenario Generation and Use", Third International Conference on Requirements Engineering (ICRE'98), Colorado-Springs, USA, 1998.
- [72] Martin, J. and Tsai, W.T. "N-Fold Inspection: A Requirements Analysis Technique", Communications of the ACM, Vol. 33, No. 2, February, 1990, pages 225 - 232.
- [73] Wilson, W., Rosenberg, L., Hyatt, L., 'Automated Analysis of Requirement Specifications', Proceedings of International Conference on Software Engineering. (1997)
- [74] Fabbrini, F, Fusani, M, Gensi, S, Lami, G, 'An Automated Quality Evaluation for Natural Language Requirements'. Proceedings of the Seventh International Workshop on RE: Foundation for Software Quality (2001).
- [75] IEEE Std 830-1993, Recommended Practice for Software Requirement Specifications, December 2, 1993
- [76] Duran, A., Ruiz, A., Toro, M., 'An Automated Approach for Verification of Software Requirements', Proceedings of JIRA. (2001)
- [77] Koesters, Georg., Six, H., Winter, M.: Coupling Use Cases and Class Models as a Means for Validation and Verification of Requirement Specifications. Requirements Engineering. Vol. 6, pp. 3-17. (2001)

- [78] Nuseibeh, B., and S. Easterbrook. "Requirements Engineering: A Road Map." Proc. of International Conference on Software Engineering, Limerick, Ireland, June 2000: Association of Computing Machinery (ACM) Press, 2000: 37-46.
- [79] Lulu He, Dr. Jeffrey C. Carver, and Dr. Rayford B. "Using Inspections to Teach Requirements Validation" Vaughn Mississippi State University January 2008
- [80] Lulu He, Jeffrey Carver "PBR vs. Checklist: A Replication in the N-Fold Inspection Context" 2006 Proceedings of the International Symposium on Empirical Software Engineering, Sept. 21-22, 2006. <http://doi.acm.org/10.1145/1159733.1159750>
- [81] N.A.M. Maiden, M. Cisse, H. Perez & D. Manuel, 'CREWS Validation Frames: Patterns for Validating Systems Requirements'
- [82] P. Loucopoulos & V. Karakostas 'Requirement Engineering /Chapter 5 - Validating Requirements ! Site [http://www.utdallas.edu/~chung/RE/RE\\_chapters.rar](http://www.utdallas.edu/~chung/RE/RE_chapters.rar)
- [83] Francis Schneiderz, Steve hf. Easterbrook, John R. Callahan and Gerard J. Holzmann 'Validating Requirements for Fault Tolerant Systems using Model Checking'
- [84] José Reinaldo Silva & Eston Almanaça dos Santos, "APPLYING PETRI NETS TO REQUIREMENTS VALIDATION" In: 17th Int. Congress of Mechanical Engineering 2003.
- [85] Durán A., Bernárdez, B., Ruiz, A., Toro M. (1999), "A Requirements Elicitation Approach Based in Templates and Patterns" Workshop de Engenharia de Requisitos. Buenos Aires, Argentina.
- [86] Rosca, D. "An Active/Collaborative Approach in Teaching Requirements Engineering." Proc. of 30th Annual Frontiers in Education Conference. Kansas City, MO., Oct., 2000: 9-12.
- [87] Abramsky, S., Gabbay, D. & Maibaum, T, Handbook of Logic in Computer Science Vol 1: Background: Mathematical Structures', Clarendon Press ,(1992).
- [88] Stevens, R., Brook, P., Jackson, K. & Arnold, S, 'Systems Engineering: Coping with Complexity ', Prentice Hall Europe. (1998).
- [89] Carter, R., Martin, J., Mayblin, B. & Munday, M. ' Systems , Management and Change: A Graphic Guide', London: Paul Chapman Publishing/Harper and Row. (1984).
- [90] Wieringa, R. J,' Requirements Engineering: Frameworks for Understanding', Wiley. (1996).
- [91] Parnas, D. 'When to formalise. Personal Communication (Email)', 17 February 2000.
- [92] Michael Poppleton & Richard Banach' Requirements Validation by Lifting Retrenchments in B' Proceedings of the Ninth IEEE International Conference on Engineering Complex Computer Systems Navigating Complexity in the e-Engineering Age 1050-4729/04 \$20.00 © 2004 IEEE
- [93] Nichols G. 'Getting engineering changes under control' Journal of engineering design 1, 1990.
- [94] Marx K. Das Kapital, 'livre premier: le développement de la production capitaliste' 1867.
- [95] K Schmidt, L J Bannon, "CSCW: four characters in search of a context, in Studies in computer Supported cooperative work ", Theory Practice and Design, J Bowers and S Benford eds, North-Holland, 1991.
- [96] S. Khoshafian, M. Buckiewicz 'Introduction to Groupware, Workflow, and Workgroup Computing ', John Wiley & Sons, Inc. New York, 1995.
- [97] P. Baumard 'Organisations déconcertées: La gestion stratégique de la conn', Masson, 1996.
- [98] Levy Pierre, 'L'intelligence collective, La Découverte', Paris, 1997.
- [99] DornJ. "Iterative Improvement Methods for Knowledge-based Scheduling", AI Communications 8(1) pp.20-34, 1995.
- [100] Bannon, L. and Schmidt, K. (1989), "CSCW: Four Characters in Search of a Context" in Proceedings of the First European Conference on Computer Supported Cooperative Work (ECCSCW' 89), London, September 1989, pp. 13-15.
- [101] Ellis, C.A., Gibbs, S.J., and Rein, G.L. (1991), "Groupware, Some Issues and Experiences" Communications of the ACM 34(1), January 1991, pp. 38-58.
- [102] R. M. Beacker 'Readings in Groupware and Computer Supported Cooperative Work', San Mateo, CA: Morgan Kaufmann Publishers, 1993.
- [103] U. Bock ' Workflow as Groupware : A Case for Group Language ? ' Groupware'92, David D. Coleman, San Mateo, CA : Morgan Kaufmann Publishers, 1992.
- [104] J. Grudin. 'Obstacles to user involvement in software product development, with implication for CSCW» In Computer Supported Cooperation Work and Groupware. Saul Greenberg ed. Academic Press, p. 313-330, London, 1991.
- [105] P. Wilson ' Computer Supported Cooperative Work: An overview Intelligent Tutoring ', Media Vol.1, n°3, p. 103-116, 1990.
- [106] M. Saadoun, ' Le projet groupware ', édition Eyrolles, 1996.
- [107] R. N. Marshak 'Workflow White Paper: On Overview of Workflow Software ', Workflow'94, Bob Bierman, San Jose, CA: The Conference Group, 1994.
- [108] A. M. Palermo, S.C. McCreedy, ' Worflow Software: a Primer Groupware'92 ", David P. Coleman, morgan kaufmann publishers, 1992.
- [109] Boulanger D., Dubois G., « Objets et coopération de systèmes d'information », Ingénierie objet : concepts, techniques et méthodes, C. Oussalah & al, InterEditions, mai 1997.
- [110] Litwin W., « From Database Systems to Multidatabase Systems: Why and How », Proc. Of the British National Conference on Databases, Cardiff, UK, Juillet 1988, p. 161-188.
- [111] Dogac A. & al. « METU Interoperable Database System », ACM SIGMOD Record, vol. 24, n° 3, septembre 1995, p. 552-562.
- [112] Dubois G., Boulanger D., Couturier V., « Coopération d'objets pour la résolution de requêtes multibases », Actes du 20ème Congrès Inforsid, Nantes, 4-7 juin 2002, p. 199-214.

- [113] Garcia-Molina H. & al., « The TSIMMIS approach to mediation: Data models and Languages », *Journal of Intelligent Information Systems*, vol. 8, 1997, p. 117-132.
- [114] Nodine M, Fowler J, Ksiezzyk T, Perry B, Taylor M, Unruh A, « Active Information Gathering in InfoSleuth », *IJCIS*, 9:1/2, 2000, p. 3-28.
- [115] Jouanot F., DILEMMA : vers une coopération de systèmes d'information basée sur la médiation et la fusion d'objets, Thèse de doctorat, Université de Bourgogne, 2001.
- [116] Conrad S, Hasselbring W, James A.E, Kambur D, Kutsche R, Thiran Ph, « Report on the EFIS 2001 Workshop » , *Computer Journal*, 45(2), 2002, p. 249-251.
- [117] Fernandez G., Zhao L., « A Pattern Language for Federated Architecture », *Proceedings of KoalaPLOP 2000*, Melbourne, Australie, 24-26 mai 2000, p. 21-31.
- [118] Zarour N., Contribution à la modélisation de la coopération des systèmes d'information distribués et hétérogènes: Le système DAArACHE- Application aux Entreprises de production. Thèse présentée pour l'obtention de grade de docteur d'Etat en Informatique, université Mentouri de Constantine, Algérie (2004).
- [119] Salzano G, Bourret C, "Health Networks and Global Health Services: An Information System Analysis", ICSSHC2004 (The 8th International Conference on System Science in Health Care) - Health Care Systems: Public and Private Management, Université de Genève, 1-4 septembre 2004.
- [120] Elmagarmid A, Rusinkiewicz M., Sheth A (eds): "Management of Heterogeneous and Autonomous Database Systems", Morgan Kaufmann Publishers, Inc., San Francisco, California, 1999.
- [121] Salzano G, "Integration Methodology for Heterogeneous Databases", in *Heterogeneous Information Exchange and Organizational Hubs*, edited by H. Bestougeff, J.E. Dubois, B. Thurasingham, Kluwer Academic Publishers, Netherlands, pages 1-16, 2002
- [122] Scheurman, & al 'report on workshop on heterogeneous database systems' *ACM SIGMOD RECORD*, 19(4) :23-31. Held at Northwestern University, Evanston, Illinois, Dec 11-13,1989. Sponsored by NSF.
- [123] Pascale Zaraté, 'Des Systèmes Interactifs d'Aide à la Décision Aux Systèmes Coopératifs d'Aide à la Décision : Contributions conceptuelles et fonctionnelles' INSTITUT NATIONAL POLYTECHNIQUE DE TOULOUSE (INPT) Habilitation Soutenue le 7 Décembre 2005 UMR 5505.
- [124] Midler C, 'L'auto qui n'existait pas, Management des projets et transformation de l'entreprise', Ed. DUNOD Collection Stratégies et Management, ISBN 2.10.004228.9, 1993.
- [125] Brodie, M.L, S. Ceri, "On Intelligent and Cooperative Information Systems" ,in *International Journal of Intelligent and Cooperative Information Systems* 1, 2 September 1992.
- [126] Boughzala I, Zacklad M, Matta N, 'Gestion des connaissances dans une entreprise Etendue', 2001, Journées francophones d'extraction et de gestion des connaissances, édition Hermès, Nantes, France, 18-19 Janvier 2001, pp 259-270.
- [127] Hérin D, Espinasse B, Andonoff E, Hannachi C, « Des systèmes d'informations coopératifs aux agents informationnels », chap. 8 du livre *Ingénierie des systèmes d'information*, Hermès, 2001.
- [128] Barret, S., Konsynski, B. (1982), « Inter Organizational Information Sharing Systems », *MIS Quarterly*, Special Issue, p. 93-105.
- [129] Huhns, "Distributed Artificial Intelligence, Volume II", *Cooperative Information Systems*, Vol. 1, No. 2, June 1992.
- [130] Mhamed Saidane 'un système de Patrons pour les Architectures Coopératives '2006
- [131] Jacqueline Vacherand-Revel 'Enjeux de la médiatisation du travail coopératif distribué dans les équipes de projets de conception' 2007
- [132] Françoise darses 'convergé vers une solution en situation coopérative de conception : analyse cognitive du processus d'argumentation' 2001.
- [133] V. Baudin 'Travail coopératif' 2007
- [134] Juan F. Sequeda "A Taxonomy of Verification and Validation of Software Requirement and Specifications" 2008
- [135] Fabbri, F., Fusani, M., Gervasi, V., Gnesi, S., Ruggieri, S.: 'Achieving Quality in Natural Language Requirements ' *Proceedings of the 11 th International Software Quality Week (1998)*.
- [136] Artem Katasonov 'Lecture 6: Requirements Validation and Verification' *Requirements Management and Systems Engineering (ITKS451)*, Autumn 2008 University of Jyväskylä.
- [137] MODELISATION DE L'ACTION Théorie des actions chapitre 3 [ihm.imag.fr/nigay/ENSEIG/RICM3/ Chap3-TheorieAction.pdf](http://ihm.imag.fr/nigay/ENSEIG/RICM3/Chap3-TheorieAction.pdf)
- [138] Stylianou, A., Kumar, R., Khouja, M. 'A Total Quality Management-based Systems Development Process' *The DATA BASE for Advances in Information Systems*, Vol. 28, 1997, No. 3, pp. 59-71.
- [139] In, H., Olson, D., Rodgers, T. 'Multi-criteria Preference Analysis for Systematic Requirements Negotiation' In *Proceedings of the 26th Annual International Computer Software and Applications Conference (COMPSAC'02)*, Oxford, England, 2002, pp. 887-892. IEEE Computer.
- [140] Bouchereau, V., Rowlands, H. 'Quality Function Deployment: The Unused Tool' *Engineering Management Journal*, Vol. 10, 2000, No. 1, pp. 45-52.
- [141] C. Sibertin-Blanc, Université Toulouse 1 cour GL chapitre : La spécification
- [142] D. Essame 'La méthode B et l'ingénierie système : Répondre à un appel d'offre'. 2007.
- [143] Pascal Fenkam , All 'Visual Requirements Validation: Case Study in a Corba-Supported Environment' 2002.
- [144] Sutcliffe, All 'Validating functional system requirements with scenarios '2002.
- [145] Amador Duran 'Supporting Requirements Verification Using XSLT' 2002.
- [146] Fleisch, W 'Applying use cases for the requirements validation of component-based real-time software' 1999.
- [147] Jean-Pierre Corriveau 'Instrumentation automatisée des contrats et des scénarios de validation des exigences en. Net '2008.
- [148] Joumana argham 'A Navigational Web Requirements Validation Through Animation' 2008.

- [149] Srihari Sukumaran, "A rigorous approach to requirements validation" 2006.
- [150] Boehm, B. |Grünbacher, P. |Briggs, R.: Developing Groupware for Requirements Negotiation: Lessons Learned. IEEE Software, Vol. 18, 2001, No. 3, pp. 46-55.
- [151] [http://wiki.univ-paris5.fr/wiki/Taxonomie\\_de\\_Bloom](http://wiki.univ-paris5.fr/wiki/Taxonomie_de_Bloom).
- [152] Fairley, R., Software Engineering Concepts, McGraw-Hill, New York, 1985.
- [153] M. Amroune. ' *Une approche cognitive centrée utilisateurs pour l'élicitation des besoins dans les organisations distribuées* ', *Mémoire de Magister*, Soutenu le 19 Novembre 2007, Université de Tebessa
- [154] Yagoub Mohamed Lamine ' Ingénierie des besoins dans les systèmes d'information coopératifs- La phase de spécification des besoins '. *Mémoire de Magister*, Soutenu le 11 Février 2009, Centre universitaire d'El Oued.
- [155] JOSÉ E, NORA K, ' Requirements Engineering for Web Applications – A Comparative Study, Journal of Web Engineering ', Vol. 2, No.3 (2004) 193-212, Rinton Press.
- [156] Maritta Heisel, Jeanine Souquières 'De l'élicitation des besoins à la spécification formelle' *11ème Technique et sciences informatique*. Vol.18, no 7/1999, pages 1 à 25.

## Résumé

La validation des besoins est un problème dans les systèmes d'information traditionnels notamment plus complexe dans les systèmes coopératifs, pour la simple raison qu'il n'y a aucune méthode ou démarche de validation rigoureuse à utiliser quant à la spécification.

La littérature la présente comme une liste de bonne pratique basée sur l'utilisation d'un grand nombre de techniques hétérogènes sans pour autant déterminer le Quoi ? Qui ? Comment ? Quand ? ... Mais en fait, l'assurance de la qualité pose de nombreux problèmes, tels que le budget, le délai, et la conformité.

Afin d'éviter tous ces problèmes, nous proposons une méthodologie de validation des exigences collaborative dans les organisations distribuées qui s'articule autour d'un processus continu, progressif, collaboratif, distribué multi point de vue. Par ailleurs, le but de cette méthodologie générale qui englobe l'ensemble des activités requises est de Vérifier, Négocier, Valider et faire Évoluer un système apportant une solution économique et performante aux besoins d'un client tout en satisfaisant l'ensemble des parties prenantes.

## Summary

The requirements validation is a problem in traditional information systems including more complex in the cooperative systems, for the simple reason that there is no method or rigorous approach to validation when used in the specification.

The literature presents it as a list of good practice based on the use of a large number of heterogeneous techniques without determining the what? Who? How? When? ... But in fact, the quality assurance poses many problems, such as budget, time, and conformity.

To avoid these problems, we propose a methodology for validating requirements collaborative in distributed organizations that revolves processes is continuous, progressive, collaborative, distributed multi-view. Furthermore, the purpose of this general methodology that encompasses all activities required to verify negotiate, validate and develop a system providing efficient and economical solution to a client's needs while satisfying all stakeholders.

## ملخص

المصادقة على متطلبات مشكلة في أنظمة المعلومات التقليدية بما في ذلك الأنظمة التعاونية الأكثر تعقيدا، لسبب بسيط هو أنه لا يوجد أسلوب أو نهج للمصادقة حازم عندما تستخدم في مواصفات الأعمال في هذا المجال تعرضها على أنها قائمة على الممارسة الجيدة على أساس استخدام عدد كبير من التقنيات الغير متجانسة من دون تحديد لماذا؟ من؟ كيف؟ متى؟ ... ولكن في الواقع، وضمانا للجودة تطرح مشاكل كثيرة، مثل الميزانية، والوقت، والامتثال لتجنب هذه المشاكل، فإننا نقترح وضع منهجية لعملية المصادقة على الاحتياجات موزعة في المنظمات التعاونية مستمرة تدريجية، تعاونية - ومتعددة تحتمل الآراء المختلفة. وعلاوة على ذلك، فإن العرض من هذه المنهجية العامة التي تشمل جميع الأنشطة المطلوبة للتحقق من التفويض، والمصادقة من وضع نظام فعال واقتصادية توفير حل لاحتياجات العميل بينما ترضى جميع الأطراف المعنية.