

UNIVERSITÉ DE MONTRÉAL

PROCESSUS RUP© ET CONCEPTION CENTRÉE SUR L'UTILISATEUR : UNE
ÉTUDE DE CAS

FRANÇOIS LEMIEUX
DÉPARTEMENT DE GÉNIE INFORMATIQUE
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION
DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES
(GÉNIE INFORMATIQUE)
AOÛT 2005

© François Lemieux, 2005.



Library and
Archives Canada

Bibliothèque et
Archives Canada

Published Heritage
Branch

Direction du
Patrimoine de l'édition

395 Wellington Street
Ottawa ON K1A 0N4
Canada

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

ISBN: 978-0-494-16807-3

Our file Notre référence

ISBN: 978-0-494-16807-3

NOTICE:

The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

AVIS:

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protègent cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.


Canada

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé:

PROCESSUS RUP© ET CONCEPTION CENTRÉE SUR L'UTILISATEUR : UNE
ÉTUDE DE CAS

présenté par: LEMIEUX François

en vue de l'obtention du diplôme de: Maîtrise ès sciences appliquées

a été dûment accepté par le jury d'examen constitué de:

M. ROBILLARD, Pierre N., Ph.D., président

M. DESMARAIS, Michel, Ph.D., membre et directeur de recherche

M. ROBERT, Jean-Marc, Doctorat, membre

REMERCIEMENTS

La réalisation d'une recherche ne se fait pas sans le précieux concours de personnes qui soutiennent la démarche. Je remercie donc :

- Michel C. Desmarais, professeur en Génie informatique à l'École polytechnique de Montréal, pour ses conseils avisés, les révisions du manuscrit et la confiance qu'il m'a témoigné en me confiant des charges de cours en gestion de projet informatique et en atelier logiciel.
- Guy Leblanc de la compagnie ErgoIGL, ergonomiste qualifié, tacticien et rigoureux qui m'a parrainé professionnellement et qui m'a permis d'effectuer des cycles ergonomiques, chose trop peu fréquente dans l'industrie du logiciel.
- Élane et Diane pour leur accord et leur précieuse collaboration.
- Murielle et Daniel pour leur aide indispensable ainsi que Benoît.
- Mes parents, Adèle et Jean, qui n'ont rien ménagé pour que je travaille dans des conditions idéales.
- Mes frères, Pierre et Benoît, et leurs familles, pour le soutien qu'ils m'ont apporté.
- Ivan Maffezzini, professeur au département d'informatique de l'université du Québec à Montréal, qui m'a encouragé à m'orienter dans cette discipline de la conception centrée sur l'utilisateur encore mal intégrée au génie logiciel.

RÉSUMÉ

Le *Rational Unified Process* (RUP©) est un processus de cycle de vie du logiciel. La conception centrée sur l'utilisateur est un processus qui permet de s'assurer qu'un logiciel répond aux besoins de l'utilisateur pour lequel il est conçu. La présente recherche est une étude de cas qui établit une comparaison entre les principes de la conception centrée sur l'utilisateur dictés par la norme ISO 13407 et les activités et artefacts prescrits par le RUP©. Cette comparaison est poussée au niveau de l'interprétation et de l'application faites du RUP© par une équipe du programme et de l'application de principes de conception centrée sur l'utilisateur faite lors d'une intervention ergonomique requise par les représentants des utilisateurs.

Les études portant sur les facteurs de réussite des projets de développement de système informatique interactif mettent en lumière que plus de 80 % de ces projets connaissent des difficultés importantes. Dans près de la moitié des cas, la gestion des exigences, l'implication des utilisateurs et le soutien de la direction sont responsables de ces échecs. Compte tenu de l'importance économique que revêt l'informatisation des activités dans nos sociétés, il est pertinent de se pencher sur la façon dont sont développés les systèmes informatiques interactifs. Or, on a recours à un processus qui définit une certaine quantité d'activités à exécuter selon un ordre prédéterminé pour développer un système informatique.

Pour s'assurer qu'un système est réponde aux besoins de l'utilisateur, on a aussi recours à un processus. La norme ISO 13407 définit ce type de processus qui ne prétend pas fixer les règles pour construire un système interactif, mais qui définit un certains nombre d'activités complémentaires à un processus de développement. La norme se fonde sur quatre grands principes : la participation active des utilisateurs et une compréhension claire des exigences liées à l'utilisateur et à la tâche, une allocation appropriée des fonctions entre les utilisateurs et la technologie, l'itération des solutions de conception et une conception pluridisciplinaire.

Le *Rational Unified Process*, propriété de la compagnie IBM©, est un processus de développement logiciel perçu comme une norme dans l'industrie. C'est un processus centré sur l'architecture et piloté par des cas d'utilisation. Les cas d'utilisation sont présentés sous la forme de scénarios et décrivent tous les types d'utilisation d'un système. Le RUP© est itératif et incrémental.

Des recherches ont porté sur l'intégration de la conception centrée sur l'utilisateur aux méthodes du génie logiciel en général et du RUP© en particulier. Il en ressort que le processus RUP© soutient mal la conception centrée sur l'utilisateur définie par ISO 13407 et que des adaptations seraient nécessaires.

Chacun des principes de la norme ISO 13407 est ici approfondi et l'interprétation qu'en fait le RUP© est scrutée. Ainsi, la compréhension des exigences-utilisateurs se traduit pour ISO 13407 par une analyse du contexte d'utilisation. L'environnement et la description des utilisateurs, selon le RUP©, se font lors de la description du système et la tâche est décrite par les cas d'utilisation et par les caractéristiques de produit.

ISO 13407 demande que les exigences liées à l'utilisateur et à l'organisation soient distinguées parmi les exigences fonctionnelles et autres. Le RUP© évoque plutôt des exigences d'automatisation qui sont les portions de processus d'affaires qui doivent être automatisées. L'itération des solutions de conception est abordée en trois parties : la matérialisation des solutions, la gestion des itérations et l'évaluation des solutions. Pour ISO 13407, la matérialisation se fait à partir des connaissances acquises, plus précisément du contexte d'utilisation, alors que le RUP© prévoit la formulation de lignes directrices en début de projet qui servent, par la suite, à la réalisation des interfaces-utilisateurs. Les itérations doivent être planifiées selon ISO 13407 et elles doivent se poursuivre jusqu'à ce que les objectifs de conception soient atteints. Le RUP© ne prévoit pas une planification des itérations pour la conception d'interfaces-utilisateurs. L'évaluation des solutions est étroitement liée à la conception pour ISO 13407 et doit faire l'objet d'un plan de surveillance. Pour le RUP©, les tests d'utilisabilité visent à exposer les prototypes aux parties prenantes afin d'obtenir leur accord et pour pallier les défaillances. ISO 13407 prescrit une conception pluridisciplinaire ou elle prévoit notamment faire appel aux connaissances scientifiques des spécialistes des facteurs humains. Pour le RUP©, la conception est faite par des travailleurs dont les rôles peuvent être interchangeables.

L'étude de cas est une étude de cas unique qui utilise les méthodes exploratoire, descriptive et explicative. L'approche de comparaison de cas de l'analyse comparative intra cas est utilisée pour l'analyse des données. Les unités d'analyse sont définies par six questions de recherche qui portent sur les principes et les activités prescrits par ISO 13407.

La recherche s'est effectuée dans une entreprise de 6 000 employés qui déclare un bénéfice annuel de 1,4 G\$ CA. Elle a porté sur un projet de développement d'un système informatique visant à assurer les opérations de gestion de l'entreprise. Le projet étudié comptait une cinquantaine de cas d'utilisation qui formait la première application à développer parmi 13 autres qui constituaient le système. L'étude de cas compare l'application du RUP© qu'a faite l'équipe du programme pour la conception des interfaces-utilisateurs avec une intervention ergonomique qui a été faite dans le projet selon les principes de ISO 13407 et les prescriptions du RUP©.

Il en ressort que l'application faite par l'équipe du programme ne prévoyait pas d'analyse du contexte d'utilisation et qu'on ne distinguait pas les fonctions de l'utilisateur de celles de la technologie. Les interfaces-utilisateurs étaient conçues à partir de guides plutôt qu'à partir du contexte d'utilisation, il n'y avait pas de planification de l'itération des solutions de conception et l'évaluation des solutions de conception se faisait par des revues plutôt que par plusieurs types d'évaluation. Enfin, l'équipe du programme ne reconnaissait pas la nécessité d'une équipe distincte formée de personnel qualifié comme le recommande ISO 13407. L'adaptation faite du processus était en conformité avec le RUP©.

Par contre, le déroulement de l'intervention ergonomique démontre qu'il est possible de faire une conception centrée sur l'utilisateur dans le cadre du RUP©. L'adaptation de certains artefacts et la formalisation d'enchaînements d'activités distincts pour la conception centrée sur l'utilisateur faciliteraient les choses.

ABSTRACT

The Rational Unified Process (RUP©) is a lifecycle software process. The user centered design is a process describing how a software application can meet the user needs for which it is designed. This research is a case study establishing a comparison between the ISO 13407 user centered design principles and the activities and artifacts prescribed by the RUP©. Beyond what each process prescribes, the comparison also extends to pragmatic issues around the application of the processes, namely the project team's interpretation of the RUP© interpretation and application levels done by a project team and at the user and the application of user centered design principles during a usability engineering intervention requested by the user representatives.

Surveys on the causes of success of interactive system development projects show that more than 80 % of these projects fail. For half of them, lacks of a clear statement of requirements, of user involvement, and of executive management support were responsible of these failures. The economic consequences of computerizing activities in our societies lead us to evaluate how interactive computer systems are developed. For computer system development, a process, defining a number of activities to perform according to a predefined order, is used.

To ensure that a system meets the user needs, a process is also needed. The ISO 13407 standard defines such a process. The standard does not pretend to establish the rules for building a system in his entirety, but it states a set of complementary activities to a development. The standard states four main principles: the active involvement of users and a clear understanding of user and task requirements, an appropriate allocation of function between users and technology, the iteration of design solutions and multi-disciplinary design.

The Rational Unified Process, owned by IBM©, is a software development process, a standard for the industry, that is architecture-centric and use-case driven. Use cases are scenarios describing all kinds of system uses. RUP© is iterative and incremental.

Studies on the integration of user centered design into software engineering methods, or, more specifically, into RUP© have been done. The conclusions show that RUP© is failing to support user centered design as ruled by ISO 13407 and that, consequently, adaptations are required.

How RUP© conforms to each ISO 13407 principle is presented here. First, clear understanding of user and task requirements is considered for ISO 13407 as specifying the context of use.

According to the RUP©, environment and user description, is done during the system description and tasks are described by use cases and product characteristics.

Second, ISO 13407 states that, among functional requirements and others, user and organisational requirements must be distinguished. For its part, RUP© identifies automation requirements which are business process portions needing that need to be automated. Third, design solutions iteration is processed in three parts: design solution materialization, iterations management and design solutions evaluation. According to ISO 13407, materialization is done according to acquired knowledge, especially context of use, whereas RUP© prescribes creating guidelines at the beginning of the project for user interface the later design of user interface. Iterations must be planned according to ISO 13407 and must be done until design goals are reached. RUP© does not have a formal iteration plan for user interface design. Design solutions evaluation is narrowly linked to design according to ISO 13407 and long term monitoring must be stated. According to RUP©, usability testing are useful in order to expose user interface prototype to stakeholders to obtain their agreement and to resolve defects. Finally, ISO 13407 rules that design should be multi-disciplinary, involving among others, human factors skilled people. According to RUP©, design is done by workers who can take on multiple roles.

The case study is a single case study using exploratory, descriptive and explicative methods. Comparison approach of comparative intra case analysis is used for data analysis. Analysis units correspond to six research questions addressing ISO 13407 principles and activities.

The survey has been done in a company of 6 000 employees with an annual income of 1,4 G\$ CA. The system developed has to process all the company's management operations. The project had around fifty (50) use cases in one application of the thirteen (13) applications envisioned for the whole system. The case study compares the RUP© application done by the project team for the user interface design with the usability engineering intervention done into the project as like ISO 13407 principles and activities and RUP© prescriptions.

The case study analysis shows that the project team application ignored the context of use analysis and that the distinction between user requirements was never done as user functions and, technological functions were not even identified. The user interfaces were designed from guidelines rather from the context of use. There was no design solutions iteration planning and design solution evaluation was done through reviews rather than through various types of

evaluation. Finally, the project team did not recognize the necessity of a distinct group of skilled people as ruled by ISO 13407.

On the other hand, the success of usability engineering intervention into the project shows that a user centered design is possible with RUP©.

TABLE DES MATIÈRES

REMERCIEMENTS	iv
RÉSUMÉ	v
ABSTRACT	viii
TABLE DES MATIÈRES	xi
TABLE DES TABLEAUX.....	xviii
TABLE DES FIGURES	xx
SIGLES ET ABRÉVIATIONS.....	xxii
LISTE DES ANNEXES	xxiii
INTRODUCTION	1
CHAPITRE I DÉVELOPPEMENT ET CONCEPTION CENTRÉE SUR L'UTILISATEUR	7
1.1 Le développement d'un système informatique	7
1.1.1 La gestion des exigences, principale difficulté	10
1.1.2 La spécification des exigences.....	13
1.1.3 Les exigences-utilisateurs	13
1.2 Les interfaces-utilisateurs	14
1.3 La conception centrée sur l'utilisateur	15
1.3.1 La norme ISO 13407.....	15
1.3.2 Le contexte d'utilisation	17
1.3.3 Le prototypage et le maquettage	19
1.3.4 Les tests d'utilisabilité	19
1.3.5 Une approche itérative	20
1.3.6 Une conception pluridisciplinaire	21
1.4 Le <i>Rational Unified Process</i>	21

1.4.1	Un processus centré sur l'architecture	21
1.4.2	Un processus piloté par les cas d'utilisation	22
1.4.3	L'utilisabilité et les cas d'utilisation	24
1.4.4	Les phases, les processus et les itérations	25
1.4.5	La conception des interfaces-utilisateurs	27
1.4.5.1	Les activités de conception des interfaces-utilisateurs	28
1.4.5.2	Le concepteur d'interface-utilisateur	32
CHAPITRE II PROBLÉMATIQUE ET RECHERCHES CONNEXES.....		34
2.1	Génie logiciel et ingénierie de l'utilisabilité	34
2.2	ISO 13407 et l'ingénierie de l'utilisabilité.....	36
2.3	Conception centrée sur l'utilisateur et le RUP©.....	37
2.4	ISO 13407 et le RUP©	39
2.5	Principes de conception centrée sur l'utilisateur.....	40
CHAPITRE III ISO 13407 ET LE RUP©		43
3.1	Participation active des utilisateurs et compréhension claire des exigences liées à l'utilisateur et à la tâche	43
3.1.1	La participation des utilisateurs	43
3.1.2	ISO 13407 : les utilisateurs, l'environnement et la tâche	44
3.1.2.1	Les utilisateurs et l'environnement.....	44
3.1.2.2	La tâche.....	44
3.1.3	RUP© : participation souhaitable et plusieurs contextes.....	46
3.1.3.1	La participation des utilisateurs	46
3.1.3.2	Les contextes	47
3.1.3.3	L'environnement: Vision de métier et Vision	47
3.1.3.4	Les utilisateurs.....	48

3.1.3.5	Les rôles.....	48
3.1.3.6	Les tâches.....	49
3.2	Répartition appropriée des fonctions entre les utilisateurs et la technologie	52
3.2.1	ISO 13407 : exigences liées à l'utilisateur et à l'organisation.....	52
3.2.2	RUP© : exigences d'automatisation.....	53
3.3	Itération des solutions de conception.....	53
3.3.1	Produire les solutions de conception et les présenter aux utilisateurs	54
3.3.1.1	ISO 13407 : utiliser les connaissances acquises	54
3.3.1.2	RUP© : normes et guides	54
3.3.2	Gérer l'itération des solutions de conception.....	54
3.3.2.1	ISO 13407 : itérations pour la conception	55
3.3.2.2	RUP© : itérations planifiées du cycle de vie	55
3.3.3	Évaluer les conceptions par rapport aux exigences	55
3.3.3.1	ISO 13407 : conception et surveillance	55
3.3.3.2	RUP© : pallier les défaillances.....	56
3.4	Conception pluridisciplinaire.....	57
3.4.1	ISO 13407 : facteurs humains et ergonomie.....	57
3.4.2	RUP© : travailleurs, rôles et qualification.....	58
3.4.3	Comparaison entre ISO 13407 et le RUP©	62
CHAPITRE IV MÉTHODE		63
4.1	Les objectifs de la recherche.....	63
4.2	Les questions de recherche	64
4.2.1	Participation active des utilisateurs et compréhension claire des exigences liées à l'utilisateur et à la tâche	64
4.2.2	Répartition appropriée des fonctions entre les utilisateurs et la technologie	64

4.2.3	Itération des solutions de conception.....	65
4.2.3.1	Produire les solutions de conception.....	65
4.2.3.2	Gérer l'itération des solutions de conception.....	65
4.2.3.3	Évaluer les conceptions par rapport aux exigences	65
4.2.4	Conception pluridisciplinaire.....	66
4.3	Protocole de la recherche	66
4.3.1	Les procédures de terrain	67
4.3.2	L'accès au site.....	67
4.3.3	Confidentialité	67
4.3.4	Les sources de données.....	68
4.4	Le format de présentation de l'étude de cas.....	69
CHAPITRE V DESCRIPTION DU CAS.....		70
5.1	Le contexte : l'entreprise et le système à développer	70
5.2	Le processus de développement utilisé.....	72
5.3	L'application : la gestion des réseaux de distribution.....	84
5.3.1	Description de l'application.....	85
5.3.2	Partie de l'application traitée par le cycle ergonomique.....	86
5.4	La nature de l'intervention en conception centrée sur l'utilisateur.....	88
5.4.1	La conception des interfaces-utilisateurs par les analystes.....	90
5.4.2	La conception des interfaces-utilisateurs par les ergonomes.....	90
5.4.3	Les activités complémentaires à l'intervention.....	93
CHAPITRE VI ANALYSE DES DONNÉES		100
6.1	(1) Est-ce que l'application du RUP© soutient la participation active des utilisateurs et définit le contexte d'utilisation selon ISO 13407 ?	100

6.1.1	Équipe du programme : architecture de référence, DPO et caractéristiques logicielles.....	100
6.1.1.1	Une participation systématique des utilisateurs.....	100
6.1.1.2	L'environnement est partiellement défini.....	101
6.1.1.3	Les acteurs décrits par leur rôle	101
6.1.1.4	Les tâches : DPO et cas d'utilisation	102
6.1.2	Intervention ergonomique : analyse de contexte.....	103
6.1.2.1	La participation des utilisateurs	103
6.1.2.2	Les environnements	103
6.1.2.3	La description des utilisateurs.....	104
6.1.2.4	L'analyse contextuelle de la tâche	104
6.1.3	Synthèse de la question (1)	105
6.2	(2) L'application du RUP© permet-elle une bonne spécification des exigences-utilisateurs notamment par une bonne répartition des fonctions entre les utilisateurs et la technologie selon ISO 13407 ?	105
6.2.1	Équipe du programme : caractéristiques logicielles non fonctionnelles.....	105
6.2.2	Intervention ergonomique : exigences incluses dans la spécification d'interface	106
6.2.3	Synthèse de la question 2.....	106
6.3	Itération des solutions de conception.....	106
6.3.1	(3) La matérialisation des solutions de conception se fait-elle par l'application du RUP© selon ISO 13407 ?	107
6.3.1.1	Équipe du programme : un cadre de conception.....	107
6.3.1.2	Intervention ergonomique : plusieurs sources	107
6.3.1.3	Synthèse de la question 3.....	108

6.3.2	(4) L'approche itérative de ISO 13407 est-elle respectée lors de l'application du RUP© ?.....	108
6.3.2.1	Équipe du programme : pas d'itération planifiée.....	108
6.3.2.2	Intervention ergonomique : plusieurs itérations	109
6.3.2.3	Synthèse de la question 4.....	109
6.3.3	(5) L'évaluation des solutions de conception retenue lors de l'application du RUP© est-elle conforme à celle qui est prescrite par ISO 13407 ?	109
6.3.3.1	Équipe du programme : des revues.....	109
6.3.3.2	Intervention ergonomique : plusieurs types d'évaluation	110
6.3.3.3	Synthèse de la question 5.....	111
6.4	(6) L'application du RUP© reconnaît-elle la nécessité d'un personnel qualifié selon ISO 13407 ?.....	111
6.4.1	Équipe du programme : des analystes fonctionnels.....	111
6.4.2	Intervention ergonomique : qualifications en ergonomie et en informatique..	111
6.4.2.1	Synthèse de la question 6.....	112
6.5	Comparaison entre ISO 13407 et les autres approches.....	112
CHAPITRE VII DISCUSSION.....		118
7.1	(1) Est-ce que l'application du RUP© soutient la participation active des utilisateurs et définit le contexte d'utilisation selon ISO 13407 ?	118
7.1.1	La participation des utilisateurs	118
7.1.2	L'environnement.....	119
7.1.3	Les utilisateurs	119
7.1.4	La tâche.....	119
7.2	(2) L'application du RUP© permet-t-elle une bonne spécification des exigences-utilisateurs notamment par une bonne répartition des fonctions entre les utilisateurs et la technologie selon ISO 13407 ?	121

7.3	(3) La matérialisation des solutions de conception se fait-elle par l'application du RUP© selon ISO 13407 ?	121
7.4	(4) L'approche itérative de ISO 13407 est-elle respectée lors de l'application du RUP© ?	122
7.5	(5) L'évaluation des solutions de conception retenue lors de l'application du RUP© est-elle conforme à celle qui est prescrite par ISO 13407 ?.....	123
7.6	(6) L'application du RUP© reconnaît-elle la nécessité d'un personnel qualifié selon ISO 13407 ?.....	124
CONCLUSION.....		125
RÉFÉRENCES		128
ANNEXES.....		134

TABLE DES TABLEAUX

Tableau 1.1 : Facteurs d'échec de projet (Standish, 1995).....	4
Tableau 2.1 : Cas d'utilisation essentiel (Constantine et Lockwood, 2001).....	35
Tableau 2.2 : Différences entre ISO TR 18529 et les approches de l'ingénierie de l'utilisabilité (Earthy, Sherwood-Jones et Bevan, 2001)	37
Tableau 3.1 : Exemple de caractéristiques du contexte d'utilisation (ISO, 1998).....	45
Tableau 3.2 : Se familiariser avec les utilisateurs, leurs tâches et exigences: suggestions (Rational 2002).....	46
Tableau 3.3 : Contextes et artefacts RUP© (Rational, 2002)	47
Tableau 3.4 : Types de demandes de partie prenante (Rational 2002)	48
Tableau 3.5 : Propriétés d'acteur (Rational, 2002).....	49
Tableau 3.6 : Éléments de définitions des exigences-utilisateurs (ISO 1999).....	52
Tableau 3.7 : Fonctions représentées dans les équipes pluridisciplinaires (ISO 1999)	57
Tableau 3.8 : Travailleurs et conception d'interface-utilisateur (Rational, 2002, Kruchten, 2001).	60
Tableau 3.9 : Tableau comparatif de ISO 13407 et du RUP©.....	62
Tableau 5.1 : Distribution des effectifs du programme Neptune.	71
Tableau 5.2 : Équivalences des termes utilisés par le RUP© et son adaptation.	73
Tableau 5.3 : Gabarit de l'adaptation de l'artefact des cas d'utilisation.....	80
Tableau 5.4 : Gabarit de l'adaptation du scénarimage.	81
Tableau 5.5 : Rôles selon l'équipe du programme.	84
Tableau 5.6 : Caractéristiques logicielles non fonctionnelles.	85
Tableau 5.7 : Sommaire du modèle des cas d'utilisation de l'application.	85
Tableau 5.8 : Cas d'utilisation traités par le cycle ergonomique.....	87

Tableau 5.9 : Scénarimages du cycle ergonomique.	87
Tableau 5.10 : Personnel ayant produit des artefacts traités par le cycle ergonomique.....	88
Tableau 5.11 : Personnel ayant révisé des artefacts.....	88
Tableau 6.1 : Comparaison entre ISO 13407 et les différentes approches.	113

TABLE DES FIGURES

Figure I.1: Caricature de Handelsman, © 1993, The New Yorker Magazine Inc.....	2
Figure 1.1 : Le modèle en cascade (Boehm, 1988).....	9
Figure 1.2 : Coûts relatifs de correction des exigences selon le stade où elles sont réalisées (Young, 2001).	10
Figure 1.3 : Cycle des activités de conception centrée sur l'opérateur humain (ISO, 1999).....	16
Figure 1.4 : Structure d'utilisabilité (ISO, 1998).....	18
Figure 1.5 : Les itérations en RUP© (Kruchten, 2001).	25
Figure 1.6 : Phases du cycle de vie selon le RUP© (Kruchten, 2001).....	26
Figure 1.7 : Exemples de concepts (Rational, 2002).	28
Figure 1.8 : Enchaînement des activités de gestion des exigences (Kruchten, Ahlqvist et Bylund, 2001) et (Kruchten, 2001).	29
Figure 1.9 : Diagramme de collaboration de scénarimage (Kruchten, Ahlqvist et Bylund, 2001).	30
Figure 1.10 : Travailleurs et artefacts dans l'enchaînement des activités de gestion des exigences (Kruchten, 2001).	32
Figure 2.1 : Relations entres les enchaînements, les travailleurs et les artefacts du RUPI (Sousa et Furtado, 2003b).	38
Figure 2.2 : Activités supplémentaires des activités de modélisation du métier du RUPI (Sousa et Furtado, 2003c).	39
Figure 3.1 : Relations entre contextes (Rational, 2002).....	47
Figure 3.2 : Demandes des parties prenantes (Kruchten, 2001).	50
Figure 3.3 : Des modèles métier aux modèles système (Kruchten, 2001).	51
Figure 3.4 : Ressources et travailleurs (Kruchten, 2001).....	58
Figure 5.1 : Organigramme de l'équipe du programme Neptune.	72

Figure 5.2 : Activités d'assurance-qualité.	74
Figure 5.3 : Modèle de développement du projet.	75
Figure 5.4 : Artefacts produits par le centre des exigences et transition.....	76
Figure 5.5 : Exemple d'un DPO.	77
Figure 5.6 : Adaptation du cheminement d'activités de la modélisation d'affaire.	77
Figure 5.7 : Adaptation du cheminement d'activités de la définition des exigences.....	78
Figure 5.8 : Diagramme UML du modèle des cas d'utilisation de l'application.....	79
Figure 5.9 : Exemple d'un diagramme de navigation.....	82
Figure 5.10 : Activités prescrites pour la production des scénarimages.	83
Figure 5.11 : Activités d'assurance-qualité des scénarimages.	83
Figure 5.12 : Exemple de diagramme UML du modèle de cas d'utilisation d'un sous-module ..	86
Figure 5.13 : Processus de production des artefacts avant l'intervention.	90
Figure 5.14 : Cycle de vie de l'ingénierie de l'utilisabilité (Mayhew, 1999).	91
Figure 5.15 : Processus de production des artefacts lors de l'intervention ergonomique.....	92
Figure 5.16 : Activités de l'intervention ergonomique.....	94

SIGLES ET ABRÉVIATIONS

ConOps	Concept des opérations
DPO	Diagrammes de processus opérationnels
IEEE	Institute of Electrical and Electronics Engineers
ISO	Organisation internationale de normalisation
ISO/IEC	<i>International Organization for Standardization / International Electrotechnical Commission</i>
MCC	Modèle des concepts clés
OTAN	Organisation du traité de l'Atlantique Nord
PCCU	Processus de conception centrée sur l'utilisateur
RUP©	Rational Unified Process
RUPI	Rational Unified Process Interactive
SEL	Spécification d'exigences logicielles
SES	Spécification d'exigences système
SWEBOK	Software Engineering Body of Knowledge
TEV	Terminaux à écrans de visualisation
UML	<i>Unified Modeling Language</i>
WIMP	Windows, icons, menus and pointers

LISTE DES ANNEXES

ANNEXE I SOURCES DE DONNÉES TRIÉES PAR ORDRE ALPHABÉTIQUE.....	134
ANNEXE II SOURCES DE DONNÉES GROUPEES PAR TYPE DE DOCUMENT	138

INTRODUCTION

La conception centrée sur l'utilisateur constitue, de nos jours, un atout déterminant pour la réussite du développement d'un système informatique interactif. Or, le développement d'un tel système suppose l'application d'un processus logiciel permettant de coordonner les efforts en vue de sa réalisation. Un cas d'intégration des principes de la conception centrée sur l'utilisateur au processus de développement *Rational Unified Process* (RUP©) est ici présenté.

La présente étude cherche à déterminer en quoi ce processus, le RUP©, est ou n'est pas une conception centrée sur l'utilisateur. Une présentation du RUP© et des principes de la conception centrée sur l'utilisateur, l'étude d'un cas d'application du RUP© dans un projet de développement ainsi que l'intégration d'une conception centrée sur l'utilisateur lors de ce projet étayeront cette démonstration.

Le RUP© est un processus de développement commercialisé par la compagnie IBM©. Le processus de conception centrée sur l'utilisateur est défini par la norme internationale ISO 13407. L'étude de cas a été faite lors des phases d'opportunité et d'élaboration, telles que définies par le RUP©, du projet logiciel d'un système interactif de gestion d'une grande entreprise. L'étude de cas porte sur l'adaptation des principes de conception centrée sur l'utilisateur appliqués lors des phases de développement d'un système et, plus particulièrement, lors de l'enchaînement des activités de la discipline de définition des exigences.

L'enjeu

Aux États-Unis, il se dépense plus de 250 G\$ chaque année en développement d'applications de technologie de l'information pour approximativement 150 000 projets. Le coût moyen de développement d'un projet pour la grande entreprise était de 2,322 M\$, de 1,331M\$ pour une moyenne entreprise et de 434 000 \$ pour une petite entreprise durant les années 1990 (Standish, 1995).

L'interaction humain-ordinateur devient de plus en plus déterminante dans une société émergente de l'information à cause de la prolifération des systèmes interactifs comme outils de communication, de collaboration et d'interaction sociale entre les groupes de personnes. Les

activités humaines vont devenir de plus en plus régies par les ordinateurs. Les ordinateurs sont donc perçus comme des outils d'amélioration de la productivité (Sousa et Furtado, 2003).

L'informatique prend ainsi beaucoup de place dans nos sociétés et elle pose fréquemment des problèmes de productivité là où elle devrait les résoudre. L'introduction de l'informatique dans les activités économiques devrait normalement avoir pour effet d'améliorer la productivité.



« Pardon, messieurs dames, ce n'est pas ce que vous avez commandé, mais tous ont droit au fettucine jusqu'à ce que le programme informatique soit corrigé. »

Figure I.1: Caricature de Handelsman, © 1993, The New Yorker Magazine Inc.

Des études sur l'utilisation de l'ordinateur au bureau révèlent que la plus grande partie du temps épargné par l'automatisation est gaspillée par un logiciel qui est inutilement difficile à utiliser, imprévisible et inefficace. Les experts en conception mettent en garde contre les tendances actuelles de l'industrie à se tourner vers des programmes de plus en plus complexes ou nouveaux, dont la présentation de l'information n'a pas été testée, qui peuvent faire plus de mal que de bien et qui font certainement moins bien que ce qui est annoncé (Gibbs, 1997).

Le développement des systèmes informatiques est à ce point périlleux qu'il n'est complété dans le temps et le budget prévu que dans moins de 15 % des cas. Or, c'est l'identification des besoins des utilisateurs qui explique le plus souvent ces échecs. Lors du développement d'un système informatique, c'est à l'étape de la spécification des exigences que les besoins des utilisateurs sont définis.

L'informatique et la productivité

Thomas Landauer, diplômé de Harvard en psychologie, spécialiste de la psychologie de l'apprentissage et de l'utilisabilité, professeur à l'université du Colorado à Boulder, a dressé un bilan des problèmes de productivité des systèmes informatiques durant la période 1970 à 1990 et a proposé des solutions pour y remédier dans un ouvrage intitulé *The Trouble with Computers: Usefulness, Usability, and Productivity* (1995). Voici ce bilan.

Plusieurs spécialistes, dont Michael Dertouzos, directeur du département informatique du MIT, Stephen Roach, économiste senior chez Morgan Stanley, Martin Neil Baily, économiste de la productivité au *Brookings Institute*, Paul Attewell, sociologue du travail à *New York University* et membre du *National Research Council Committee on Human Factors*, et Paul Strassmann, ancien chef de l'informatique chez Xerox, ont conclu, au début des années 1990, que les ordinateurs ont peu d'effets positifs sur la productivité (Landauer, 1995).

Landauer (1995) constate que, par exemple, dans les compagnies de services téléphoniques, les projets de développement de systèmes informatiques ignorent les exigences et compétences présentes dans l'organisation du travail que ces systèmes devaient supporter. Ainsi, de nouveaux systèmes sont mis en place les uns après les autres avec des taux d'erreur inacceptables. Ils entraînent un accroissement net des temps de traitement de transaction. Ils nécessitent plus d'employés et augmentent le roulement de personnel. Ils rendent plus difficile aux gestionnaires la tâche d'harmoniser les opérations du système informatique aux autres opérations d'affaire. Le problème est que les concepteurs s'assurent du fonctionnement du matériel et du logiciel sans se préoccuper du problème réel qui est celui d'améliorer le travail et le fonctionnement de l'organisation. Les gestionnaires, qui achètent des systèmes sans s'assurer de leur utilité et de leur utilisabilité, achètent des problèmes.

Toujours selon Landauer (1995), l'alternative aux pratiques classiques de développement est la conception centrée sur l'utilisateur, pilotée par la conception, alimentée et encadrée par une évaluation empirique de l'utilité et l'utilisabilité.

Les facteurs d'échec des projets de développement des systèmes d'information

Le groupe Standish, un chef de file reconnu en évaluation de projets dans le domaine des technologies de l'information, fait notamment des études sur le déroulement de projets de développement de système informatique.

Un sondage, portant sur ce sujet, a été fait en 1994, auprès d'un échantillon de gestionnaires de petites, moyennes et grandes entreprises dans les principaux secteurs industriels, tels les banques, les assurances, la fabrication, la vente au détail, le commerce, la santé, les services, les organisations locales et nationales aux États-Unis. L'échantillonnage regroupait 365 répondants et portait sur 8 380 programmes logiciels (Standish, 1995).

Les résultats sont éloquentes. Les projets logiciels qui sont complétés en temps et qui respectent les prévisions budgétaires comptent pour seulement 16,2 %. Dans les grandes entreprises, seuls 9 % des projets y sont réalisés en temps et respectent les prévisions budgétaires. Et, même lorsque ces projets sont complétés, plusieurs ne sont plus que l'ombre de la spécification originale de leurs exigences. Les projets complétés par les grandes entreprises américaines ont approximativement 42 % des fonctions et caractéristiques originellement proposées. Les plus petites entreprises font cependant beaucoup mieux. Un total de 78,4 % de leurs projets logiciels sont déployés avec au moins 74,2 % de leurs caractéristiques et fonctions originales (Standish, 1995).

De cette étude, dont on voit les résultats dans le tableau 1, on retient que les trois principales raisons de l'échec d'un projet, dans près 37 % des cas, sont le manque d'implication de l'utilisateur, des exigences et spécifications incomplètes et de trop nombreux changements aux exigences et spécifications. Il y a d'autres facteurs. Mais en contrôlant ces trois éléments, les chances de réussite sont beaucoup plus grandes (Standish, 1995).

Tableau 1.1 : Facteurs d'échec de projet (Standish, 1995)

Facteur d'échec	Réponses %
1. Manque d'implication de l'utilisateur	12,8
2. Exigences et spécifications incomplètes	12,3
3. Changements aux exigences et spécifications	11,8
4. Manque de soutien de la direction	7,5
5. Incompétence technologique	7,0
6. Manque de ressources	3,4
7. Attentes irréalistes	5,9
8. Objectifs mal définis	5,3
9. Calendrier irréaliste	4,3
10. Nouvelle technologie	3,7
Autres	23,0

La gestion des exigences est largement reconnue pour être la partie la plus cruciale du développement logiciel. En effet, le succès du développement d'un programme logiciel peut largement dépendre de la qualité de cette activité. L'étude faite par le groupe Standish le démontre. Dans les sites de commerce électronique, on a découvert que les utilisateurs ne font un achat que dans 25 à 42 % des cas, parce qu'ils sont incapables de trouver le produit à acheter dans un temps raisonnable. Phil Terry de *Creative Good* a aussi estimé que, durant l'années 2000, il y a eu 19 G\$ de pertes de ventes aux États-Unis à cause de problèmes d'utilisabilité des sites de commerce électronique (Maguire, 2001).

Selon Emam et Birk (2000), d'autres études indiquent que 80 % des projets de développement de système d'information de gestion courent des risques de dérapage des exigences-utilisateurs ; c'est aussi le cas pour 70% des projets militaires. Selon les mêmes auteurs, une étude européenne récente, portant sur des projets de développement logiciel, indique, que dans plus de 40% des cas, ils ont des difficultés majeures de gestion des exigences client, et dans plus de 50% des cas, ils ont de difficultés majeures dans le champ de la spécification des exigences.

Une recherche, qui comportait un questionnaire détaillé adressé à 38 membres de la *British Computer Society*, de l'*Association of Project Managers* et de l'*Institute of Management*, a produit les résultats suivants. Sur les 1 027 projets étudiés, seulement 130, soit 12,7 % ont été réussis (Taylor, 2000). Des 130 projets réussis, seuls 2,3 % d'entre eux étaient des projets de développement, 18,2 % étaient des projets de maintenance et 79,5 % étaient des projets de conversion alors que les projets de développement constituaient la moitié des 1 027 (Taylor, 2000). En d'autres termes, les projets de développement connaissent d'importants problèmes. La définition des exigences, la gestion des demandes de changement et la gestion de la portée du projet sont les principales causes de l'échec d'un projet d'après cette recherche. Ces études rejoignent les conclusions de Landauer (1995) sur la gestion des exigences et le rôle de l'utilisateur dans le processus de développement.

Le développement des systèmes informatiques est encadré par un processus. Ce dernier définit un certain nombre d'activités et de biens livrables à produire. Le traitement réservé à l'utilisateur de ce système en développement dépend de la place qui lui est accordée dans le processus. C'est donc l'importance qu'on accorde à l'utilisateur dans le processus de développement qui sera étudiée dans la présente étude de cas.

Le cas est un projet de développement de système encadré par un processus de développement, le RUP®, où une intervention en conception centrée sur l'utilisateur, une intervention ergonomique, a été faite lors de l'étape de la définition des exigences. Le chapitre IV portant sur la méthode précise les objectifs de la recherche.

CHAPITRE I

DÉVELOPPEMENT ET CONCEPTION CENTRÉE SUR L'UTILISATEUR

Pour répondre aux problèmes de construction des systèmes informatiques, on a défini des processus qui fixent les étapes et les moyens utilisés pour le développement. Les premières étapes consistent à identifier les besoins des parties prenantes, parmi lesquels on compte, pour les systèmes interactifs, les besoins des utilisateurs. Ces besoins sont traduits en exigences qui déterminent les fonctionnalités et les contraintes du système. La définition des interfaces-utilisateurs fait partie de ces exigences. L'identification des exigences qui sont inspirées des utilisateurs et de leur organisation, ainsi que la définition des interfaces-utilisateurs font l'objet d'un processus particulier qu'on désigne sous le vocable de conception centrée sur l'utilisateur.

Le *Rational Unified Process* (RUP©) est un processus commercial de développement de système informatique qui prescrit, entre autres, une démarche pour définir les exigences et pour concevoir les interfaces-utilisateurs. À cause de l'importance de sa diffusion, le RUP© est devenu *de facto* une norme (Gulliksen, Göransson et Lif, 2001). Le présent chapitre décrit ces processus de développement et plus particulièrement l'étape de définition des exigences. Il décrit la conception centrée sur l'utilisateur et plus précisément les principaux aspects de la norme ISO 13407. Enfin, il brosse un portrait général du RUP© et de la place qu'il accorde à la conception des interfaces-utilisateurs.

1.1 Le développement d'un système informatique

En 1968, à l'occasion d'une conférence parrainée par le Comité des sciences de l'Organisation du traité de l'Atlantique Nord (OTAN) qui portait sur l'utilisation des ordinateurs, la notion de génie logiciel était officialisée. On reconnaissait la nécessité d'organiser le développement des systèmes informatiques.

À l'automne 1967, le Comité des sciences de l'OTAN mandatait un groupe de travail ayant pour tâche d'évaluer tout le champ des sciences informatiques. À la fin de l'année, ce groupe de travail recommandait la tenue d'une conférence sur le génie logiciel. Le terme « génie logiciel » était délibérément choisi pour inclure désormais les fondements théoriques et les disciplines appliquées, traditionnelles aux différentes branches du génie. Une cinquantaine d'experts

provenant de différentes sphères du monde informatique furent invités à donner leur avis. La conférence s'est tenue à Bruxelles en mars 1968 (OTAN, 1969).

Lors de la conférence, on reconnaît la nécessité pour le génie logiciel de fixer des étapes à la production de système informatique. Ces étapes doivent se composer d'activités. L'ensemble de ces étapes et activités est défini par une méthodologie qui comprend la connaissance et la compréhension de ce que doit être le programme ainsi que l'ensemble des méthodes, procédures et techniques utilisées pour son développement (OTAN, 1969).

Différents modèles décrivent ces étapes. L'un des premiers et des plus prédominants du génie logiciel a été le modèle en cascade. Le développement logiciel y est organisé en une série de phases modulaires dont on peut en voir une représentation à la figure 1.1. Chacune des phases fait l'objet d'une production de documents qui tiennent lieu de spécification pour la phase suivante (Rosson et Carroll, 2002). C'est au début des années 1970 que Winston Royce a défini ce « modèle en cascade » (Leffingwell et Widrig, 2000).

Le modèle en cascades a été une réussite quant au renforcement du rôle des exigences. Cependant, sa force est devenue une source de difficultés lorsque l'élaboration des documents d'exigences et de conception est devenue telle qu'elle empêchait souvent de terminer chaque phase. Aussi, cette mauvaise application a fini par en faire une approche rigide du développement, où les exigences sont gelées pour toute la vie du projet, où le changement est un anathème et où le processus de développement, lui-même, supprime le reste (Leffingwell et Widrig, 2000).

Barry Boehm (1988) propose le modèle en spirale. Dans ce modèle, le développement consiste à produire des prototypes définis selon les risques identifiés du projet. Par la suite, un processus structuré semblable au modèle en cascade est utilisé pour compléter le développement du système final.

En 1995, Philippe Kruchten propose l'approche itérative. Selon cette approche, le projet connaît plusieurs itérations lors de chacune des phases du développement. Une itération est une séquence d'activités selon un calendrier et des critères d'évaluation qui se concrétise par un exécutable quelconque. Chaque itération se construit à partir de la fonctionnalité de l'itération précédente. Le projet est ainsi développé de façon itérative et incrémentale (Leffingwell et Widrig, 2000).

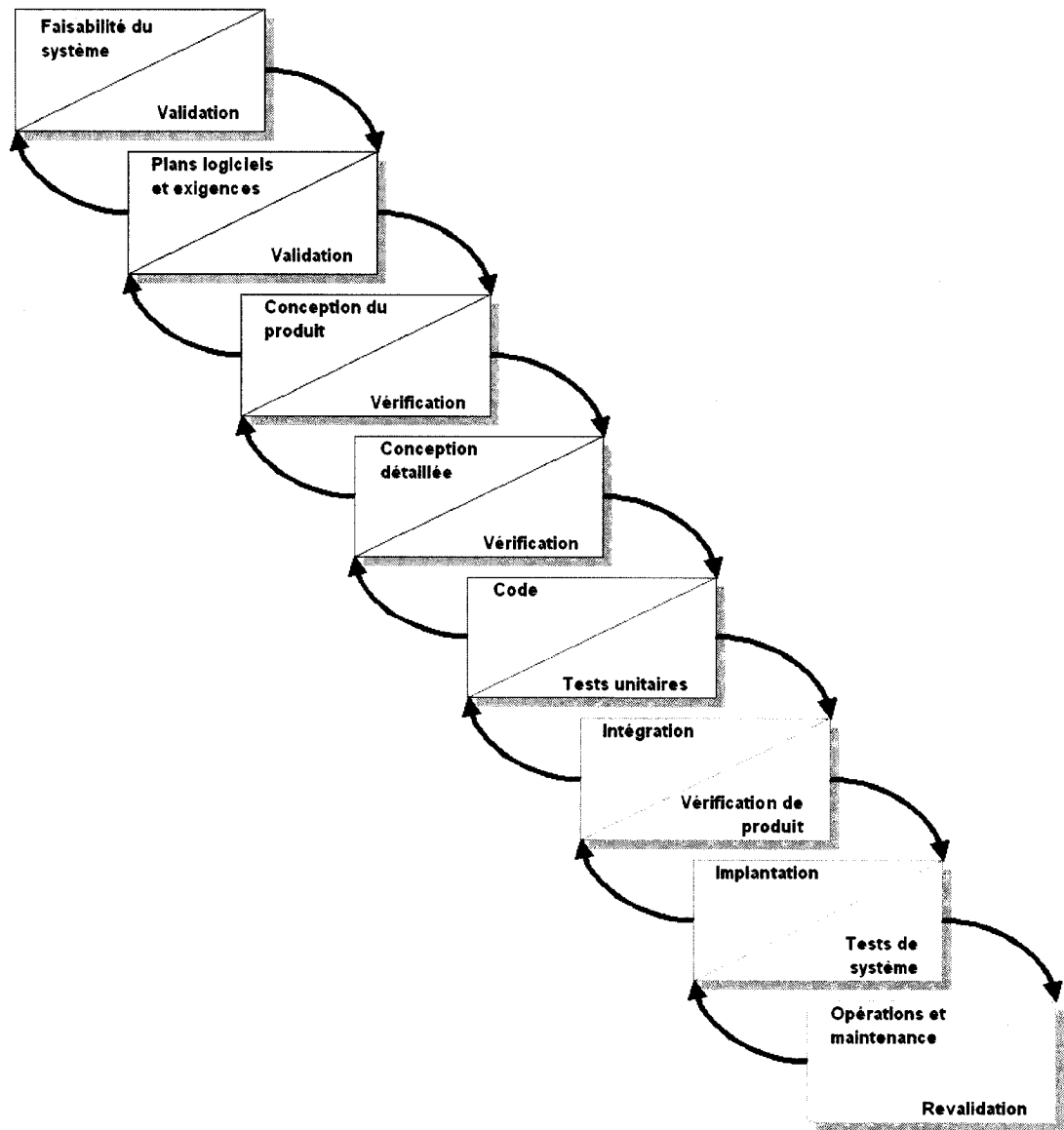


Figure 1.1 : Le modèle en cascade (Boehm, 1988)

Bien que l'organisation de ces phases puisse différer selon les modèles, les étapes, comme la conception ou l'intégration du système, demeurent les mêmes. Et tous ces modèles comprennent une étape de définition des exigences.

1.1.1 La gestion des exigences, principale difficulté

La recherche dans l'industrie démontre que les erreurs d'exigence sont les plus communes et leurs conséquences, les défauts qui en découlent, sont les plus coûteuses lors du travail technique (Young, 2001). Avec de bonnes exigences, 80 % des problèmes de reconstruction peuvent être éliminés.

Les coûts de reconstruction sont estimés à 45 % des coûts totaux du projet. Ainsi, 80 % de 45 % constituent 36 %, soit plus du tiers des coûts totaux du projet, selon les données de l'industrie. Ils peuvent être évités en éliminant les erreurs d'exigences (Young, 2001). La figure 1.2 donne un aperçu de la différence de coûts que représente leur correction selon l'avancement du développement. On y constate que, par exemple, pour une erreur d'exigence qu'on souhaiterait corriger lors des opérations, la correction coûterait au moins 40 fois ce qu'il en aurait coûté si elle avait été faite à l'étape de la définition des exigences. La correction pourrait être jusqu'à 1 000 fois plus onéreuse selon l'étude de Young (2001).

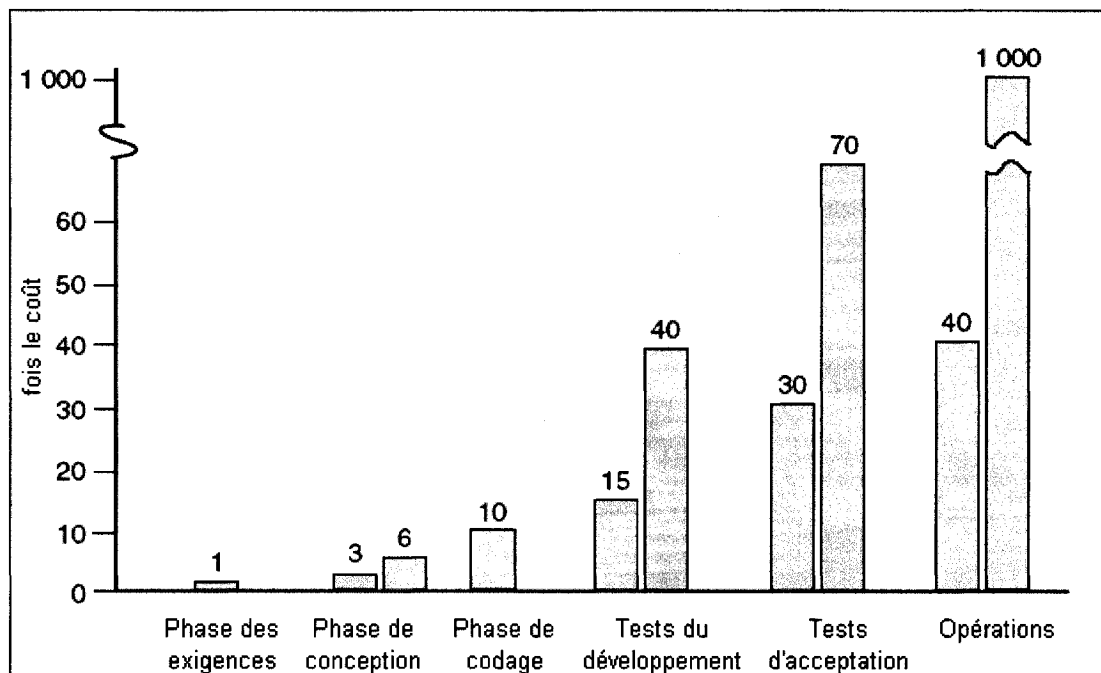


Figure 1.2 : Coûts relatifs de correction des exigences selon le stade où elles sont réalisées (Young, 2001).

Le développement d'un système logiciel interactif consiste d'abord à identifier les besoins du

client et de l'utilisateur. Une spécification, le premier résultat de l'ingénierie des exigences, est un énoncé concis des exigences que le logiciel doit satisfaire. Ce sont les conditions ou capacités qui permettent à l'utilisateur d'atteindre un objectif ou celles que le système possède pour satisfaire un contrat ou un standard. Idéalement, la spécification permet aux parties prenantes d'apprendre rapidement ce qu'est le logiciel et aux développeurs de comprendre exactement ce que veulent les parties prenantes (Hofman et Lehner, 2001).

La démarche de définition des exigences est définie par des normes élaborées, entre autres, par l'*Institute of Electrical and Electronics Engineers (IEEE)*. Les trois principales sont :

1. *IEEE guide for information technology - system definition - Concept of Operations (ConOps) document* (1998) ou Concept des opérations dont l'objectif est de décrire les caractéristiques d'un système du point de vue de l'utilisateur. Il sert à communiquer les caractéristiques du système à l'utilisateur, à l'acheteur, au développeur et aux autres parties prenantes de l'organisation. Il décrit l'organisation à laquelle appartient l'utilisateur, sa mission et ses objectifs selon le point de vue du système.
2. *IEEE guide for developing system requirements specifications* (1998) ou Spécification d'exigences-systèmes (SES) vise à guider le développement de l'ensemble des exigences. Cela suppose l'identification, l'organisation, la présentation et la modification des exigences. Elle détermine aussi les conditions d'incorporation de concepts opérationnels, des contraintes de conception et des exigences de configuration de la conception. Elle couvre enfin les qualités et caractéristiques de chacune des exigences et de l'ensemble des exigences.
3. *IEEE recommended practice for software requirements specifications* (1998) ou Spécification des exigences logicielles (SEL) définit l'approche pour déterminer la spécification des exigences logicielles. Elle décrit le processus pour créer cette spécification.

D'autre part, on peut catégoriser les exigences logicielles ainsi :

- les exigences fonctionnelles, qui spécifient les fonctions qu'un système ou un composant de système doivent être capables d'exécuter ;

- les exigences de performance, qui spécifient les caractéristiques de performance qu'un système ou un composant de système peuvent avoir comme la vitesse, l'exactitude et la fréquence ;
- les exigences d'interfaces externes, qui spécifient les éléments matériels, logiciels ou de base de données avec lesquels le système ou un composant doit s'interfacer ou fixer les contraintes sur les formats, le synchronisme ou tous autres facteurs imputables à une interface ;
- les contraintes de conception d'un système logiciel ou d'un composant de système logiciel comme, par exemple, les exigences de langage ou les exigences physiques de matériel, les normes de développement logiciel ou d'assurance qualité ;
- les attributs de qualité qui spécifient dans quelle mesure le logiciel doit posséder des attributs qui affectent sa qualité tels que la rectitude, la fiabilité, la maintenabilité et la portabilité (Thayer, 2002).

Les interfaces externes comprennent les interfaces-utilisateurs. L'interface-utilisateur est la partie visible et accessible à l'utilisateur à partir de laquelle il peut utiliser le logiciel. Il existe aussi l'équivalent de la notion d'exigence d'utilisabilité qu'on peut inclure dans la description des interfaces-utilisateurs de la SEL. Elles définissent les buts d'utilisabilité et les mesures qui lui sont associées pour un produit particulier. Les exigences d'utilisabilité sont liées aux autres exigences, comme les types d'utilisateurs qui doivent interagir avec le produit (Preece, Sharp et Rogers, 2002). La description des types d'utilisateurs fait partie de la spécification des exigences logicielles. La SEL propose une organisation des exigences fonctionnelles par classe d'utilisateur.

Le *ConOps* décrit les opérations du système, automatisé ou manuel, existant. Il prévoit la publication de scénarios d'utilisation, appelés scénarios opérationnels. La SES reprend ce type de scénarios pour décrire le nouveau système. Le *ConOps*, la SES et la SEL comprennent notamment la description des interfaces-utilisateurs.

La norme ISO/IEC 12207 : 1994, intitulée *Information Technology Software Life Cycle Processes*, prévoit que la définition des interfaces-utilisateurs ainsi que les spécifications de ce qu'elle appelle l'ingénierie des facteurs humains se fassent à l'étape de l'analyse des exigences du cycle de vie du logiciel (ISO/IEC, 1995).

1.1.2 La spécification des exigences

Une des premières étapes du développement d'un système informatique, une étape cruciale, est la spécification des exigences. Dans le cas d'un système informatique interactif, ces exigences seront définies en bonne partie selon les besoins du client et des utilisateurs de ce système. Il faut investir entre 8 % et 14 % du coût total d'un programme dans le processus de définition des exigences. Il faut consacrer du temps et un effort supplémentaire lors du début du projet pour identifier les exigences réelles. Il faut s'assurer d'une responsabilité conjointe de l'utilisateur et du fournisseur face aux exigences (Young, 2001).

L'analyse des exigences, en génie logiciel, aborde le domaine du problème, plus précisément sa structure, incluant la structure logique des objets dans le monde réel. L'objectif de l'analyse des exigences est de fournir une spécification non ambiguë et compréhensible avec laquelle le système peut être conçu et construit (Wasserman, 1996). La conception, la phase suivante du développement, au contraire, traite de la façon dont la structure du système implémente une solution du problème (Wasserman, 1996).

Lors de cette analyse, il est nécessaire de formuler les exigences à un certain niveau de généralisation qu'on obtient par un exercice d'abstraction. Cet exercice permet de se concentrer sur un problème à un niveau de généralisation quelconque sans égard aux détails de bas niveau non pertinents. Il permet aussi de travailler avec des concepts et des termes du domaine du problème sans avoir à les transformer dans une structure qui lui est étrangère (Wasserman, 1996).

1.1.3 Les exigences-utilisateurs

La société IEEE Computer et l'*Association for Computing Machinery* ont développé conjointement un guide intitulé *Software Engineering Body of Knowledge* (SWEBOK). C'est un corps des connaissances du génie logiciel qui représente un large consensus sur le contenu de la discipline. Il constitue une référence (Abran et al., 2004). Les exigences logicielles et les exigences-utilisateurs sont définies comme suit, selon le SWEBOK. Les exigences logicielles expriment les besoins et contraintes qui affectent un produit logiciel contribuant à la satisfaction d'une quelconque application du monde réel (Sawyer et Kotonya, 2001). Par ailleurs, parmi les exigences logicielles, on peut définir le terme exigence utilisateur comme celles exprimées par les clients ou les utilisateurs finaux (Sawyer et Kotonya, 2001).

La définition et l'élaboration des exigences sont appelées explicitation d'exigence. En pratique, l'ingénieur des exigences doit systématiquement extraire et inventorier les exigences à partir des demandes des parties prenantes, de l'environnement du système, d'études de faisabilité, d'analyse de marché, de plans d'affaire, d'analyse de produits concurrents et d'une connaissance du domaine (Sawyer et Kotonya, 2001).

Ce travail ne se fait pas sans difficulté. Les problèmes majeurs surviennent habituellement d'un manque de connaissance du domaine d'application et de difficultés de communication entre l'utilisateur et le concepteur (Pimenta et Faust, 1997). Un des problèmes les plus frustrants et dominants de toutes les applications est ce qu'il est convenu d'appeler le syndrome « Oui, mais... » constaté à l'observation de la réaction de l'utilisateur à chaque partie de logiciel développée. On observe toujours deux réactions immédiates et distinctes lorsque l'utilisateur voit l'implantation du système pour la première fois :

- Bravo, c'est bien, on peut vraiment l'utiliser, c'est un excellent travail !
- Oui, mais..., maintenant que je le vois, qu'arrive-t-il avec ceci ? Est-ce qu'il ne serait pas bien que... ? (Leffingwell et Widrig, 2000).

Un autre syndrome émerge de la communication : la distance entre l'utilisateur et le développeur. On l'appelle le syndrome « de l'utilisateur et du développeur ». Il est imputable au fait que les utilisateurs et les développeurs proviennent de mondes différents, qu'ils parlent des langages différents et qu'ils ont des formations, des motivations et des objectifs différents (Leffingwell et Widrig, 2000).

1.2 Les interfaces-utilisateurs

La conception actuelle de l'interface-utilisateur, telle que prévue par les processus de développement de système informatique classique, est une phase de développement qui est par nature non structurée. Il n'est pas rare, dans la pratique, que peu ou pas d'attention ne soit portée à cette phase du développement. Il est difficile, semble-t-il, de séparer le travail créatif de la conception de celui de la programmation. La conception d'interface-utilisateur est trop souvent le résultat d'une tâche de programmation, parmi d'autres. Il n'y a alors personne qui assume cette responsabilité de la conception des interfaces-utilisateurs (Gulliksen, Göransson et Lif, 2001).

L'implémentation des interfaces-utilisateurs compte habituellement pour 40 % à 50 % du montant total des lignes de code dans une application, mais le temps qui est consacré à cette partie du code compte pour beaucoup moins que 40 % à 50 % (Gulliksen, Göransson et Lif, 2001). Or, les incohérences, mêmes mineures, qui parsèment un logiciel, préoccupent les utilisateurs, détournant ainsi leur attention qui devrait être consacrée à la tâche qu'ils ont à exécuter (Johnson, 2000). Ces incohérences se manifestent par le comportement du système et par l'organisation des interfaces-utilisateurs.

C'est dans ce contexte, que l'adoption d'une approche centrée sur l'utilisateur pour la conception d'un système peut réduire le temps de développement et de reprise d'une nouvelle version. Elle peut améliorer la productivité des utilisateurs et réduire la formation, la documentation et les coûts d'assistance technique (Earthy, Sherwood-Jones et Bevan, 2001).

1.3 La conception centrée sur l'utilisateur

Lorsque les utilisateurs finaux sont devenus plus nombreux et moins qualifiés, les systèmes interactifs ont commencé à être évalués et comparés selon leur utilisabilité : leur facilité d'apprentissage, leur facilité d'utilisation et la satisfaction de l'utilisateur. Les raisons qui rendent un système plus ou moins utilisable sont complexes et sont l'objet d'une recherche considérable. On peut identifier trois perspectives: la performance humaine, l'apprentissage et la cognition, et l'activité coopérative (Rosson et Carroll, 2002).

L'approche de la conception centrée sur l'utilisateur est celle dont la première intention est la considération des intérêts ou des besoins des individus ou des groupes qui vont travailler avec un système ou utiliser son extrant (Earthy, 2000).

1.3.1 La norme ISO 13407

L'Organisation internationale de normalisation (ISO) a publié la norme ISO 13407, « Processus de conception centrée sur l'opérateur humain pour les systèmes interactifs », qui définit le processus de conception centrée sur l'utilisateur. Selon ISO, la norme ne prétend pas couvrir un cycle de développement de système ; elle est complémentaire aux méthodes existantes et elle doit s'y intégrer selon le contexte (ISO, 1999).

ISO 13407 énonce quatre grands principes :

- la participation active des utilisateurs et une compréhension claire des exigences liées à

l'utilisateur et à la tâche ;

- une allocation appropriée des fonctions entre les utilisateurs et la technologie ;
- l'itération des solutions de conception ;
- une conception pluridisciplinaire (ISO, 1999).

Et elle propose des activités :

- comprendre et spécifier le contexte d'utilisation ;
- spécifier les exigences liées à l'utilisateur et à l'organisation ;
- proposer des solutions de conception ;
- évaluer les conceptions par rapport aux exigences (ISO, 1999).

La figure 1.3 illustre le processus et ses activités.

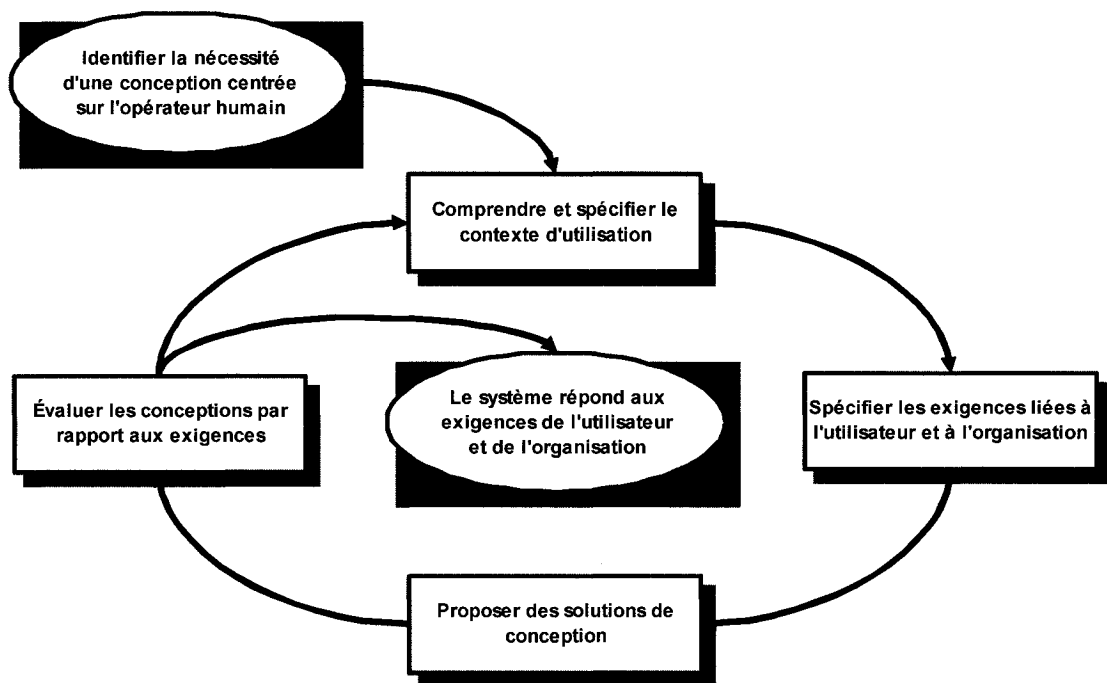


Figure 1.3 : Cycle des activités de conception centrée sur l'opérateur humain (ISO, 1999).

Pour la norme, à l'activité de définition des exigences fonctionnelles et autres prévues par les processus de conception, il faut ajouter la définition explicite des exigences liées à l'utilisateur et à l'organisation à partir du contexte d'utilisation. Cela comprend, outre la spécification des

exigences liées à l'utilisateur et à l'organisation, l'évaluation de l'impact du système sur les utilisateurs, un énoncé d'objectifs de conception centré sur l'utilisateur, les exigences d'ordre juridique et des références mesurables pour tester l'utilisabilité du système (ISO, 1999, ISO, 2000). La norme prévoit des itérations qui entraînent l'obligation de tester le système. Cela a pour conséquence de valider les solutions proposées et de tester les exigences.

La conception, qui doit être pluridisciplinaire, commande la participation de différentes parties prenantes aux qualifications variées. La norme en fait une énumération. Elle distingue, notamment, le concepteur d'interface de l'ergonome.

1.3.2 Le contexte d'utilisation

Un système ou un produit est développé pour être utilisé dans un certain contexte. Il peut être utilisé dans un environnement par des utilisateurs qui ont certaines caractéristiques et qui souhaitent atteindre certains objectifs. Le système est utilisé dans un environnement technique, physique, social et organisationnel qui affecte son utilisation (Maguire, 2001).

L'analyse du contexte est une méthode structurée d'explicitation d'une information détaillée sur le contexte d'utilisation comme base des futures activités d'utilisabilité, particulièrement la spécification et l'évaluation des exigences-utilisateurs (Maguire, 2001). La figure 1.4 situe le contexte d'utilisation dans la perspective de spécification de l'utilisabilité.

Comprendre les utilisateurs nécessite des recherches empiriques afin de connaître les caractéristiques pertinentes des utilisateurs potentiels (Johnson, 2000). La description doit être la plus exacte possible. Il faut décrire, par exemple, le nombre d'utilisateurs qui utiliseront le système, leurs qualifications, leur expérience et ainsi de suite.

L'analyse de la tâche, incluse dans l'analyse du contexte d'utilisation, peut être définie comme l'étude de ce que l'utilisateur doit faire en termes d'actions ou de processus cognitifs pour accomplir une tâche (Maguire, 2001). L'analyse de la tâche révèle des informations déterminantes pour le bon fonctionnement du système. Avec cette méthode, on observe l'exécution des tâches dans l'environnement de travail normal (Rosson et Carroll, 2002). L'analyse de la tâche est utilisée pour investiguer une situation existante et non pas pour envisager de nouveaux systèmes ou appareils. Elle est utilisée pour analyser la justification et l'intention des personnes qui agissent : ce qu'elles tentent d'atteindre, pourquoi elles tentent de l'atteindre et comment elles vont l'atteindre. L'information glanée à partir de l'analyse de la tâche

établit une base sur laquelle on construit de nouvelles exigences ou on conçoit de nouvelles tâches (Preece, Rogers et Sharp, 2002).

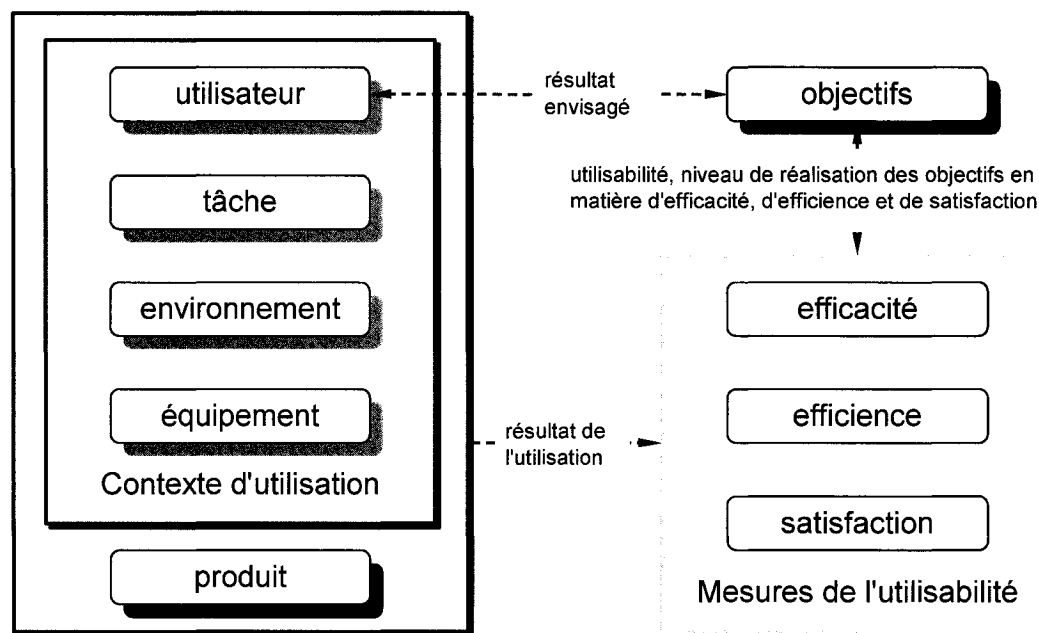


Figure 1.4 : Structure d'utilisabilité (ISO, 1998).

Les ingénieurs de l'utilisabilité participent à l'analyse des exigences en étudiant l'exécution du travail pour évaluer les problèmes ou les opportunités que peut offrir la nouvelle technologie (Rosson et Carroll, 2002). Il y a des tâches requises qui ne sont pas nécessaires et qui peuvent tout simplement être ignorées, libérant ainsi le logiciel pour l'exécution de tâches nécessaires (Johnson, 2000).

La production de scénarios qui impliquent un utilisateur, est intimement liée à l'analyse de la tâche. Normalement, elle devrait inspirer la rédaction du scénario. L'analyse de la tâche révèle des scénarios d'utilisation typiques et critiques : ce que les utilisateurs veulent habituellement obtenir ainsi que les événements les plus notoires de l'interaction des utilisateurs, tant ce qui est génial que les erreurs (Carroll et Rosson, 1992).

L'implantation d'un nouveau système suppose que la répartition des tâches entre le nouveau système et l'utilisateur sera différente. Parfois, une réorganisation du travail modifie cette

répartition entre les utilisateurs. Certaines tâches disparaîtront. Mais cette analyse de la tâche demeure toujours indispensable. Les scénarios, produits suite à l'analyse de la tâche, combinés aux exigences du nouveau système permettent de produire des scénarios d'utilisation qui pourront être utilisés pour les tests d'utilisabilité. L'analyse de la tâche permet aussi d'identifier les priorités de réalisation du nouveau système et la criticité des fonctions.

1.3.3 Le prototypage et le maquetage

Les solutions de conception qu'évoque ISO 13407 peuvent prendre différentes formes, comme les maquettes papier fidèles aux écrans finaux, les prototypes logiciels jetables plus ou moins fidèles à la solution finale ou les premières itérations exécutables du système.

Ainsi, les maquettes sont évaluées et modifiées avec des techniques d'évaluation comme des tests d'utilisabilité formels durant lesquels des représentants des utilisateurs exécutent des tâches représentatives avec une formation et une intervention minimales, en utilisant la maquette comme une interface-utilisateur réelle du produit (Mayhew, 1999).

Les améliorations au processus de développement logiciel et à la qualité d'un système ne peuvent pas être évaluées sans un effort de mesures efficaces (Wasserman, 1996). Dès le début du processus de développement et durant le processus, les utilisateurs concernés doivent effectuer un réel travail d'utilisation des premières versions du matériel de formation et des manuels, des simulations et des prototypes d'interfaces-utilisateurs, et ainsi de suite. L'importance est ici accordée aux mesures formelles et informelles (Gould, Boies et Ukelson, 1997).

1.3.4 Les tests d'utilisabilité

ISO 13407 insiste sur la nécessité de l'évaluation des solutions à l'étape de ce que le RUP© appelle l'élaboration, lorsque les prototypes sont produits pour définir la conception détaillée (John, Bass et Adams, 2003). Une technique, comme celle des tests d'utilisabilité formels, est utilisée durant tout le développement du produit pour étendre l'évaluation aux sous-ensembles de fonctionnalités et aux catégories d'utilisateur non traitées et aussi pour continuer de raffiner l'interface-utilisateur et la valider par rapport aux objectifs d'utilisabilité (Mayhew, 1999). ISO 13407 (1999) recommande de « présenter les solutions de conception aux utilisateurs et de leur donner la possibilité d'accomplir des tâches ou des simulations de tâches ».

Lemeunier (2000) décrit une technique de test d'utilisabilité typique de la conception centrée sur l'utilisateur : « Par exemple, le protocole du type Magicien d'Oz est une méthode d'essai bien connue. Il permet d'obtenir des corpus de dialogues ou d'interactions entre une machine et l'utilisateur. Plus précisément, il s'agit de faire simuler les comportements de la machine par un être humain. Ce dernier simule donc un dialogue avec une machine. Sans l'existence de système de dialogue opérationnel, cette méthode est nécessaire, d'une part pour étudier les spécificités des interactions humain-machine, bien qu'il soit clair qu'elles n'en sont pas vraiment, et, d'autre part, pour amorcer le développement d'un système informatique qui prendra la place du compère humain. »

1.3.5 Une approche itérative

La conception itérative consiste à soumettre les solutions à des évaluations. Ces solutions peuvent être de simples maquettes d'écran ou des prototypes logiciels de plus haute fidélité. Lors de certaines de ces évaluations, l'utilisateur tente d'accomplir des tâches réelles en utilisant le prototype. C'est la rétroaction obtenue de cet exercice qui est utilisée pour développer plus avant la conception (Maguire, 2001). ISO 13407 prescrit que le prototypage, les simulations, la modélisation et les maquettes soient améliorées par les tests avec les utilisateurs à différentes étapes de la conception (John, Bass et Adams, 2003).

L'itération permet, suite aux tests effectués avec des scénarios de situation réelle, d'améliorer les solutions de conception avec les informations recueillies. Conjuguée à l'implication des utilisateurs, elle minimise les risques que le système ne réponde pas aux exigences liées à l'utilisateur et à l'organisation, notamment les exigences non apparentes ou non explicites (ISO, 1999).

Le nombre d'itérations peut varier selon différents facteurs. Ce sont les objectifs d'utilisabilité qui fixent la fin du cycle itératif. Un bon travail d'analyse des exigences rapproche de l'interface finale à la première itération de la conception, accroissant ainsi l'atteinte des objectifs d'utilisabilité et réduisant le nombre d'itérations de conception et d'évaluation (Mayhew, 1999). Dans d'autres cas, lorsque l'interface est complexe ou, en début de projet, le nombre d'itérations peut être plus élevé.

1.3.6 Une conception pluridisciplinaire

Le développement d'un système centré sur l'utilisateur est un processus coopératif qui profite de l'implication active des différentes parties prenantes, chacune d'elle ayant sa vision et son expertise à partager. L'équipe de développement doit être constituée d'experts ayant des qualifications techniques appropriées et de ceux qui ont un intérêt dans le logiciel proposé. L'équipe doit donc inclure des gestionnaires, des spécialistes de l'utilisabilité, des utilisateurs finaux, des ingénieurs logiciel, des concepteurs graphiques, des concepteurs de l'interaction, du personnel responsable de la formation et du soutien technique ainsi que des experts de la tâche (Maguire, 2001).

1.4 Le *Rational Unified Process*

Les approches de développement en cascade, en spirale et itérative proposent une méthodologie pour le développement d'un système informatique. Il existe des traductions commerciales de ces modèles qui sont utilisés par les organisations qui développent des logiciels. Le *Rational Unified Process* est un de ces processus commerciaux.

Ivar Jacobson, un des chefs de file de la conception orientée objet, a développé *Objectory*, qui est une contraction de *Object* et de *Factory*, après son départ de la compagnie Ericsson où il était employé. *Objectory* est un processus de développement de produit. C'est ainsi que la notion de « cas d'utilisation » a été introduite officiellement lors d'une conférence en 1987.

En 1995, la compagnie Rational a acquis le processus *Objectory*. Ivar Jacobson rejoignait ainsi Grady Booch et James Rumbaugh avec lesquels il a développé le *Unified Modeling Language* (UML) dont la version 1.1 fut reconnue comme une norme par le *Object Management Group* en 1997. L'UML est un langage de modélisation par objets utilisé pour présenter les composants d'un système informatique. En juin 1998, le *Rational Objectory Process* devient le *Rational Unified Process* (Jacobson, Booch et Rumbaugh, 1999). Philippe Kruchten était et demeure le directeur du RUP© chez IBM© qui a acquis la compagnie Rational© en décembre 2002.

1.4.1 Un processus centré sur l'architecture

Selon Kruchten (2001), le RUP© est un processus centré sur l'architecture qu'il décrit en donnant « la version du *Rational Unified Process* :

- 1) L'architecture est l'ensemble des décisions significatives quant à :

- l'organisation d'un système informatique ;
- le choix des éléments structuraux qui composent le système, la définition de leurs interfaces et de leur comportement, spécifié par la collaboration entre ces éléments ;
- la composition de ces éléments en sous-systèmes de plus en plus importants ;
- le style architectural qui guide cette organisation, le choix des éléments, la définition de leurs interfaces et de leur collaboration, et leur composition.

2) L'architecture logicielle s'intéresse non seulement à la structure et au comportement du système, mais aussi à son contexte : usage, fonctionnalité, performance, souplesse, réutilisation, compréhensibilité, esthétique, contraintes et compromis économiques et technologiques. »

En incluant des aspects comme l'usage et l'esthétique, la notion d'architecture prend pour le RUP© un sens très large. Cela confère aux travailleurs responsables de l'architecture des prérogatives très étendues sur la définition du système.

1.4.2 Un processus piloté par les cas d'utilisation

Le RUP© est un processus piloté par les cas d'utilisation. Jacobson, Booch et Rumbaugh (1999) présentent le principe du processus centré sur l'architecture après celui du processus piloté par les cas d'utilisation. Kruchten (2001) fait l'inverse. Nonobstant cette nuance, le RUP© s'approche de la conception par scénarios qui est populaire en conception centrée sur l'utilisateur. Cette préoccupation de rapprocher l'utilisateur du système n'est pas sans laisser croire que le RUP© pourrait effectivement être un processus de conception centrée sur l'utilisateur. Voici ce que sont les cas d'utilisation.

« Un cas d'utilisation est une séquence d'actions que réalise un système et qui fournit un résultat observable ayant une valeur pour un acteur particulier. Un acteur est quelqu'un ou quelque chose se situant à l'extérieur du système et qui interagit avec lui. » (Kruchten, 2001). Le modèle des cas d'utilisation, qui est la représentation de l'ensemble des cas d'utilisation d'un système, est un moyen de communiquer des exigences entre les clients et les développeurs et un moyen de structurer un modèle objet afin qu'il soit gérable (Jacobson, 1994).

Dans le RUP©, les exigences-utilisateurs sont, en bonne partie, décrites par les cas d'utilisation. Ceux-ci décrivent tous les types d'utilisation du système (John, Bass et Adams, 2003). Un cas

d'utilisation est une description générique d'une entière transaction impliquant plusieurs objets du système (Elkoutbi, Khriss et Keller, 1999). Il est une généralisation qui tente de récupérer la nature de toute une catégorie d'activités ayant un lien entre elles (Mayhew, 1999) et il décrit une tâche de haut niveau qui peut être reliée à un but de haut niveau des acteurs qui y participent (Homrighausen, Six et Winter, 2002).

Un cas d'utilisation définit un contrat portant sur le comportement d'un système entre les parties prenantes. Le cas d'utilisation décrit le comportement d'un système sous diverses conditions lorsqu'il répond à des requêtes d'une des parties prenantes, appelée acteur primaire. L'acteur primaire enclenche une interaction avec le système pour atteindre un objectif. Le système répond en protégeant les intérêts de toutes les parties prenantes. Les différentes séquences de comportement ou scénarios peuvent s'exécuter selon les particularités des requêtes et les conditions entourant les requêtes. Le cas d'utilisation rassemble ces différents scénarios (Cockburn, 2001).

Les acteurs sont des objets qui résident à l'extérieur du modèle du système. Les cas d'utilisation sont des objets qui résident à l'intérieur du système. Les acteurs interagissent avec le système. Ils représentent tout ce qui est nécessaire à l'échange d'information avec le système de telle sorte que rien d'autre à l'extérieur n'a d'impact sur le système. Les acteurs sont des êtres humains ou d'autres systèmes (Jacobson, 1995).

Il faut distinguer les acteurs des utilisateurs. Un utilisateur est une personne qui utilise le système alors qu'un acteur représente un rôle spécifique qu'un utilisateur, entre autres, peut jouer. Les acteurs sont des instances d'une classe. L'instance d'une classe acteur existe seulement lorsque l'utilisateur agit sur le système. Le même utilisateur peut alors agir comme instances de plusieurs acteurs différents (Jacobson, 1995).

Lorsque les cas d'utilisation documentent les processus d'affaire d'une organisation, le système sous étude est l'organisation elle-même. Les parties prenantes sont les propriétaires de l'entreprise, les clients, les vendeurs et les gouvernements. Les acteurs primaires incluent les clients de l'entreprise et peuvent être les fournisseurs. Par ailleurs, lorsque les cas d'utilisation reflètent les exigences de comportement pour une partie de logiciel, le système sous étude est le programme informatique. Les parties prenantes sont alors les personnes qui utilisent le programme, l'entreprise en est le propriétaire, le gouvernement et les autres programmes

informatiques. L'acteur, dit primaire, est l'utilisateur assis devant l'écran d'un ordinateur ou un autre système informatique (Cockburn, 2001).

La description d'un cas d'utilisation rend compte de ce qui se passe dans le système quand le cas d'utilisation se réalise. La fonctionnalité complète du système est définie par un ensemble de cas d'utilisation, chacun d'eux représentant un flot spécifique d'événements (Kruchten, 2001).

Les cas d'utilisation ressemblent aux scénarios d'utilisation tels que présentés par quelques auteurs du monde de l'orienté objet. Cependant, il y a plusieurs différences, tant syntaxiques que sémantiques. Les scénarios correspondent normalement à des instances de cas d'utilisation et ils n'ont pas d'équivalent de classes de cas d'utilisation. Les cas d'utilisation sont traités plus formellement et sont décrits comme un modèle en soi, autant que les interactions entre les objets de différents modèles objet. Les scénarios sont normalement décrits comme des interactions entre objets seulement (Jacobson, 1994).

1.4.3 L'utilisabilité et les cas d'utilisation

Selon Ivar Jacobson (1995), la modélisation des cas d'utilisation devrait être précédée par une activité de pré-analyse où les utilisateurs participent à la découverte des cas d'utilisation et des interfaces-utilisateurs associées. Il serait approprié de faire ce travail de pré-analyse lors d'une série d'ateliers de travail où les concepteurs d'interfaces-utilisateurs sont observateurs et les utilisateurs sont le centre d'intérêt. Les utilisateurs sont observés dans leur milieu de travail, ils sont interviewés et on leur demande de décrire d'une façon épisodique les différents scénarios d'utilisation qui sont des instances de cas d'utilisation. Pour assurer une meilleure compréhension des besoins de l'utilisateur, des maquettes de l'interface-utilisateur évoluent et, lorsqu'elles deviennent stables, pas avant, des prototypes peuvent être développés. Ce type de conception orientée-utilisation, les tests d'utilisabilité et autres seraient très importants à la compréhension (Jacobson, 1995).

Cependant, cette dernière approche de modélisation des cas d'utilisation qui s'alimentent des scénarios d'utilisation, inspirés par des spécialistes de l'utilisabilité, n'est pas reprise ou évoquée par le RUP®. Cet aspect est important. Ainsi définis, les cas d'utilisation deviennent des scénarios génériques qui décrivent le comportement du système de façon abstraite. Le scénario d'utilisation décrit lui, par contre, l'exécution d'une tâche précise faite par un utilisateur avec le système. Cette approche n'est que partiellement celles des spécialistes de l'utilisabilité qui ne

demandent pas « de décrire d'une façon épisodique les différents scénarios d'utilisation », mais qui font des analyses de la tâche, sans égard au système à construire, pour alimenter la description du contexte d'utilisation.

1.4.4 Les phases, les processus et les itérations

Le RUP© est un processus itératif et incrémental. À chaque phase, le projet connaît plusieurs itérations comme l'illustre la figure 1.5. Une itération est une séquence d'activités, fixée selon un calendrier établi et soumis à des critères d'évaluation, qui se concrétisent par une livraison quelconque. Chaque itération repose sur la fonctionnalité de l'itération précédente (Leffingwell et Widrig, 2000).

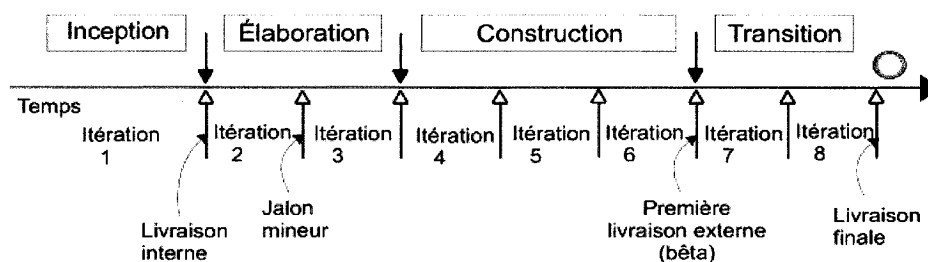


Figure 1.5 : Les itérations en RUP© (Kruchten, 2001).

Pour le RUP, la notion d'itération est une pierre d'angle du processus. Alors que Kruchten (2001) note avec pertinence que « les chefs de projet s'opposent souvent à l'idée de l'approche itérative, qu'ils considèrent comme un bricolage sans fin et incontrôlé », il faut souligner que le RUP© insiste constamment sur la notion d'itération et commande même que son utilisation soit planifiée.

La nature d'une itération varie selon le contexte où elle s'exécute. Mais elle comporte habituellement une version exécutable et fait l'objet d'une planification. Ainsi, une itération lors de la phase d'inception, comporte un prototype qui constitue une preuve de concept de la faisabilité du système. Lors de la phase d'élaboration une ou plusieurs itérations sont associées à la production d'un nombre équivalent de prototypes architecturaux (Kruchten, 2001, Rational, 2002). Plusieurs « patrons d'itération », incrémental ou évolutionniste, par exemple, sont proposés, mais ils sont applicables à l'ensemble du cycle de vie (Rational, 2002).

Dans l'approche itérative, les activités associées au développement du logiciel sont organisées selon un flot de travail. Chaque flot de travail consiste en un ensemble cohérent d'activités et chacun détermine comment les activités seront séquencées pour produire un document de travail viable, un exécutable, ou un artefact qui est un modèle, un élément de modèle ou un document.

L'approche itérative du RUP© consiste en quatre phases du cycle de vie : inception ou opportunité, élaboration, construction et transition telles que le représente la figure 1.6 :

1. L'inception porte sur la compréhension du domaine d'affaire du projet, la portée du projet et la faisabilité de son implantation. L'analyse du problème est effectuée. Un document « Vision » est créé. Les estimations préliminaires du calendrier et du budget sont faites. Les facteurs de risque du projet sont définis.
2. L'élaboration consiste à raffiner les exigences du système, établir l'architecture initiale d'un exécutable et développer un prototype pour démontrer la faisabilité.
3. La construction porte sur l'implémentation. La plus grande partie du codage est faite à cette étape. L'architecture et la conception sont complétées.
4. La transition comporte une version pour tests. Les utilisateurs et les responsables de la maintenance sont formés à l'utilisation de l'application. Lorsque l'application a répondu aux critères de références lors des tests, elle est envoyée à la communauté des utilisateurs et déployée pour utilisation (Leffingwell et Widrig, 2000).

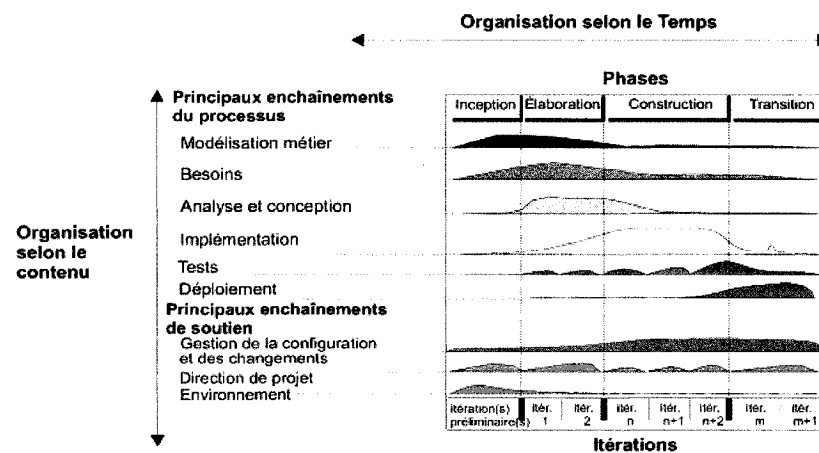


Figure 1.6 : Phases du cycle de vie selon le RUP© (Kruchten, 2001).

L'explicitation des exigences, la rédaction des cas d'utilisation et la spécification des interfaces-utilisateurs se fait donc lors de la phase d'élaboration. La phase d'élaboration correspond à l'étape de l'analyse prévue dans le modèle en cascades.

Enfin, les guides jouent un rôle important dans le RUP®. Ils illustrent les artefacts bien faits et insistent sur la qualité. Les directives sont des règles, recommandations ou heuristiques qui soutiennent les activités. Elles décrivent des techniques spécifiques, comme les transformations d'un artefact à un autre ou l'utilisation du langage UML. Elles améliorent aussi la qualité des artefacts sous la forme de listes de vérification associées aux artefacts ou utilisées dans les activités de révision (Kruchten, Ahlqvist et Bylund, 2001). Il y a des guides et des normes qui indiquent comment exécuter les activités de conception d'interfaces-utilisateurs et à quoi ces dernières doivent ressembler.

1.4.5 La conception des interfaces-utilisateurs

La conception des interfaces-utilisateurs, selon le RUP®, consiste à modéliser l'interface dans un scénarimage de cas d'utilisation et à concevoir un prototype d'interface-utilisateur. Les deux plus récentes versions du RUP® ont été publiées en 2002 et en 2003. La version 2002 incluait la conception de ce prototype utilisateur dans la discipline de la définition des exigences. La version 2003, par contre, a déplacé la conception du prototype d'interface-utilisateur dans la discipline Analyse et conception (Rational, 2003). La décision de déplacer l'activité dans cette discipline n'est pas expliquée. Cependant, l'approche de la version 2002 sera ici abordée, puisque c'est celle-là qui a été utilisée pour le cas étudié.

Selon le RUP®, l'ingénierie de l'utilisabilité est une feuille de route et la conception centrée sur l'utilisateur est un concept (Rational, 2002). Rational décrit une feuille de route comme l'adaptation du processus générique à des problèmes spécifiques. Des feuilles de route sont ainsi disponibles, par exemple, pour les petits projets ou les pratiques Agile dans le RUP® (Rational, 2002). Les concepts sont les travailleurs, les activités, les artefacts et les enchaînements d'activités, ainsi que d'autres éléments qui permettent de décrire le processus (Kruchten, 2001). La description de certains de ces concepts est incluse à l'intérieur des enchaînements d'activités appropriés comme l'illustre la figure 1.7.

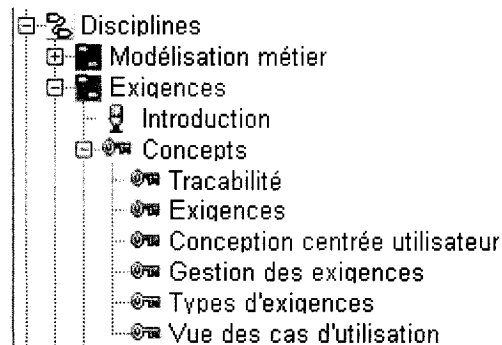


Figure 1.7 : Exemples de concepts (Rational, 2002).

1.4.5.1 Les activités de conception des interfaces-utilisateurs

Les activités de conception d'interface-utilisateur dans le RUP© font partie de l'enchaînement d'activités des exigences parce qu'elles sont étroitement associées aux artefacts d'exigences et plus particulièrement aux cas d'utilisation. De plus, les exigences sont souvent améliorées et raffinées grâce aux prototypes d'interface. Enfin, il est important de rappeler que puisque le RUP© est itératif, cet enchaînement est révisé à chaque itération, incluant les activités de développement de l'interface-utilisateur (Kruchten, Ahlqvist et Bylund, 2001).

Selon Kruchten (2001), « la conception de l'interface-utilisateur comprend deux étapes :

- la mise en forme graphique de l'interface-utilisateur opérée avec un souci de facilité d'utilisation ;
- la conception de l'interface-utilisateur en termes de classes de conception (et de composants logiciels tels que des classes ActiveX ou JavaBeans) qui conduit à son implémentation finale ».

« Dans l'enchaînement des activités de gestion des exigences qui apparaît à la figure 1.8, on effectue la première étape, en se basant sur le modèle des cas d'utilisation et les spécifications supplémentaires, même s'ils sont juste ébauchés. Il en résulte des définitions détaillées des caractéristiques des utilisateurs et des réalisations des parties des cas d'utilisation qui requièrent une interface avec les utilisateurs. En s'appuyant sur ces résultats, on construit un prototype de l'interface-utilisateur, le plus souvent à l'aide d'un outil de prototypage (Kruchten, 2001).

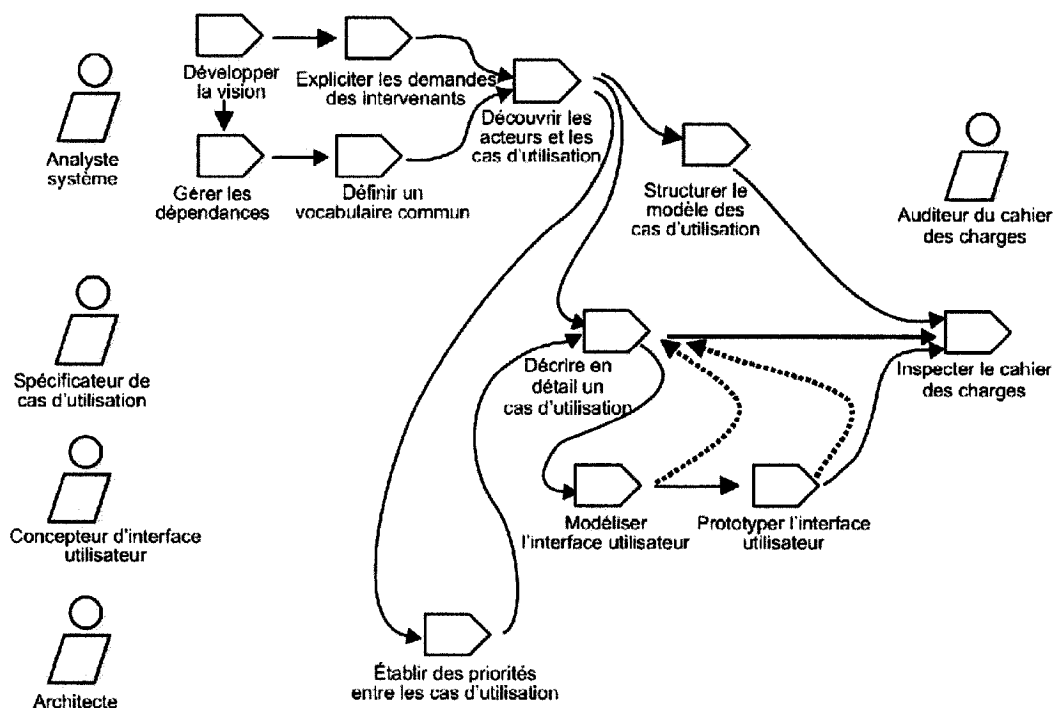


Figure 1.8 : Enchaînement des activités de gestion des exigences
(Kruchten, Ahlqvist et Bylund, 2001) et (Kruchten, 2001).

« Un concepteur d'interface-utilisateur complète, à l'aide de scénarimages, les cas d'utilisation détaillés et construit un prototype de l'interface-utilisateur qu'il soumet aux utilisateurs potentiels pour obtenir leurs réactions » (Kruchten, 2001).

Les propriétés les plus importantes d'un cas d'utilisation sont le flot principal des événements et les flot des événements alternatifs. Le scénarimage de cas d'utilisation comprend une version simplifiée du flot d'événements. Un scénarimage de cas d'utilisation est une description logique et conceptuelle de la façon dont un cas d'utilisation est traduit par une interface-utilisateur, incluant l'interaction entre les acteurs et le système. Il est représenté en UML par un diagramme de collaboration tel qu'illustré par la figure 1.9 (Kruchten, Ahlqvist et Bylund, 2001).

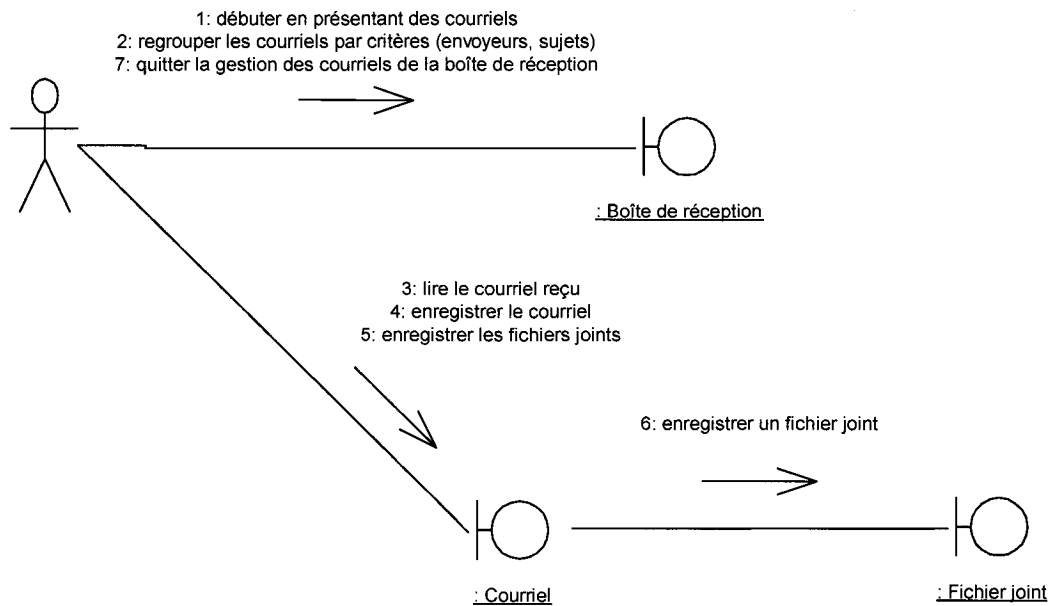


Figure 1.9 : Diagramme de collaboration de scénarimage (Kruchten, Ahlqvist et Bylund, 2001).

Un scénarimage de cas d'utilisation est décrit en termes de classe frontière en relations statiques et dynamiques que sont des agrégations, les associations et les liens. Chaque classe frontière est à son tour une représentation de haut niveau d'une fenêtre ou de son équivalent dans une interface-utilisateur (Rational, 2002).

Chaque cas d'utilisation, priorisé selon une perspective d'utilisabilité dans une itération, est traité ainsi:

1. décrire les caractéristiques des acteurs concernés ;
2. créer un scénarimage de cas d'utilisation ;
3. décrire l'enchaînement des événements du scénarimage ;
4. identifier les exigences d'utilisabilité dans le scénarimage de cas d'utilisation ;
5. rechercher les classes frontières nécessaires au scénarimage de cas d'utilisation ;
6. décrire les interactions entre les objets frontières et les acteurs ;
7. compléter les diagrammes du scénarimage du cas d'utilisation ;

8. définir les références à l'interface-utilisateur (si elle existe déjà) du scénarimage de cas d'utilisation. Pour chaque classe frontière, il faut suivre ces étapes:
 - a) décrire les responsabilités de la classe frontière ;
 - b) décrire les attributs de la classe frontière ;
 - c) décrire les relations entre les classes frontières ;
 - d) décrire les exigences d'utilisabilité de la classe frontière ;
 - e) insérer la classe frontière dans le diagramme global de classes ;
9. évaluer les résultats (Kruchten, Ahlqvist et Bylund, 2001).

Le scénarimage se distingue de la scénarimagination, qui est une technique d'explicitation des exigences inspirée du monde du cinéma. Ainsi, les productions cinématographiques commencent par des scénarimages qui décrivent les personnages, ce qui leur arrive et comment cela arrive. Pour le RUP©, la scénarimagination aide à recueillir et à raffiner les exigences du client d'une façon conviviale. La scénarimagination est un outil pour illustrer aux utilisateurs comment le système s'insérera dans l'organisation et comment il va se comporter.

Un prototype d'interface-utilisateur est une représentation de basse fidélité de l'interface-utilisateur. Par exemple, le prototype peut être une maquette papier ou une image produite à partir d'un outil de dessin, une maquette détaillée ou un prototype interactif exécutable, produit, par exemple, en Visual Basic (Kruchten, Ahlqvist et Bylund, 2001).

Il faut exécuter les étapes suivantes lors de la réalisation du prototype d'interface-utilisateur.

1. identifier la fenêtre principale ;
2. concevoir l'apparence des fenêtres principales ;
3. concevoir les opérations des fenêtres principales ;
4. concevoir les propriétés des fenêtres ;
5. concevoir les opérations impliquant plusieurs objets ;
6. concevoir les autres caractéristiques (Kruchten, Ahlqvist et Bylund, 2001).

1.4.5.2 Le concepteur d'interface-utilisateur

Le travailleur responsable de la conception des interfaces-utilisateurs est le concepteur d'interface-utilisateur. Il est impliqué dans deux activités : la modélisation de l'interface et le prototypage de l'interface-utilisateur. Les deux artefacts sont développés durant la modélisation et le prototypage de l'interface. Le concepteur d'interface-utilisateur mène et coordonne le prototypage et la conception d'interface-utilisateur en :

- explicitant les exigences de l'interface-utilisateur, incluant les exigences-utilisateurs ;
- construisant les prototypes d'interface-utilisateur ;
- impliquant les autres parties prenantes de l'interface-utilisateur, notamment les utilisateurs finaux, lors de revues d'utilisabilité et de sessions de tests d'utilisabilité ;
- révisant et fournissant la rétroaction appropriée sur l'implémentation finale de l'interface-utilisateur, telle que créée par les autres développeurs et concepteurs (Kruchten, Ahlqvist et Bylund, 2001).

La figure 1.10 illustre les rôles des travailleurs impliqués dans la gestion des exigences.

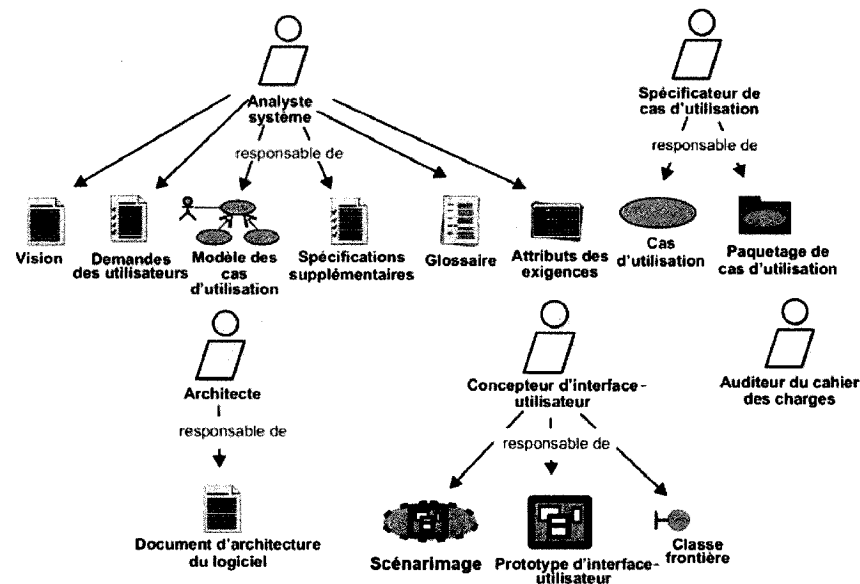


Figure 1.10 : Travailleurs et artefacts dans l'enchaînement des activités de gestion des exigences (Kruchten, 2001).

Le concepteur d'interface-utilisateur ne doit pas implémenter lui-même l'interface-utilisateur. Il doit plutôt s'appliquer à la conception et à l'aménagement visuel de l'interface-utilisateur pour deux raisons. D'abord, les qualifications requises pour un concepteur d'interface-utilisateur doivent être mises à niveau fréquemment et être optimisées pour le projet en cours et pour le type d'application. Les exigences d'utilisabilité sont souvent uniques à un projet. Leur définition et leur articulation demandent du temps et de la concentration. Ensuite, les considérations d'implémentation par opposition aux préoccupations d'utilisabilité peuvent entrer en conflit. Le concepteur d'interface-utilisateur doit être à l'abri de cette influence (Kruchten, Ahlqvist et Bylund, 2001).

CHAPITRE II

PROBLÉMATIQUE ET RECHERCHES CONNEXES

La conformité du RUP© à ISO 13407 relève d'une problématique plus vaste qui est celle de l'intégration de la conception centrée sur l'utilisateur aux pratiques du génie logiciel. Cette intégration au génie logiciel a fait l'objet de plusieurs recherches, particulièrement dans le domaine de la conception centrée sur l'utilisateur. Par ailleurs, des recherches ont relevé des différences entre la norme ISO 13407 et l'ingénierie de l'utilisabilité. Enfin, des chercheurs ont identifié des problèmes pour le RUP© à répondre aux principes de la conception centrée sur l'utilisateur en général et à ISO 13407 en particulier. Ils proposent des solutions.

2.1 Génie logiciel et ingénierie de l'utilisabilité

Judy Brown (1997) note que l'approche de l'ingénierie de l'utilisabilité et celle du génie logiciel sont différentes. Ainsi, le génie logiciel modélise bien le domaine du problème et sait traiter de questions financières et de gestion. Par contre, l'ingénierie de l'utilisabilité traite de l'utilisateur et de sa tâche et sait rester proche de la réalité. Selon elle, l'écart qui existe entre les deux disciplines est imputable à la non-reconnaissance de l'ingénierie de l'utilisabilité par le génie logiciel, d'une part. D'autre part, l'ingénierie de l'utilisabilité ne clarifie pas la nature de sa relation avec le génie logiciel.

Depuis quelques années, plusieurs travaux dans le monde de l'interaction humain-machine ont été réalisés pour rapprocher le monde de la conception centrée sur l'utilisateur avec le monde du génie logiciel. Seffah, Desmarais et Metzker (2005) ont identifié quatre catégories de recherches qui traitent de ce rapprochement :

- rajouter aux artefacts du génie logiciel des attributs de la spécification d'interface-utilisateur, comme l'annotation des cas d'utilisation avec des descriptions de tâche ;
- étendre la notation et les modèles du génie logiciel orienté objet ;
- ajouter des éléments des méthodes de conception centrée sur l'utilisateur pour la cueillette des exigences par des entrevues et des observations sur le terrain, dériver des modèles conceptuels à partir de scénarios, des modèles de tâches et des cas d'utilisation et utiliser des « persona » pour comprendre et modéliser les utilisateurs finaux ;

- élaborer de nouvelles méthodologies pour la conception de systèmes interactifs comme celle de Mayhew (1999) ou en complétant les méthodologies existantes.

La conception centrée « utilisation » imaginée par Constantine et Lockwood (2001) est un exemple de la première catégorie. Selon eux, ce n'est pas l'analyse du contexte de la tâche, mais la modélisation de la tâche qui est le pivot de la conception d'interface-utilisateur. Les cas d'utilisation de l'approche orientée objet pourraient traduire cette modélisation de la tâche s'ils étaient modifiés. Ils prendraient le nom de cas d'utilisation « essentiels ». Les étapes de scénario qu'ils décrivent devraient avoir une forme plus abstraite afin de ne pas se compromettre sur les détails d'implémentation à l'étape de l'analyse des exigences. Le tableau 2.1 présente un cas d'utilisation d'un retrait de guichet automatique de banque avec un cas d'utilisation essentiel correspondant.

Tableau 2.1 : Cas d'utilisation essentiel (Constantine et Lockwood, 2001)

Cas d'utilisation: modèle utilisateur action		Cas d'utilisation essentiel	
Utilisateur	Réponse du système	Intention de l'utilisateur	Responsabilité du système
Insérer carte	Lire bande magnétique Demander NIP	S'identifier	Vérifier identité Offrir des choix
Saisir NIP	Vérifier NIP Afficher menu des options de transaction	Choisir Prendre argent	Remettre argent
Appuyer clé	Affiche menu des comptes		
Appuyer clé	Demander montant		
Saisir montant	Afficher montant		
Appuyer clé	Remettre carte		
Prendre carte	Remettre argent		
Prendre argent			

Cette approche ne fait cependant pas l'unanimité en ingénierie de l'utilisabilité. Ainsi, Gulliksen, Göransson et Lif (2001) ne minimise pas l'importance des cas d'utilisation pour représenter le système tout en reconnaissant l'effort de Constantine et Lockwood à clarifier le style de rédaction des cas. Cependant, ils les jugent insuffisants pour présenter aux utilisateurs le fonctionnement du système à cause de leur complexité. À leurs yeux, un prototype d'interface-utilisateur est nécessaire pour illustrer comment le système peut soutenir la tâche. Par ailleurs, toujours selon Gulliksen, Göransson et Lif (2001), les concepteurs sont tentés de produire une interface-utilisateur par cas d'utilisation, alors que la tâche d'un utilisateur peut s'exécuter sur

plusieurs cas d'utilisation. Les ingénieurs logiciels préfèrent de plus petits cas d'utilisation pour faciliter leur manipulation lors de la conception des composants.

Sousa et Furtado (2003b) vont plus loin. Selon eux, les cas d'utilisation se différencient des descriptions de tâche nécessaires à la conception centrée sur l'utilisateur en ne fournissant pas les détails liés aux objectifs, aux préconditions, à la fréquence et aux postconditions. Ces spécifications ne sont pas un objectif des cas d'utilisation, mais sont la responsabilité de la description de tâche. Ils proposent donc que les cas d'utilisation et la modélisation de tâche soient utilisés de façon complémentaire.

2.2 ISO 13407 et l'ingénierie de l'utilisabilité

Earthy, Sherwood-Jones et Bevan (2001) affirment que la norme ISO 13407 et le rapport technique ISO TR 18529 qui la complète, donnent une définition de la conception centrée sur l'utilisateur qui peut être intégrée aux définitions de l'ingénierie logicielle et de système.

John, Bass et Adams (2003) ont fait une comparaison entre le RUP© et la conception centrée sur l'utilisateur telle que définie par ISO 13407 afin de vérifier la compatibilité des deux processus. Leur analyse préliminaire a révélé plusieurs similitudes et d'importantes asymétries entre les deux. Selon eux, les deux processus peuvent se compléter dans la mesure où des hypothèses sont formulées sur la nature des exigences, de la conception et des tests. Cependant, des différences de terminologie peuvent provoquer des problèmes de communication. D'autre part, des asymétries portent sur les qualifications du personnel et les méthodes qui assurent une validation des exigences par les utilisateurs.

Par ailleurs, selon Earthy, Sherwood-Jones et Bevan (2001), la norme ISO 13407 et le rapport technique ISO TR 18529 abordent le domaine de l'utilisabilité avec une approche plus large que l'ingénierie de l'utilisabilité traditionnelle telle que proposée, par exemple, par Mayhew (1999). Selon eux, l'ingénierie de l'utilisabilité se limite à la conception de l'interface-utilisateur alors que la conception centrée sur l'utilisateur se consacre à des activités liées à l'identification et à la spécification des fonctions qui soutiennent la tâche de l'utilisateur. Le rapport technique ISO TR 18529 identifie 8 processus de conception centrée sur l'utilisateur (PCCU) qui comprennent 36 pratiques.

Le tableau 2.2 identifie ces différences selon les processus qui sont touchés :

Tableau 2.2 : Différences entre ISO TR 18529 et les approches de l'ingénierie de l'utilisabilité
(Earthy, Sherwood-Jones et Bevan, 2001)

Processus	Titre	Différence
PCCU.1	• Garantir un contenu PCCU dans la stratégie des systèmes	Ces processus organisationnels n'ont pas été traditionnellement inclus à l'utilisabilité et ne sont que partiellement couverts par ISO 13407. Cependant l'expérience a démontré que l'engagement organisationnel est essentiel à une implémentation réussie d'une conception centrée sur l'utilisateur et des activités énumérées en PCCU1 et PCCU2.
PCCU.2	• Planifier et gérer le processus PCCU	
PCCU.7	• Introduire et opérer le système	Les activités qui ont pour objet d'évaluer l'impact sur l'organisation et les utilisateurs, la personnalisation, la formation et le soutien sont essentiels à une implémentation réussie ont rarement été intégrés à l'utilisabilité et ne sont d'ailleurs pas inclus à ISO 13407
PCCU.3	• Spécifier les exigences liées aux utilisateurs et à l'organisation	Les activités d'exigences d'utilisabilité sont souvent limitées à l'interface-utilisateur alors que ISO 13407 et ISO TR 18529 traitent les exigences-utilisateurs pour tout le système.
PCCU.5	• Produire des solutions de conception	Il y a deux activités recommandées pour la conception qui manquent à plusieurs méthodologie d'ingénierie de l'utilisabilité mais qui sont centrales à l'ergonomie et aux facteurs humains: l'allocation des fonctions et une définition de l'expérience globale de l'utilisation du système.
PCCU.4	• Comprendre et spécifier le contexte d'utilisation	Cet aspect est central à l'ingénierie de l'utilisabilité bien que les normes soient plus explicites sur le besoin de vérifier que le contexte utilisé pour l'évaluation correspond au contexte d'utilisation souhaité. Le contexte d'utilisation est complètement absent des systèmes traditionnels et du génie logiciel.
PCCU.6	• Évaluer la conception en fonction des exigences	

2.3 Conception centrée sur l'utilisateur et le RUP©

Sousa et Furtado (2003) relèvent des difficultés rencontrées avec le RUP© lors du développement d'applications interactives. Ils signalent, durant le processus de développement logiciel, l'absence de préoccupation de facteurs humains, des aspects internationaux et interculturels, du modèle centré sur l'utilisateur durant la conception d'interface-utilisateur et, enfin, de la personnalisation de l'interface-utilisateur selon l'interaction offerte par la technologie et selon les utilisateurs.

Pour y remédier, ils proposent un nouveau processus de développement pour systèmes interactifs inspiré du RUP©, le *Rational Unified Process Interactive* (RUPI) auquel ils ajoutent des travailleurs, des enchaînements d'activités et des artefacts. Le RUPI propose 9 enchaînements d'activités : exigences, analyse et conception, implémentation, test, gestion de projet, modélisation d'affaire, configuration et gestion du changement, environnement et déploiement (Sousa et Furtado 2003b). La figure 2.1 décrit les relations dans les enchaînements RUPI entre les travailleurs et leurs artefacts.

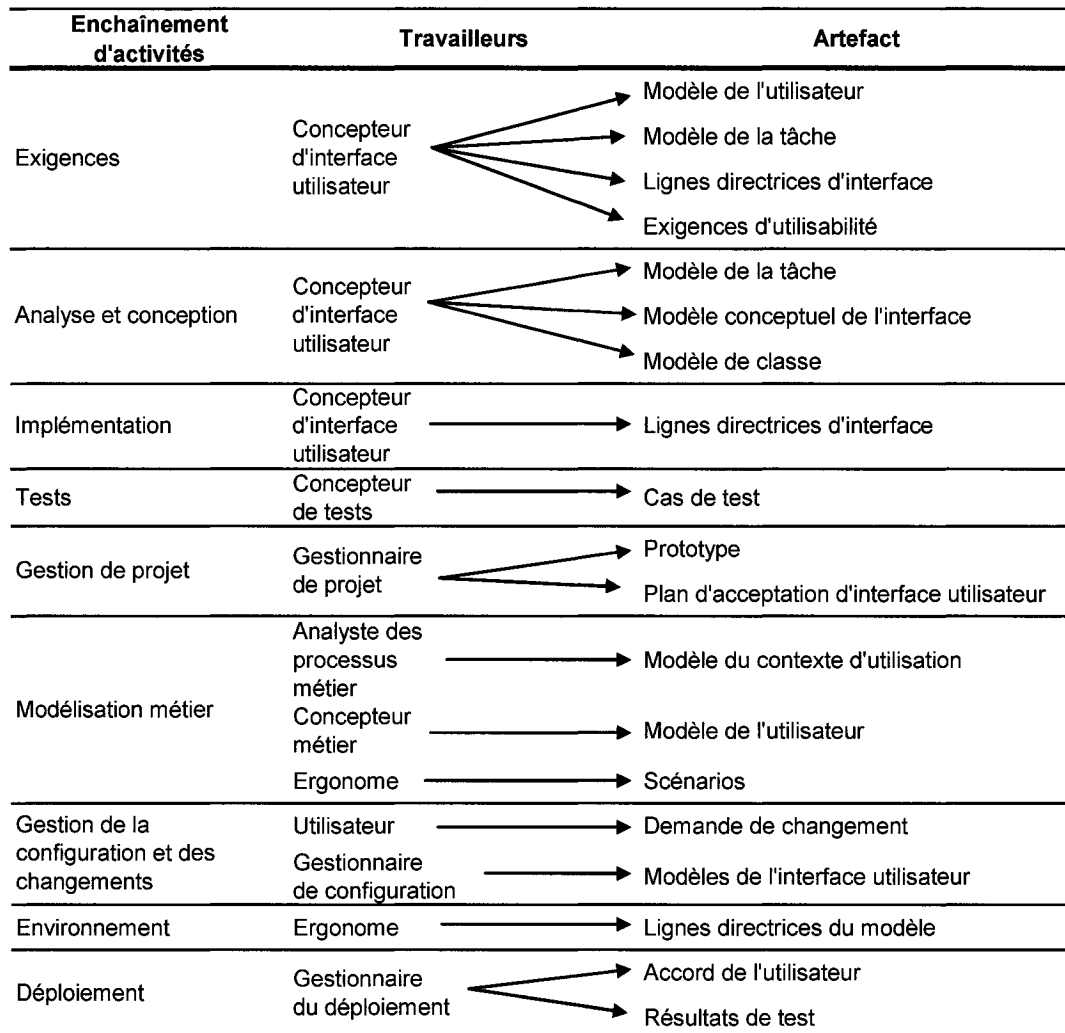


Figure 2.1 : Relations entres les enchaînements, les travailleurs et les artefacts du RUI
(Sousa et Furtado, 2003b).

La définition du processus a évolué et s'est précisée. Ainsi, dans une recherche ultérieure de Sousa et Furtado (2003c), la modélisation métier sera le premier enchaînement du processus. Le contenu de certains enchaînements a été modifié et les responsabilités de certains travailleurs ont été réaffectées.

Par exemple, lors de la modélisation métier, l'ergonome définit le contexte d'utilisation lorsque que l'analyste des processus métiers s'emploie à comprendre le domaine d'affaire (Sousa et Furtado 2003c) comme l'illustre la figure 2.2.

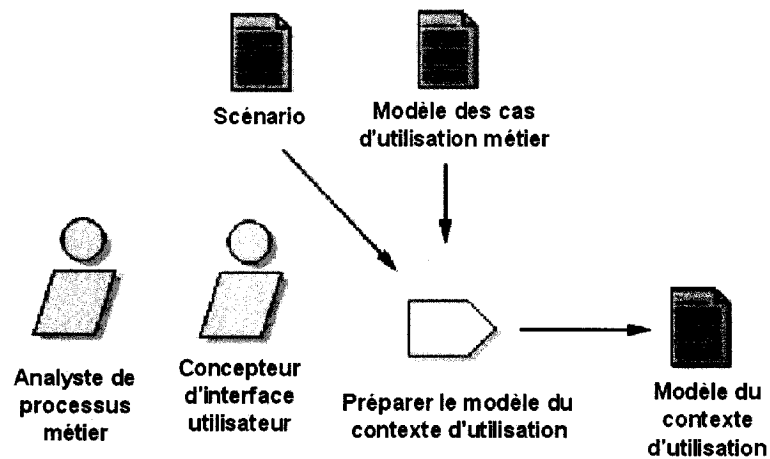


Figure 2.2 : Activités supplémentaires des activités de modélisation du métier du RUP
(Sousa et Furtado, 2003c).

2.4 ISO 13407 et le RUP©

Selon Gulliksen, Göransson et Lif (2001), l'étude de plusieurs organisations utilisant le RUP© fournit plusieurs exemples de faiblesses sur la façon dont plusieurs aspects de l'utilisabilité sont traités pour rencontrer les objectifs de la conception centrée sur l'utilisateur telle que définie par ISO 13407 :

- le RUP© donne l'impression de répondre à toutes les questions sur la conception des interfaces-utilisateurs et fait que le concepteur d'interface-utilisateur cesse de penser ;
- il n'y a pas ce soutien à un engagement profond à l'utilisabilité en dehors de la définition des exigences là où l'utilisabilité doit être perçue comme une entité mesurable contenant des demandes fonctionnelles et non fonctionnelles qui doivent être intégrées à tout le processus de développement et pas seulement à la conception des interfaces-utilisateurs ;
- la participation attendue de l'utilisateur ne distingue pas clairement l'utilisateur final de l'expert du domaine. Les utilisateurs engagés trop étroitement dans le processus de développement deviennent rapidement biaisés en faveur du développement ;

- le seul travailleur qui a un lien avec l'utilisabilité est le concepteur d'interface-utilisateur qui, selon le RUP©, a la responsabilité d'aménager visuellement l'interface-utilisateur sans pour autant avoir une quelconque compétence en utilisabilité ;
- le processus soutient peu l'analyse de la tâche des utilisateurs ou l'identification des groupes d'utilisateur selon la perspective de l'analyse du contexte d'utilisation. Or, le contexte d'utilisation est fondamental en ISO 13407, puisqu'il procure une information mesurable sur les utilisateurs, les tâches, l'équipement et sur l'environnement physique et social de l'utilisateur ;
- il y a un risque évident que les utilisateurs soient introduits seulement au début du projet et soient par la suite oubliés lorsque les préoccupations se tourneront de plus en plus vers l'architecture du système.

Afin d'y remédier, Gulliksen, Göransson et Lif (2001) ne recommandent pas l'ajout d'artefacts mais proposent un nouveau travailleur : un concepteur de l'utilisabilité. Selon leur observation de plusieurs projets appliquant le RUP© dans le processus de conception d'interface d'utilisateur, la conception centrée sur l'utilisateur réussie avec le RUP© est le résultat de la participation de personnes ayant des qualifications élevées et une expérience de l'interaction humain ordinateur. Les projets avec un personnel moins expérimenté, utilisant le RUP© dans le processus de conception d'interface-utilisateur, connaissent de graves problèmes d'utilisabilité qui sont imputables à un manque de soutien de la part du processus lui-même.

2.5 Principes de conception centrée sur l'utilisateur

Pour leur part, Gulliksen et al. (2005) ont soulevé une série de problèmes en étudiant l'application d'un processus de conception centrée sur l'utilisateur lors d'un développement logiciel avec le processus RUP©.

Les problèmes observés sont :

1. les développeurs sont plus préoccupés par l'implémentation de toutes les fonctionnalités que par l'utilisabilité du système. Il n'y a pas de préoccupation soutenue tout au long du cycle de vie pour l'utilisateur ;
2. malgré leurs importantes qualifications, l'opinion des concepteurs en utilisabilité est ignorée dans les dernières phases du projet ;

3. les cas d'utilisation sont la préoccupation prédominante du projet et leur production supplante la compréhension des besoins réels des utilisateurs ;
4. le langage UML est inaccessible aux utilisateurs au point qu'ils considèrent comme une délivrance la présentation de prototypes d'interface-utilisateur pour comprendre comment le système fonctionnera lors de leur utilisation ;
5. en cours de développement, les décisions majeures qui affectent l'utilisabilité du système sont prises contre l'avis des concepteurs en utilisabilité ;
6. l'attitude centrée sur l'utilisateur est compromise par des intérêts professionnels incompatibles avec les principes de conception pluridisciplinaire.

Pour parer à ces problèmes, ils proposent de formuler des principes de conception centrée sur l'utilisateur afin de compléter ceux de ISO 13407 et de Gould, Boies et Ukelson (1997). Ces derniers en avaient énoncé quatre :

1. une attention immédiate et continue aux utilisateurs ;
2. une évaluation empirique ;
3. une conception itérative ;
4. une conception intégrée où tous les aspects de l'utilisabilité évoluent ensemble.

À ces principes, Gulliksen et al. (2005) proposent d'en ajouter 12 autres :

1. une préoccupation soutenue pour l'utilisateur ;
2. une participation active de l'utilisateur ;
3. un développement itératif ;
4. des représentations simples de la conception ;
5. un prototypage continu ;
6. une évaluation de l'utilisation en contexte ;
7. des activités de conception dédiées ;
8. une attitude professionnelle ;
9. un champion de l'utilisabilité ;

10. une conception holistique ;
11. une personnalisation des processus ;
12. une attitude centrée sur l'utilisateur.

CHAPITRE III

ISO 13407 ET LE RUP©

La norme ISO 13407 énonce quatre grands principes pour définir la conception centrée sur l'utilisateur. Dans la présente étude de cas, ces principes serviront d'étalon pour évaluer comment le RUP©, l'application faite du RUP© par l'équipe du programme et comment l'intervention ergonomique se sont conformés à ces principes. Il convient donc d'abord de préciser la portée de ces principes et comment le RUP© reprend ces principes.

Ainsi, pour chaque principe, ce chapitre décrit tour à tour comment la norme les formule et comment le RUP© les interprète. Le principe de l'itération des solutions de conception est scruté plus à fond. On y aborde la gestion des itérations, la matérialisation des solutions de conception et l'évaluation des solutions.

3.1 Participation active des utilisateurs et compréhension claire des exigences liées à l'utilisateur et à la tâche

Selon le premier principe, la participation active des utilisateurs est nécessaire. La nature de cette participation peut varier selon les activités. ISO 13407 propose différents types de participation. Le RUP© prescrit des ateliers de travail pour différentes circonstances dans lesquelles, il souhaite voir participer les parties prenantes. Par ailleurs, il recommande une série d'activités afin de mieux connaître l'utilisateur.

La compréhension claire des exigences repose sur la connaissance du contexte d'utilisation. ISO 13407 établit une liste élaborée d'éléments à décrire pour définir le contexte d'utilisation. Le RUP© ne définit pas formellement le contexte d'utilisation, mais on peut, selon lui, retrouver des éléments de ce contexte dans les documents Vision et Vision métier. Selon le RUP©, les cas d'utilisation et les scénarimages correspondent à l'analyse contextuelle de la tâche.

3.1.1 La participation des utilisateurs

Pour ISO 13407, la participation des utilisateurs est une source de connaissances et l'incidence de cette participation est proportionnelle à l'interaction avec les développeurs. ISO propose que, lors de la mise au point de produits personnalisés, comme une application de gestion dans une

entreprise, l'implication des utilisateurs soit directe dans le processus de développement. Pour des produits génériques, comme des produits grand public, ISO trouve tout aussi essentiel que des utilisateurs ou des représentants appropriés soient impliqués dans le développement.

3.1.2 ISO 13407 : les utilisateurs, l'environnement et la tâche

Selon ISO 13407, la première activité de la conception centrée sur l'opérateur humain consiste à « comprendre et à spécifier le contexte d'utilisation » (ISO, 1999). Le contexte d'utilisation comprend les caractéristiques des utilisateurs potentiels, les tâches confiées aux utilisateurs et l'environnement dans lequel le système sera utilisé.

Le tableau 3.1 donne un exemple de caractéristiques de contexte d'utilisation tel que proposé par la norme ISO 9412-11 (ISO, 1998). Cet exemple rassemble des caractéristiques qui ne sont pas nécessairement exclusives. Selon le contexte, certaines peuvent ne pas être pertinentes et d'autres peuvent se rajouter. Les types d'utilisateurs y sont des regroupements qui se distinguent notamment par des niveaux d'expérience et par des rôles tenus. ISO 13407 regroupe la description des équipements avec la description de l'environnement.

3.1.2.1 Les utilisateurs et l'environnement

La description des utilisateurs, de l'environnement et des équipements consiste en une cueillette de renseignements qui ne nécessitent pas de technique particulière autre que de faire l'inventaire des informations pertinentes. La description de la tâche, par contre, procède d'une analyse plus élaborée.

3.1.2.2 La tâche

Selon ISO, la tâche se définit comme une « action ou ensemble d'actions que doit effectuer le travailleur pour atteindre l'objectif du système de travail » (ISO, 2004). Elle est un élément d'une activité qui se définit comme une « organisation et succession dans le temps et l'espace des tâches d'un individu ou enchaînement de toutes les actions d'un travailleur au sein d'un système de travail » (ISO, 2004). En contrepartie, un processus se définit comme « un ensemble d'activités compatibles qui transforment des intrants en extrants » (ISO, 2000).

La description des activités et des processus peut être nécessaire « pour concevoir ou évaluer les détails et interactions avec le produit ». Cependant, on ne doit pas « décrire les tâches

uniquement du point de vue des fonctions ou des caractéristiques fournies par un produit ou un système » (ISO, 1998). Il faut décrire les tâches selon les objectifs à atteindre (ISO, 1998).

Tableau 3.1 : Exemple de caractéristiques du contexte d'utilisation (ISO, 1998)

Utilisateurs	Tâches	Équipement
Types d'utilisateurs	Décomposition de la tâche	Description de base
Utilisateurs principaux	Nom de la tâche	Identification du produit
Utilisateurs secondaires et indirects	Fréquence d'utilisation de la tâche	Description du produit
	Durée de la tâche	Grands domaines d'application
	Fréquence des événements	Fonctions principales
Connaissances et savoir-faire	Souplesse de la tâche	
Connaissance, savoir-faire du produit	Exigences physiques et mentales	Spécifications
Connaissance, savoir-faire du système	Dépendances de la tâche	Matériel
Expérience de la tâche	Résultats de la tâche	Logiciel
Expérience de l'organisation	Risques encourus en cas d'erreurs	Documents
Niveau de formation	Exigences essentielles en matière	Services
Compétences dactylo		Autres éléments
Diplômes		
Connaissances linguistiques		
Culture générale		
Caractéristiques personnelles		
Age		
Sexe		
Aptitudes physiques		
Limitations et handicaps physiques		
Aptitudes intellectuelles		
Attitude		
Motivation		

Environnement organisationnel	Environnement	
	Environnement technique	Environnement physique
Structure	Configuration	Environnement du poste de travail
Horaires de travail	Matériel	Conditions atmosphériques
Travail en groupe	Logiciel	Environnement acoustique
Fonctions exercées	Documents de référence	Environnement thermique
Habitudes de travail		Environnement visuel
Assistance		Instabilité de l'environnement
Interruptions		
Structure de l'encadrement		Conception du poste de travail
Structure de communication		Espace et mobilier
		Posture
		Emplacement géographique
Attitudes et culture		Sécurité du lieu de travail
Politique d'utilisation des ordinateurs		Risques pour la santé
Objectifs organisationnels		Vêtements et équipements de protection
Relations industrielles		
Conception du poste		
Flexibilité du poste		
Contrôle des performances		
Feed-back des performances		
Rythme		
Autonomie		
Liberté d'usage		

La description doit inclure « la répartition des activités et des phases opérationnelles entre les composantes humaines et techniques » (ISO, 1999). Cette répartition repose sur une analyse des capacités et une attribution des fonctions qui tient compte des « capacités et limites, à la fois humaines et techniques, des éléments planifiés pour la satisfaction des exigences du système » (ISO, 2004).

ISO (1998) précise que les tâches peuvent être identifiées par une analyse de la tâche. Et « pour les besoins de l'évaluation d'utilisabilité, on sélectionne généralement un ensemble de tâches fondamentales représentant les aspects significatifs de la tâche complète ».

3.1.3 RUP© : participation souhaitable et plusieurs contextes

Pour le RUP©, la participation des utilisateurs est souhaitable et s'exprime dans des ateliers. Le RUP© définit différents contextes sans préciser s'il s'agit de contextes d'utilisation ou de contextes du problème. Des associations sont faites entre certaines activités et artefacts du RUP© et les éléments de contexte attendus par ISO 13407.

3.1.3.1 La participation des utilisateurs

La participation active des utilisateurs n'est pas un objectif du RUP©. Néanmoins, il recommande de les impliquer tôt dans le processus. Il propose de les impliquer dans des ateliers de travail portant sur l'explicitation des exigences et les cas d'utilisation avec les autres parties prenantes. Il est aussi recommandé de les exposer tôt aux prototypes d'interface-utilisateur.

Par ailleurs, afin d'avoir un portrait plus précis d'une situation, il suggère différentes façons de se familiariser avec les utilisateurs, leurs tâches et exigences qu'on peut voir dans le tableau 3.2 :

**Tableau 3.2 : Se familiariser avec les utilisateurs, leurs tâches et exigences: suggestions
(Rational 2002)**

Activités	
• Parler avec les utilisateurs • Observer le travail des utilisateurs • S'informer sur l'organisation du travail • Demander aux utilisateurs de parler tout haut en travaillant • Intégrer un utilisateur expert à l'équipe • Utiliser des questionnaires et faire des sondages	• Visiter les locaux des clients • Filmer le travail des utilisateurs • Essayer soi-même • Faire de la conception coopérative • Faire une analyse de tâche • Développer des objectifs de test

3.1.3.2 Les contextes

Pour le RUP©, il y a plusieurs contextes. Ce qui pourrait correspondre au contexte de l'utilisation est divisé en deux parties : le rôle et l'utilisateur. La figure 3.1 illustre l'interprétation faite par le RUP©.

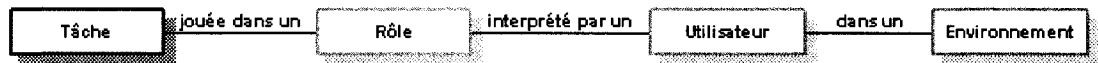


Figure 3.1 : Relations entre contextes (Rational, 2002).

Et le RUP© associe ces contextes à des artefacts qui sont inventoriés dans le tableau 3.3.

Tableau 3.3 : Contextes et artefacts RUP© (Rational, 2002)

Contexte	Artefact RUP
Environnement	Haut niveau • Environnement du client dans Vision de métier • Demandes de partie prenante • Environnement de l'utilisateur dans Vision
Utilisateurs	Haut niveau • Profils du client dans Vision de métier • Demandes de partie prenante • Profils de l'utilisateur dans Vision
Rôles	Haut niveau • Acteurs métier • Travailleurs métier Détaillé • Acteur
Tâches	Haut niveau • Demandes de partie prenante • Caractéristiques de produit dans Vision Détaillé • Scénarimage • Cas d'utilisation

3.1.3.3 L'environnement: Vision de métier et Vision

L'artefact Vision de métier décrit les objectifs et le fonctionnement de l'organisation. L'environnement du client comprend l'environnement de travail : les personnes impliquées dans l'exécution des processus et des activités ainsi que les autres organisations avec lesquelles le client fait affaire.

L'artefact Vision décrit le système à construire selon la perspective des parties prenantes. L'environnement de l'utilisateur correspond à l'environnement de travail du client. On y décrit

le nombre de personnes impliquées dans l'exécution de tâches, la longueur des cycles de tâche, les plateformes des systèmes patrimoniaux, les futures plateformes ainsi que les autres applications déjà en opération et le degré d'intégration avec l'application en développement.

Les demandes de partie prenante sont tout type de requête portant sur le système à développer. Les types de demandes de partie prenante sont énumérés dans le tableau 3.4.

Tableau 3.4 : Types de demandes de partie prenante (Rational 2002)

Types de demandes	
• Le résultat d'entrevue avec une partie prenante • Une demande de changement, un énoncé de travail • Une description d'anomalie, des règles d'affaire • Des systèmes patrimoniaux	• Le résultat d'une session d'explicitation • Un appel d'offres, un énoncé de mission • Des lois et règlements • Un modèle d'affaire

3.1.3.4 Les utilisateurs

Les utilisateurs sont décrits par les artefacts Vision métier et Vision. Dans le document Vision métier, bien qu'il s'agisse de types d'utilisateurs, ils sont décrits à la rubrique profils de client. On y énumère les personnes qui représentent les groupes d'utilisateurs en y précisant notamment les caractéristiques suivantes : une description, leurs qualifications, leurs responsabilités par rapport au système en développement, ce qu'ils considèrent comme un succès dans le cadre de l'entreprise, leur implication dans le projet traduit par un rôle RUP©, les biens livrables qu'ils produisent dans l'entreprise et les autres problématiques qui les concernent.

Dans le document Vision, le même exercice est fait avec les mêmes paramètres pour chacun des utilisateurs du système. On y a ajouté le nom du représentant des utilisateurs correspondant défini dans le document Vision métier. Selon les prescriptions du gabarit de l'artefact, les utilisateurs sont ainsi identifiés pour s'assurer qu'ils participeront au processus de modélisation des exigences. Il s'agit de connaître les principaux problèmes auxquels ils sont confrontés (Rational, 2002).

3.1.3.5 Les rôles

Les acteurs métier représentent les personnes ou les choses qui jouent un rôle auprès de l'organisation. Les clients, les partenaires, les fournisseurs, un marché cible et les autorités locales sont notamment des acteurs métier. Un travailleur métier est une abstraction de personne

ou de système informatique qui joue un rôle à l'intérieur de l'organisation lors de l'exécution d'un cas d'utilisation-métier.

Un acteur, appelé aussi acteur système pour le distinguer de l'acteur métier, se définit comme un ensemble cohérent de rôles que les utilisateurs peuvent jouer lors de leurs interactions avec le système. Une instance d'acteur peut donc être une personne, un composant matériel externe ou un système externe. Le tableau 3.5 décrit les propriétés d'un acteur.

Tableau 3.5 : Propriétés d'acteur (Rational, 2002)

Propriété	Description
Nom	Nom de l'acteur
Description	Courte description des responsabilités et des besoins de l'acteur face au système.
Caractéristiques (acteur humain)	• Environnement physique • Nombre d'utilisateurs représentés • Qualifications et formation • Connaissance de l'informatique • Autres applications utilisées • Sexe • Age • Antécédents culturels • Autres
Relations	• Associations avec les cas d'utilisation • Généralisations

La différence entre un acteur et un utilisateur du système est que l'acteur est une classe d'utilisateur plutôt qu'un utilisateur particulier. Plusieurs utilisateurs peuvent donc jouer le même rôle et correspondre au même acteur. Seuls ceux qui communiquent directement avec le système doivent être considérés comme acteurs du système contrairement aux acteurs métier. L'identification des acteurs et de leurs associations relève de l'analyste système. Le concepteur d'interface-utilisateur identifie cependant les caractéristiques de l'utilisateur (Rational, 2002).

3.1.3.6 Les tâches

Pour le RUP®, les tâches sont définies à partir des caractéristiques de produit inventoriées dans le document Vision et à un niveau détaillé par les scénarimages et par les cas d'utilisation.

Les caractéristiques de produit sont des capacités du système définies à haut niveau qui procurent un avantage aux utilisateurs et qui doivent être perceptibles par les utilisateurs, les opérateurs ou par d'autres systèmes externes. Les caractéristiques de produit servent notamment à élaborer le modèle des cas d'utilisation et sont éventuellement transformées en exigences

logicielles. Elles découlent des règles d'affaire et des demandes des parties prenantes. La figure 3.2 illustre les relations entre les demandes des parties prenantes et les autres artefacts.

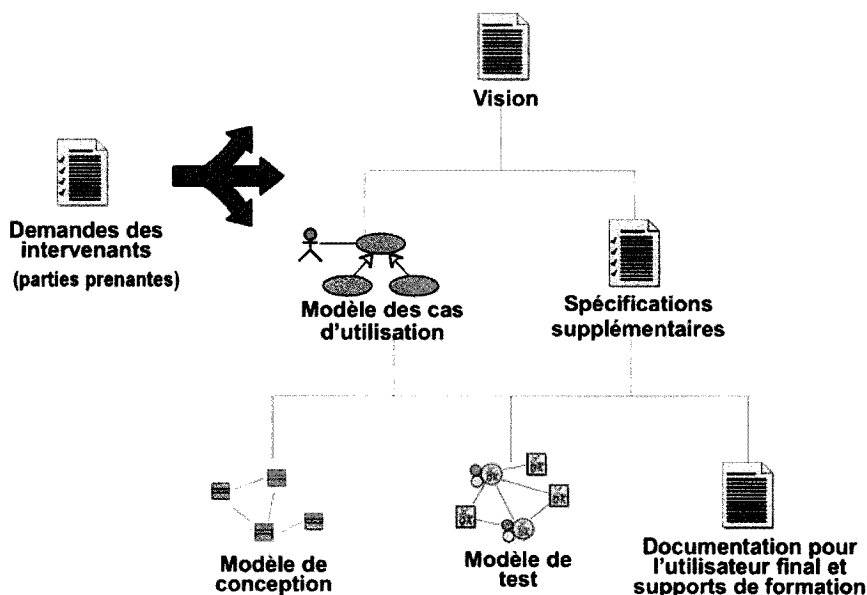


Figure 3.2 : Demandes des parties prenantes (Kruchten, 2001).

Le RUP® associe la tâche d'un utilisateur à un cas d'utilisation. Les interfaces-utilisateurs sont produites à partir des cas d'utilisation qui décrivent le fonctionnement du système. Selon Kruchten (2001), on se base sur le modèle des cas d'utilisation et sur les spécifications supplémentaires.

Le modèle des cas d'utilisation identifie les cas d'utilisation. Un cas d'utilisation « est une séquence d'actions que réalise un système et qui fournit un résultat observable ayant une valeur pour un acteur particulier » (Kruchten, 2001). Les cas d'utilisation sont inspirés des processus d'affaires ou de ce que le RUP® appelle les cas d'utilisation-métier. On identifie les acteurs du système à partir des travailleurs métier qui ont été définis lors de la modélisation du métier.

Les cas d'utilisation-métier sont des processus d'affaire qui transcendent les frontières de l'organisation en incluant les partenaires et fournisseurs, dans la mesure où ils fournissent une valeur ajoutée à la partie prenante de l'entreprise. De la même façon, le modèle des cas

d'utilisation est défini à partir du modèle objet métier et les cas d'utilisation système s'inspirent des cas d'utilisation-métier. C'est ce qui est reproduit à la figure 3.3.

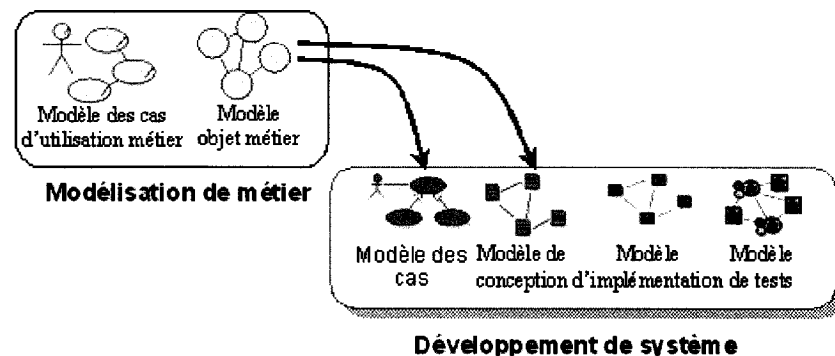


Figure 3.3 : Des modèles métier aux modèles système (Kruchten, 2001).

Le modèle objet métier décrit comment les cas d'utilisation-métier sont réalisés en termes de travailleurs et d'entités métier, ainsi que leurs interactions. Il est une abstraction des liens entre les travailleurs et les entités métier et de leur collaboration nécessaire à l'exécution des cas d'utilisation-métier (Rational, 2002).

Kruchten (2001) résume ainsi le lien entre les processus métier et les cas d'utilisation :

« Les relations entre les éléments du modèle métier et ceux des modèles informatiques sont les suivantes :

- les travailleurs métier deviennent les acteurs du système informatique ;
- à partir des actions et des comportements des travailleurs métier, on peut déterminer les cas d'utilisation du système, pour les parties que l'on souhaite automatiser ;
- à partir des entités métier, on peut déduire les classes d'entités du modèle d'analyse du système. »

Lorsque la modélisation métier n'est pas faite, les cas d'utilisation sont alimentés par les caractéristiques de produit incluses dans le document Vision, les demandes des parties prenantes et les exigences fonctionnelles (Rational, 2002).

Le cas d'utilisation doit « fournir un résultat utile à un type d'utilisateur du système et on s'assure qu'ils en obtiennent exactement ce qu'ils attendent » afin d'éviter des cas d'utilisation trop grands dans lesquels on essaie de satisfaire les exigences de tous sans en satisfaire aucune

(Kruchten, 2001). Les scénarimages, selon le RUP© contribuent à décrire la tâche par leur description des flots d'événement.

En résumé, un cas d'utilisation est dérivé d'un cas d'utilisation-métier qui correspond à un processus d'affaires. Lorsqu'il n'y a pas de modélisation d'affaire, le cas d'utilisation est dérivé de caractéristiques de produit et de demandes de partie prenante. Le cas d'utilisation ne doit pas être trop grand et doit fournir un résultat utile. Le scénarimage décrit le flot des événements.

3.2 Répartition appropriée des fonctions entre les utilisateurs et la technologie

Les fonctions d'un système sont définies par des exigences. Selon ce deuxième principe d'ISO 13407, l'identification de ces exigences doit se faire à partir de ce qu'il convient d'automatiser selon la nature de la tâche et des responsabilités de l'utilisateur. Pour le RUP©, le contexte d'utilisation n'intervient pas pour déterminer ce qu'il désigne comme les exigences d'automatisation qui sont définies par la capacité d'automatisation.

3.2.1 ISO 13407 : exigences liées à l'utilisateur et à l'organisation

La spécification doit définir l'allocation des fonctions, soit le partage des tâches accomplies par la personne et celles accomplies par des moyens techniques.

Il faut spécifier les exigences-utilisateurs que la norme désigne comme les exigences liées à l'utilisateur et à l'organisation en s'inspirant des éléments de définition qu'on peut trouver dans le tableau 3.6.

Tableau 3.6 : Éléments de définitions des exigences-utilisateurs (ISO 1999)

Exigences-utilisateurs	
• Les performances requises • La coopération et la communication entre les utilisateurs et le reste des parties prenantes • L'exécution des tâches • La gestion des changements, y compris la formation et le personnel à impliquer • La conception de l'interface-utilisateur	• Les exigences d'ordre statutaire ou juridique, y compris les aspects relatifs à la sécurité et à la santé • Le travail confié aux utilisateurs y compris la répartition des tâches, le bien-être et la motivation • La conception et l'organisation du travail • La faisabilité du traitement et de la maintenance

3.2.2 RUP© : exigences d'automatisation

Il n'y a pas de distinction qui est faite entre les fonctions assumées par l'utilisateur et la technologie. Le RUP© ne distingue pas les exigences-utilisateurs des exigences fonctionnelles et autres. Par contre, il existe des exigences d'automatisation. Ces dernières sont des exigences fonctionnelles qui désignent quelles portions des processus d'affaires seront automatisées et des exigences non fonctionnelles qui indiquent les contraintes techniques qui baliseront cette automatisation. Il faut donc, pour y arriver, explorer quelles portions des processus d'affaire peuvent et doivent être automatisées, comprendre comment les systèmes patrimoniaux sont intégrés à l'organisation et en dériver les exigences système. L'état de l'art de la technologie disponible à l'intérieur comme à l'extérieur de l'organisation contribue à identifier les exigences d'automatisation.

Les exigences liées à la performance et à l'utilisabilité sont traitées. Leur explicitation ne repose cependant pas sur une étude du contexte d'utilisation mais sur des demandes formulées par les parties prenantes. Le scénarimage regroupe les exigences d'utilisabilité applicables au cas d'utilisation correspondant. Elles sont définies selon l'efficacité attendue par rapport à l'utilisation d'une alternative au système en construction. Un produit compétiteur, un système patrimonial, une version précédente du système ou un manuel de procédures peuvent être utilisés pour déterminer les exigences d'utilisabilité (Rational, 2002). Elles sont cataloguées comme des exigences non fonctionnelles. Les exigences de sécurité sont traitées comme des exigences non fonctionnelles.

3.3 Itération des solutions de conception

Le troisième principe d'ISO 13407 est l'itération des solutions de conception. Il suppose la matérialisation de solutions, des évaluations des solutions et une gestion des itérations. Elle est ici présentée sous ces différents aspects. Pour ISO 13407, l'élaboration des solutions de conception repose sur une approche itérative où une solution est conçue et présentée aux utilisateurs, puis soumise à une évaluation et reprise itérativement jusqu'à ce que la solution soit satisfaisante pour toutes les parties. Le RUP© reconnaît l'approche itérative de la conception centrée sur l'utilisateur, mais n'en fait pas spécifiquement un enchaînement d'activités défini.

3.3.1 Produire les solutions de conception et les présenter aux utilisateurs

La production initiale d'une solution de conception s'inspire d'abord du contexte d'utilisation pour ISO 13407. L'évolution de la solution suppose que des modifications sont apportées à partir de constatations tirées de différentes formes d'évaluation. Pour le RUP©, la conception de la solution de conception s'inspire d'abord de guides et normes définis en début de projet.

3.3.1.1 ISO 13407 : utiliser les connaissances acquises

Le contexte d'utilisation fait partie des connaissances acquises qui inspirent la version initiale de la solution de conception. Les autres connaissances issues de l'état des techniques, de l'expérience et des connaissances des participants ainsi que des normes et guides alimentent aussi cette conception initiale. Par la suite, la solution est soumise à différentes évaluations. Les connaissances issues de ces modifications alimentent les modifications de la solution.

3.3.1.2 RUP© : normes et guides

La matérialisation des solutions de conception repose, pour le RUP©, sur des normes de conception d'interface et des guides de style qui regroupent des directives pertinentes à la conception des interfaces-utilisateurs. Ces guides et directives sont produits dans la discipline environnement au début de la phase inception par le concepteur d'interface-utilisateur.

Le RUP© n'est pas très explicite sur la façon de créer ces guides sauf pour indiquer qu'ils doivent s'inspirer de normes et guides reconnus selon la plateforme technologique utilisée et qu'ils doivent être personnalisés. Le RUP© prévoit que ces guides soient améliorés durant un projet suite à des activités de « test de guide ». Le guide d'interface-utilisateur proposé par le RUP© donne des directives sur la façon de concevoir des fenêtres *Windows*.

Quant au prototype d'interface-utilisateur, le RUP© recommande qu'il soit présenté tôt à l'utilisateur pour obtenir son avis.

3.3.2 Gérer l'itération des solutions de conception

Pour ISO 13407, un processus itératif doit être planifié. Pour le RUP©, la planification des itérations s'effectue à l'échelle du cycle de vie. Il reconnaît un cadre intégré des itérations dans le cadre d'une conception centrée sur l'utilisateur.

3.3.2.1 ISO 13407 : itérations pour la conception

Les itérations conjuguées à l'implication des utilisateurs procurent un retour d'information permettant d'améliorer les solutions de conception. Cela diminue les risques de se retrouver face à un système ne répondant pas aux exigences-utilisateurs. Les itérations doivent se poursuivre jusqu'à ce que les objectifs de conception soient atteints. ISO (1999) juge important que « la planification de projet permette l'itération et l'incorporation d'un retour d'information en provenance de l'utilisateur ».

3.3.2.2 RUP© : itérations planifiées du cycle de vie

Le RUP© prévoit que les prototypes doivent faire l'objet d'une rétroaction des utilisateurs selon les prescriptions du « concept » de la conception centrée sur l'utilisateur (Rational, 2002). Cependant, bien que le RUP© prévoie une planification des itérations à l'échelle du cycle de vie, il ne prescrit pas une planification des itérations pour la conception du prototype d'interface-utilisateur. Le RUP© reconnaît, à la conception centrée sur l'utilisateur, un processus itératif qu'il partage. L'itération est nécessaire à cause des changements des besoins des utilisateurs dans le temps et de la complexité de la production de solutions de conception qui ont à conjuguer avec divers besoins. Rational (2002) note que, pour les méthodes centrées sur l'utilisateur, la conception itérative doit prendre place dans un cadre intégré.

3.3.3 Évaluer les conceptions par rapport aux exigences

L'évaluation permet d'élaborer les solutions de conception, de valider les exigences et d'assurer une surveillance pour les autres étapes de la conception selon ISO 13407. Pour le RUP©, l'évaluation correspond aux tests d'utilisabilité qui permettent de pallier les lacunes du concepteur d'interface et d'obtenir l'assentiment des parties prenantes.

3.3.3.1 ISO 13407 : conception et surveillance

Pour ISO (1999), l'évaluation des interfaces-utilisateurs est liée à leur conception, d'une part en début de projet, et, d'autre part, à un plan de surveillance à long terme pour la suite du projet et la durée de vie du système. Lors de la conception des solutions, les tests permettent d'obtenir les informations nécessaires à l'amélioration des interfaces. En fait, le processus consiste à produire des solutions de conception, et à l'aide de simulations, modèles ou maquettes, soumettre ces solutions aux utilisateurs afin que ceux-ci exécutent ou simulent l'exécution de tâches. Selon les

informations recueillies à cette occasion, les solutions de conception sont modifiées. L'évaluation doit donc être utilisée pour fournir un retour d'information, évaluer si les objectifs liés aux exigences ont été atteints et adapter l'utilisation du système.

L'évaluation doit faire l'objet d'un plan d'évaluation où l'on retrouve les objectifs de conception, les personnes responsables de l'évaluation, ce qu'il faut évaluer et comment les moyens nécessaires et l'implication des utilisateurs, la correspondance du plan avec l'échéancier du projet et les retours d'information ainsi que leur utilisation dans d'autres activités de conception (ISO, 1999).

3.3.3.2 RUP© : pallier les défaillances

Selon Rational (2002), « les tests d'utilisabilité ne sont pas un substitut à une bonne conception et ils sont plus efficaces lorsque combinés à une conception centrée sur l'utilisateur ».

Toujours selon le RUP©, les tests d'utilisabilité visent à évaluer le système selon la perspective de l'utilisateur final et comprennent :

- les facteurs humains,
- l'esthétique,
- la cohérence de l'interface-utilisateur,
- l'aide en ligne et l'aide contextuelle,
- les assistants et les agents,
- la documentation utilisateur,
- le matériel de formation (Rational, 2002).

Les tests consistent à « exposer » les prototypes aux parties prenantes. Un des principaux objectifs des tests est de pallier les défaillances du concepteur d'interface. Selon le RUP©, soumettre les utilisateurs à des tests d'utilisabilité permet d'obtenir l'approbation des parties prenantes et de corriger les mésinterprétations des besoins des parties prenantes (Rational, 2002). Les tests d'utilisabilité sont conçus par le concepteur de tests et exécutés par un testeur. Toujours selon le RUP©, les interfaces-utilisateurs doivent faire l'objet de tests d'utilisabilité dès que les premières maquettes sont disponibles (Rational, 2002).

Le RUP© recommande de soumettre les interfaces-utilisateurs pour évaluation aux autres membres du projet, à des experts de l'utilisabilité externes et aux utilisateurs. Les types d'évaluation avec les utilisateurs sont le pas-à-pas et les tests d'utilisabilité formels effectués avec des simulations de tâche (Rational, 2002).

3.4 Conception pluridisciplinaire

Selon le quatrième et dernier principe d'ISO 13407, la conception centrée sur l'utilisateur doit se fonder sur différents champs de connaissance, notamment celui de l'ergonomie selon ISO 13407. Selon le RUP© les concepteurs exercent des rôles interchangeables quoique la conception d'interface-utilisateur puisse faire appel aux qualification de graphistes et l'évaluation, à des experts externes.

3.4.1 ISO 13407 : facteurs humains et ergonomie

« Il existe une grande quantité de connaissances scientifiques et théoriques sur l'ergonomie, la psychologie, les sciences cognitives, la conception de produits et autres disciplines pertinentes, qui peuvent suggérer des solutions potentielles de conception » (ISO, 1999). Pour ISO (1999), la conception centrée sur l'utilisateur est « une activité pluridisciplinaire faisant appel aux connaissances et techniques du domaine des facteurs humains et de l'ergonomie ».

Quelle que soit sa taille, l'équipe de conception doit être suffisamment diversifiée pour que « les compromis de conception soient décidés de manière appropriée » (ISO, 1999). Elle doit refléter « les relations entre l'organisme responsable du développement et le client » (ISO, 1999). On peut donc y voir représentée une diversité de domaines de compétences et de points de vue où peuvent être représentées les fonctions énumérées au tableau 3.7.

Tableau 3.7 : Fonctions représentées dans les équipes pluridisciplinaires (ISO 1999)

Fonctions	
• Utilisateur final • Spécialiste du domaine de l'application, analyste économique • Responsable marketing, vendeur • Expert en facteurs humains et ergonomie, spécialiste de l'interaction humain-machine	• Acheteur, le supérieur hiérarchique de l'utilisateur • Analyste de systèmes, ingénieur système ou programmeur • Concepteur d'interface-utilisateur, concepteur graphique • Auteur des documents techniques, personnel de formation et d'assistance technique

La taille de l'équipe pluridisciplinaire de conception peut être petite ou grande selon la taille et le type de produit. Ainsi, les nouveaux produits peuvent requérir une équipe complète alors que, pour des produits d'importance réduite, des produits issus de versions précédentes ou ciblés à des marchés restreints, par exemple, peuvent nécessiter des équipes moins importantes au sein desquelles les membres assument plusieurs rôles à la fois (ISO, 1999).

3.4.2 RUP© : travailleurs, rôles et qualification

Selon le RUP©, le système est réalisé par des « travailleurs ». Un travailleur joue un rôle comme un acteur dans une pièce de théâtre. Ainsi, une personne peut jouer plusieurs rôles. La figure 3.4 illustre cette approche.

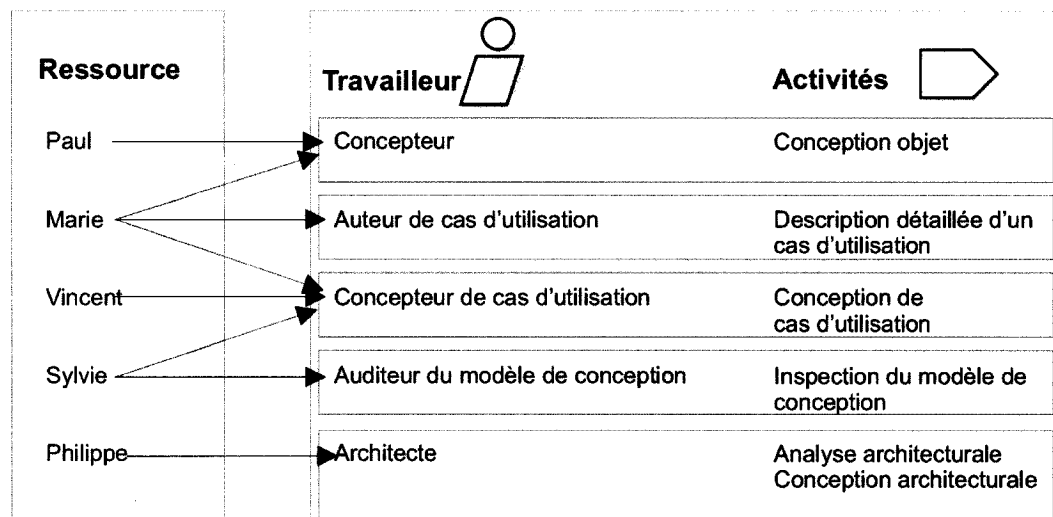


Figure 3.4 : Ressources et travailleurs (Kruchten, 2001).

Néanmoins, le RUP© identifie les qualifications que chacun des travailleurs devrait avoir. Selon la feuille de route de l'ingénierie de l'utilisabilité, l'analyste de processus métier, l'analyste système et le spécificateur de cas d'utilisation doivent être qualifiés pour recueillir l'information sur les utilisateurs, leurs tâches et leur environnement afin d'intégrer cette information aux artefacts de modélisation de métier et d'exigence. Le tableau 3.8 dresse la liste de ces travailleurs et de leurs qualifications.

Selon Kruchten (2001), le concepteur d'interface-utilisateur « dirige et coordonne le prototypage et la conception de l'interface-utilisateur. Plus précisément, il collecte les exigences concernant

l'interface-utilisateur, y compris les exigences de facilité d'utilisation, et construit les prototypes correspondants ; il fait participer les utilisateurs finaux aux revues au moment de vérifier la facilité d'utilisation et aux sessions de test d'utilisation. Il contrôle et commente l'implémentation finale de l'interface-utilisateur. »

Tableau 3.8 : Travailleurs et conception d'interface-utilisateur (Rational, 2002, Kruchten, 2001).

Travailleur	Rôle	Qualifications
Analyste de processus métier	Dirige et coordonne la modélisation des cas d'utilisation-métier	- Facilitateur - Habilités de communicateurs - Connaissance du domaine d'affaires.
Analyste système	Dirige et coordonne l'expression des exigences et la modélisation des cas d'utilisation.	- Identification et compréhension des problèmes et des opportunités. - Définir les besoins associés à ces problèmes et opportunités.
Spécificateur de cas d'utilisation	Détaille les spécifications d'une partie des fonctionnalités du système en décrivant un ou plusieurs cas d'utilisation.	- Habilités de communicateur, tant orales qu'écrites. - Connaissance du domaine d'affaire et de la technologie - Familier avec les outils utilisés de gestion des exigences.
Auditeur du cahier des charges	Planifie et dirige les revues formelles du modèle de cas d'utilisation.	- Connaissance du domaine ou expertise appropriée en gestion des exigences. - Habilités à produire un cahier des charges ou responsabilité d'autres artefacts en lien avec le cahier.
Auditeur du modèle métier	Participe aux revues formelles du modèle des cas d'utilisation-métier et du modèle objet métier.	- Connaissance du domaine ou expertise appropriée au modèle des cas métier. - Habilités à produire un modèle des cas métier.
Concepteur d'interface-utilisateur	- Dirige et coordonne le prototypage et la conception de l'interface-utilisateur. - Collecte les exigences concernant l'interface-utilisateur - Fait participer les utilisateurs finaux aux revues et aux sessions de test d'utilisation. - Contrôle et commente l'implémentation finale de l'interface-utilisateur.	Peut provenir du domaine de la création du domaine des arts plutôt que de celui des affaires, du génie ou de l'informatique.
Implémenteur	- Développe et teste des composants avant leur intégration dans des sous-systèmes plus importants. - Responsable du développement et du test des composants de test et des sous-systèmes correspondants.	- Connaissance du système ou de l'application en construction. - Familier avec les tests et les outils de test automatique - Qualifications en programmation.
Testeur	Responsable de l'exécution des tests.	- Connaissance des approches et techniques de test - Habilités à diagnostiquer et à résoudre des problèmes - Connaissance du système ou de l'application testée - Notions de fonctionnement de réseau et d'architecture de système - Formation et expérience des tests automatiques - Qualifications en programmation
Concepteur de test	- Responsable de l'identification et de la définition des tests requis, du suivi de la progression des tests, des résultats de chacun des cycles de test et de l'évaluation de l'ensemble de la qualité qui découle des activités de test. - Responsable de représenter adéquatement les besoins des parties prenantes	- Aptitudes à l'analyse et à la recherche - Prêt à relever des défis, - Compréhension des pannes et défaillances logicielles - Connaissance du domaine et du système où l'application testée - Expérience de l'application de plans de test.

On prévoit cependant faire appel à un expert externe en utilisabilité pour identifier les problèmes d'utilisabilité qui, selon son expérience, sont les plus communs. Toujours selon le RUP© (2002), il peut être profitable de faire appel à eux tôt dans le processus de conception d'interface-utilisateur afin de refaire une conception qui incorpore leurs recommandations.

Ainsi, le RUP© voit le rôle de l'expert en utilisabilité comme celui qui, à partir de ce qui équivaut à des évaluations heuristiques, diagnostique des problèmes et permet leur résolution.

Le RUP© recommande que, pour les grands projets, les concepteurs d'interface-utilisateur soient groupés dans une équipe distincte. Cela est important parce que les qualifications doivent être fréquemment améliorées et optimisées pour le projet en cours. Le projet a des exigences d'utilisabilité qui lui sont propres et qui requièrent temps et concentration. C'est aussi important parce qu'il y a un risque de « double allégeance », c'est-à-dire que le concepteur d'interface-utilisateur doit être guidé d'abord par des considérations d'utilisabilité plutôt que d'implantation (Rational, 2002).

3.4.3 Comparaison entre ISO 13407 et le RUP©

Le tableau 3.9 récapitule les différences et les similarités de la conception centrée sur l'utilisateur selon ISO 13407 et le RUP©.

Tableau 3.9 : Tableau comparatif de ISO 13407 et du RUP©.

Principe	ISO 13407	RUP©
Participation active des utilisateurs et compréhension claire des exigences liées à l'utilisateur et à la tâche		
Participation des utilisateurs	• Obligatoire à toutes les étapes • Variée selon les activités	• Facultative, mais préférable lors de la conception d'interface-utilisateur • Atelier de travail mais peut être variée
Contexte d'utilisation		
Utilisateur	• Description élaborée	• Décrits dans une description à haut niveau du système • Acteurs
Environnement	• Description élaborée	• Défini dans une description à haut niveau du système
Tâche	• Analyse de la tâche	• Cas d'utilisation qui décrit le système et qui est dérivé des fonctionnalités et des processus d'affaires
Répartition appropriée des fonctions entre les utilisateurs et la technologie	• Répartition des fonctions • exigences-utilisateurs	• Pas de répartition des fonctions • Exigences d'automatisation • Exigences d'utilisabilité
Itération des solutions de conception		
Produire les solutions de conception et les présenter aux utilisateurs	• Contexte d'utilisation • Autres connaissances	• Normes et guides
Gérer l'itération des solutions de conception	• Évaluations avec utilisateur • Planifié • Intégré à la conception	• Non planifié • Peut être utile lorsque besoins changeants ou problème complexe
Évaluer les conceptions par rapport aux exigences	• Liée à la conception • Surveillance à long terme • Planifiée • Plusieurs types	• Accord des parties prenantes • Pallier les défaillances • Plusieurs types • Participation d'expert
Conception pluridisciplinaire	• Obligatoire • Participation d'expert en facteur humain • Équipe distincte	• Facultatif, mais recommandé • Graphiste facultatif • Équipe distincte pour grand projet

CHAPITRE IV

MÉTHODE

Le présent chapitre comprend les objectifs de la recherche et les questions de recherche. Ces dernières sont au nombre de six et correspondent aux principes ISO 13407 lors de la comparaison faite entre la norme et le RUP© dans le chapitre III. Le protocole de recherche et le format de présentation de l'étude de cas sont aussi décrits ici.

4.1 Les objectifs de la recherche

L'objectif principal de la recherche consiste donc à déterminer comment les principes de la conception centrée sur l'utilisateur, définis par la norme ISO 13407, sont appliqués lors du développement d'un système informatique interactif encadré par le RUP©.

Plus spécifiquement, l'étude vérifie comment le RUP© se conforme aux principes de conception prescrits par la norme ISO 13407, si l'application du RUP© qu'en a faite l'équipe du programme est une conception centrée sur l'utilisateur et comment un processus de conception centrée sur l'utilisateur a été exécuté dans le contexte d'un développement encadré par le RUP©.

Le cas étudié est une évaluation d'une conception centrée sur l'utilisateur faite dans le cadre du RUP© lors du développement d'un système informatique de gestion d'entreprise. Pour y arriver, une comparaison sera donc établie entre :

- l'approche de conception centrée sur l'utilisateur, prescrite par la norme ISO 13407 ;
- les prescriptions de conception d'interfaces-utilisateurs formulées par le RUP© ;
- l'interprétation et l'application qu'en a faites l'équipe du programme des prescriptions du RUP© ;
- l'intervention d'ergonomes durant le projet pour réaliser la conception d'interfaces.

La recherche identifie les principaux principes et activités de la conception centrée sur l'utilisateur prescrits par la norme ISO 13407. Elle décrit comment les prescriptions du RUP© reprennent ces principes et activités. Elle inventorie les activités, les artefacts, les normes et les règles que l'équipe du programme a retenus du RUP© pour le développement du système et vérifie s'ils sont conformes à la norme ISO 13407. Enfin, elle décrit le processus qui a été utilisé

par les ergonomes pour la conception centrée sur l'utilisateur. La description de l'intervention des ergonomes permet de souligner ces différences et indique des pistes pour ajuster le RUP© à la conception centrée sur l'utilisateur.

4.2 Les questions de recherche

Les unités d'analyse, telles que définies par Yin (1989) pour fournir un cadre méthodologique à l'analyse de cas, seront déterminées par les questions de recherche. Les questions visent à vérifier si les principes de conception centrée sur l'utilisateur, fixés par la norme ISO 13407, sont respectés par le RUP© et appliqués par l'équipe du programme et comment ils ont été traités lors du cycle ergonomique. Les six questions de recherche retenues correspondent aux principes et activités qui ont servi au chapitre précédent à comparer la norme ISO 13407 avec le RUP©. Chacune porte sur l'un des principes, sauf trois d'entre elles qui portent sur le principe de l'itération des solutions de conception. Ce principe est scruté plus à fond. Outre la gestion des itérations, la matérialisation des solutions de conception et l'évaluation des solutions sont aussi étudiées.

4.2.1 Participation active des utilisateurs et compréhension claire des exigences liées à l'utilisateur et à la tâche

La participation active des utilisateurs est nécessaire. La compréhension des exigences liées à l'utilisateur et à la tâche est tributaire de la connaissance du contexte d'utilisation. Le contexte d'utilisation comprend la connaissance des utilisateurs, de l'environnement et de la tâche.

(1) Est-ce que l'application du RUP© soutient la participation active des utilisateurs et définit le contexte d'utilisation selon ISO 13407 ?

4.2.2 Répartition appropriée des fonctions entre les utilisateurs et la technologie

La répartition appropriée des fonctions entre les utilisateurs et la technologie se traduit par la formulation des exigences et s'inspire du contexte d'utilisation.

(2) L'application du RUP© permet-elle une bonne spécification des exigences-utilisateurs notamment par une bonne répartition des fonctions entre les utilisateurs et la technologie selon ISO 13407 ?

4.2.3 Itération des solutions de conception

L'itération suppose une production de solution de conception, une gestion des itérations et une évaluation des solutions. En d'autres termes, il faut produire des prototypes, planifier leur présentation aux utilisateurs et aux parties prenantes et s'assurer de leur évaluation pour faire évoluer les solutions de conception. Trois activités importantes liées à ce principe sont retenues auxquelles sont associées à chacune une question de recherche.

4.2.3.1 Produire les solutions de conception

À l'aide de l'information tirée du contexte d'utilisation, des qualifications de concepteurs et de la participation des utilisateurs et autres parties prenantes, des solutions, des maquettes et des prototypes doivent être produits, puis être évalués. La réalisation des solutions de conception est indissociable de la participation des utilisateurs. La simulation de tâches avec des maquettes facilite la communication entre les utilisateurs et les parties prenantes et favorise la qualité de la spécification fonctionnelle.

(3) La matérialisation des solutions de conception se fait-elle par l'application du RUP© selon ISO 13407 ?

4.2.3.2 Gérer l'itération des solutions de conception

Le type d'itérations recommandé par le RUP©, l'interprétation de l'équipe du programme et la planification d'un processus itératif par les ergonomes comme le prévoit la norme sont évalués.

(4) L'approche itérative de ISO 13407 est-elle respectée lors de l'application du RUP© ?

4.2.3.3 Évaluer les conceptions par rapport aux exigences

La nature des tests et des évaluations, notamment l'implication des utilisateurs et des parties prenantes, est déterminante. Ces évaluations permettent notamment d'obtenir un retour d'information sur la solution proposée et de juger si les objectifs de l'organisation ont été atteints.

(5) L'évaluation des solutions de conception retenue lors de l'application du RUP© est-elle conforme à celle qui est prescrite par ISO 13407 ?

4.2.4 Conception pluridisciplinaire

La conception pluridisciplinaire prescrite par ISO 13407 suppose l'intervention de personnes qualifiées.

(6) L'application du RUP© reconnaît-elle la nécessité d'un personnel qualifié selon ISO 13407 ?

4.3 Protocole de la recherche

Une étude de cas est une recherche empirique qui examine un phénomène contemporain dans un contexte de vie réelle lorsque les limites entre ce phénomène et le contexte ne sont pas clairement définies et lorsque plusieurs sources de preuve sont utilisées (Yin, 1989). Par ailleurs, ce dernier précise qu'une étude de cas unique peut être utilisée pour tester une théorie afin de déterminer si les propositions que soutient cette théorie sont correctes ou s'il y a des alternatives. On peut aussi y recourir pour étudier un phénomène jusque-là ignoré (Gagnon, 2005). On peut étendre les conclusions d'une étude de cas par « généralisation analytique » (Yin, 1989). Wallace et Zelkowitz (1998) identifie l'analyse de cas comme une méthode de recherche pertinente à l'étude du développement logiciel.

Lamoureux (1992) identifie quatre catégories de méthodes de recherche en sciences humaines : les méthodes exploratoires, descriptives, explicatives et d'intervention. Yin (1989 et 2003) affirme que les méthodes exploratoires, descriptives et explicatives sont applicables à l'étude de cas.

Le protocole de recherche est inspiré des règles définies pour la conception d'étude de cas par Robert K. Yin (1989). L'approche est l'étude de cas unique. La méthode de cueillette de données descriptive est privilégiée. L'approche de comparaison de cas de l'analyse comparative permet de comparer les descriptions et explications de cas différents ou du même cas (Yin, 1989 et Cunningham, 1997).

Hlady Rispal (2002) décrit l'analyse comparative comme une démarche qui consiste à comparer les classes de données qui correspondent aux unités d'analyse « à plusieurs reprises dans une perspective temporelle, d'examen de l'évolution du processus à l'étude ». Ces données sont ensuite comparées à la littérature théorique (Hlady Rispal, 2002) qui correspond ici aux dispositions du RUP© et de la norme ISO 13407.

La période étudiée porte sur une durée de vingt semaines. Cette période débute lorsque la phase d'inception est terminée et la phase d'élaboration est en cours. La modélisation d'affaires du projet est terminée et la définition des exigences, notamment celle des cas d'utilisation, est en cours.

L'auteur de l'étude est intervenu pour appliquer le cycle ergonomique selon les principes de la conception centrée sur l'utilisateur. Il n'a pas influencé l'adaptation faite du processus RUP© par l'équipe du programme pour la définition des exigences, en général, et la conception des interfaces-utilisateurs, en particulier. Cette adaptation avait été faite avant son arrivée.

4.3.1 Les procédures de terrain

L'analyse de contenu et la méthode *ex post facto* telles que définies par Lamoureux (1992) seront utilisées pour l'analyse des données. La connaissance du cas a été au préalable approfondie puisque l'intervention ergonomique, par sa nature, a nécessité la meilleure compréhension possible des processus et des techniques utilisés.

La décision de faire du projet l'objet d'une recherche a été prise après l'intervention. Ainsi, la demande d'utilisation des données a été faite après la période étudiée. Le chercheur n'a donc pas pu influencer le cours du projet pour les besoins de sa recherche. Les autres intervenants n'ont donc pas été influencés par le statut de chercheur de l'auteur.

4.3.2 L'accès au site

L'auteur a été un intervenant actif dans le projet de développement. En agissant comme ergonomiste responsable de la conception des interfaces-utilisateurs du projet, l'accès aux informations pertinentes était dans l'exercice de sa fonction et lui a été accordé après l'exercice de sa fonction. L'accès au site après l'intervention a permis la consultation de tous les documents pertinents à la présente recherche et l'accès à tous les intervenants concernés, lorsque requis. L'accès aux informations a été d'autant facilité que c'est le client lui-même qui a retenu les services du chercheur.

4.3.3 Confidentialité

L'accès aux données est conditionnel à la confidentialité de l'identification de l'organisation étudiée et de la nature de ses opérations. Certaines données sur l'ampleur du projet, sa

planification et son budget sont aussi confidentielles. Le respect de cette confidentialité impose que la présentation de certaines données est faite de façon à les rendre anonymes.

4.3.4 Les sources de données

La collecte de données s'est faite d'abord par l'observation participante, des entrevues et la consultation de documents. L'observation participante a été faite par l'auteur de l'étude.

L'intervention des ergonomes est survenue une fois que la modélisation d'affaire, la définition des exigences fonctionnelles et celle du modèle des cas d'utilisation ont été complétées. Ces activités n'ont donc pas été influencées par les ergonomes. L'intervention des ergonomes s'est terminée alors que l'enchaînement des activités d'analyse et de conception, tel que défini par le RUP©, démarrait. Le chercheur n'a donc pas eu d'impact sur ces aspects des données.

Les principales sources de données correspondent aux documents qui décrivent chacun des processus et les biens livrables qui en découlent.

Les types de documents consultés sont les suivants :

- les biens livrables, produits par les travailleurs de l'équipe du programme : 38 documents de 1 à 145 pages ;
- les bulletins internes du projet, produits par la direction du projet : 9 documents de 2 pages ;
- les comptes à rendus de rencontres faites avec différents intervenants : 6 documents de 3 à 4 pages ;
- un documents de gestion de la direction du projet : 1 document de 27 pages ;
- les gabarits de document, produits par l'équipe des façons de faire : 2 documents de 5 à 8 pages ;
- les guides, produits par l'équipe des façons de faire : 4 documents de 9 à 28 pages ;
- les lignes directrices, produits par l'équipe des façons de faire : 8 documents de 5 à 25 pages ;
- les planifications, soumises durant le projet : 2 documents de 4 à 9 pages ;
- les présentations, faites par l'équipe d'ergonomie : 2 documents de 17 à 18 pages ;

- un rapport : 1 documents de 7 pages ;
- les rapports annuels de l'entreprise, publiés entre 2002 et 2005 : 4 documents de 64 à 78 pages ;
- les rapports hebdomadaires, produits par les ergonomes : 1 document de 19 pages ;
- les recommandations, soumises pour modifier les façons de faire : 1 document de 8 pages.

La liste exhaustive des sources de données consultées pour l'étude de cas apparaît aux annexes I et II à la fin de cet ouvrage.

4.4 Le format de présentation de l'étude de cas

Pour une étude de cas unique, plusieurs types de présentation sont possibles. Parmi celles-ci, une analyse, faite à partir des questions de recherche, est une approche inductive qui permet d'approfondir la compréhension (Hlady Rispal, 2002). La structure retenue pour la présentation de l'étude de cas est la structure comparative, soit la comparaison des descriptions et explications comparables (Yin, 1989). Ces descriptions et explications sont ici les unités d'analyse définies par les questions de recherche. Le traitement fait par chaque approche est décrit.

Le chapitre V décrit le cas afin d'en identifier le contexte. Le chapitre VI interprète les données en les présentant, pour chaque question de recherche, sous l'angle de l'application faite par l'équipe du programme et de celui de l'intervention des ergonomes.

Le chapitre VII, la discussion, identifie des explications et des tendances qui ressortent des données présentées dans le chapitre précédent en les confrontant avec les analyses faites au chapitre III des différences et similitudes entre le ISO 13407 et RUP© sur la conception centrée sur l'utilisateur. Ainsi, pour chaque question de recherche ISO 13407, le RUP©, l'application faite par l'équipe de programme et l'intervention ergonomique seront tenus en compte.

Les chapitre V et VI sont annotés pour identifier la source des données. Afin de respecter la confidentialité des informations, certaines données sont maquillées ou partiellement présentées. Les sources de données sont identifiées par des titres génériques. Les auteurs n'apparaissent pas.

CHAPITRE V

DESCRIPTION DU CAS

Voici le contexte dans lequel se situe l'étude. Il identifie l'entreprise et le système étudié. Il précise comment le RUP© a été adapté pour le développement du système, plus spécifiquement, pour les disciplines de modélisation de métier et de définition des exigences. La nature et l'ampleur du système sont ensuite décrites. Enfin, la façon dont les spécialistes de la conception centrée sur l'utilisateur se sont acquittés de leur mandat est exposée. Cela permettra de mettre en perspective l'application faite du RUP© et de la conception centrée sur l'utilisateur.

5.1 Le contexte : l'entreprise et le système à développer

La conception centrée sur l'utilisateur a été étudiée lors d'un projet informatique visant à réaliser un système de gestion d'opérations d'une entreprise. L'entreprise et le système à développer sont ici sommairement décrits. Les ressources humaines mobilisées pour le projet, et plus précisément celles qui sont affectées à la modélisation d'affaires et à la spécification des exigences, y sont présentées.

L'entreprise compte plus de 7 000 employés¹. Son bénéfice annuel dépasse 1,5 G\$ CA avec des revenus totaux de près de 4 G\$ CA². Le système à développer a pour mission d'assurer la gestion des opérations d'une des principales activités de l'entreprise. Cette activité est la vente de produits distribués dans un réseau de commerces de vente au détail³. Elle compte pour 1,8 G\$ des revenus de l'entreprise et mobilise plus de 800 employés⁴. Le système est développé dans le cadre du programme « Neptune »

Le système du programme Neptune est une consolidation de plus d'une quinzaine de systèmes existants auxquels il faut ajouter une multitude d'applications produites à partir de logiciels de bureautiques utilisées pour consulter une base de données centrale, des feuilles de calcul et autres opérations qui ne sont pas automatisées⁵. On estime à 200 le nombre d'employés appelés à utiliser l'ensemble du système une fois qu'il sera opérationnel⁶.

Un inventaire des besoins d'affaires a permis d'identifier 133 activités d'affaires regroupées en 13 domaines d'affaires⁷. Pour les besoins du développement, le système a été découpé en 12 de ces sous-domaines d'affaires qui font l'objet d'autant de projets pour l'ensemble du programme Neptune. Ainsi, l'un de ces projets portait sur la « Gestion financière » et un autre sur la

« Gestion des réseaux de distribution »⁸. Ces projets pouvaient eux-mêmes être découpés en sous-projets pour atténuer les risques. L'étude de cas porte sur le processus de spécification des interfaces-utilisateurs élaboré lors de la définition des exigences de l'application « Gestion des réseaux de distribution »⁹.

Initialement, un le processus prévoyait qu'un projet devait lui-même être découpé en itérations. Une itération devait correspondre à une sélection d'un certain nombre de cas d'utilisation de l'application qui devaient être réalisés de bout en bout. Cette approche n'a pas été appliquée. Par contre, une itération 0, prototype exécutable d'une partie du sous-système de la « Gestion des réseaux de distribution » a été livrée pendant la définition des cas d'utilisation du projet. Cette itération correspondait plutôt à une preuve de concept qui a été présentée à l'ensemble des membres du projet¹⁰.

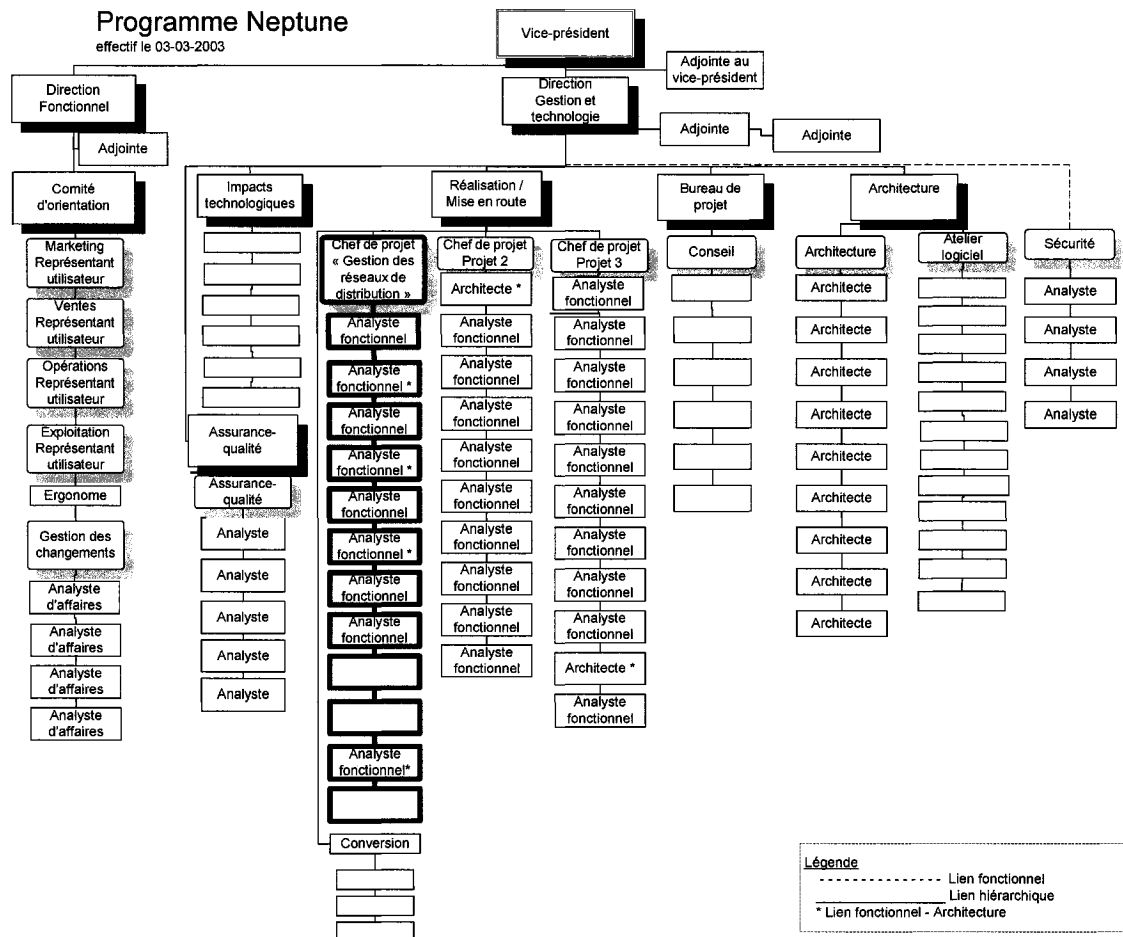
Le personnel affecté à l'ensemble du programme Neptune compte plus d'une centaine de personnes¹¹. À ces effectifs, le service des ressources humaines prévoyait ajouter 100 autres personnes¹². Le tableau 5.1 ventile l'affectation des effectifs du programme.

Tableau 5.1 : Distribution des effectifs du programme Neptune¹³.

Service	Personnel affecté
Direction fonctionnelle	12
Impacts technologiques	6
Assurance qualité	6
Projet Gestion des réseaux de distribution	13
Conversion	4
Projet 2	11
Projet 3	12
Bureau de projet	7
Architecture	11
Atelier logiciel	14
Sécurité	5
Gestion et technologie	8
Vice-présidence	2
Total	111

Les utilisateurs et leurs représentants comptent sur une équipe structurée avec des ressources appropriées. Les analystes d'affaires et les ergonomes font partie de cette équipe. Ces analystes d'affaires et les représentants des utilisateurs relèvent d'une direction appelée direction fonctionnelle. Toutes les autres équipes appartiennent à un service appelé gestion et technologie.

Ces deux directions relèvent du premier vice-président de l'entreprise¹⁴. L'organigramme présenté à la figure 5.1 illustre cette organisation des effectifs.



L'organigramme reflète l'organisation du programme à l'étape de la modélisation d'affaires et de la définition des exigences. Les effectifs nécessaires à l'analyse et la conception ainsi qu'à la réalisation n'étaient pas encore recrutés lors de la période de l'étude de cas¹⁶.

5.2 Le processus de développement utilisé

Le processus retenu pour le développement de l'application est une adaptation du processus RUP© faite pour les besoins du programme. Certaines appellations sont modifiées. Ainsi, la

phase d'inception devient la phase d'opportunité. La discipline de la modélisation de métier devient celle de la modélisation d'affaires et la discipline des besoins devient celle de la définition des exigences. Le tableau 5.2 donne les équivalences pour le RUP© en français et en anglais ainsi que l'adaptation qui en a été faite.

Tableau 5.2 : Équivalences des termes utilisés par le RUP© et son adaptation¹⁷.

Français	RUP©	Anglais	Adaptation
Phase			
Inception	Inception		Opportunité
Enchaînement			
Modélisation de métier	Business Modeling		Modélisation d'affaire
Besoins	Requirements		Définition des exigences
Artefacts de la modélisation de métier			
Glossaire	Business Glossary Business Rules		Dictionnaire des termes d'affaires Règles d'affaires
Modèle des cas d'utilisation-métier	Business Use Case Model		Diagrammes de processus opérationnels
Modèle objet métier	Business Object Model		Modèle de concepts clefs
Principes et conseils pour l'interface utilisateur	User-Interface Guidelines		Cadre de conception des interfaces Web
Artefacts des besoins			
Vision	Product Features (Vision)		Caractéristiques logicielles fonctionnelles
Spécifications supplémentaires	Supplementary Specifications		Caractéristiques logicielles non fonctionnelles
Document d'architecture logicielle (Vue logique)	Software Architecture Document (Logical View)		Architecture applicative
Modèle des cas d'utilisation	Use-Case Model		Modèle des cas d'utilisation
Cas d'utilisation	Use-Case		Cas d'utilisation
Prototype de l'interface-utilisateur	User-Interface Prototype		Scénarimages

Une équipe dont le nom est « Les façons de faire » est responsable de la discipline de l'environnement définie par le RUP©. Cette discipline consiste à assister l'équipe du programme en adaptant le processus de développement au projet et en les outils de travail qui l'aident à réaliser les activités et les artefacts du processus (Kruchten 2001). Ainsi, les Façons de faire fournissent, pour la réalisation des artefacts :

- une fiche signalétique qui décrit les objectifs, les activités de production et de contrôle de la qualité, les intrants, les responsabilités des travailleurs et les outils pour réaliser un artefact ;
- un gabarit de l'artefact ;

- une guide d'utilisation du gabarit ;
- une formation pour expliquer comment réaliser l'artefact¹⁸.

Une équipe d'assurance-qualité est responsable de définir les activités de contrôle de la qualité pour chaque type artefact. La figure 5.2 illustre la démarche.

Les activités d'assurance-qualité

Objectifs:

S'assurer que les livrables respectent les normes établies et que l'application développée couvre les besoins exprimés par les utilisateurs. Les activités d'assurance-qualité sont sous la responsabilité de tous.



S'assurer que les pré-requis sont respectés. Doit être effectué avant le début de la production du livrable.

- ☐ S'assurer que l'on commence avec une série d'intrants (artefacts) ayant été soumis aux processus qualité.
- ☐ S'assurer que l'on commence avec la dernière version disponible des gabarits et des guides.
- ☐ S'assurer de la disponibilité des outils de travail.



Effectuer pour chaque activité :

- ☐ Auto-vérification.
- ☐ Vérification préliminaire du mentor.
- ☐ Vérification préliminaire par l'utilisateur.
- ☐ Vérification préliminaire du consommateur.



S'assurer que les activités AQL suivantes sont effectuées avant le dépôt du bien livrable

- ☐ Revue par les pairs.
- ☐ Revue AQL.
- ☐ Revue clients.
- ☐ Processus d'approbation du bien livrable
 - Revue d'approbation
 - Signature officielle

Figure 5.2 : Activités d'assurance-qualité¹⁹.

Elle assure aussi le suivi de la réalisation de ces activités. Ses membres participent aux revues par les pairs²⁰.

L'équipe du programme s'est donnée un modèle de développement qui apparaît à la figure 5.3.

Modèle de développement

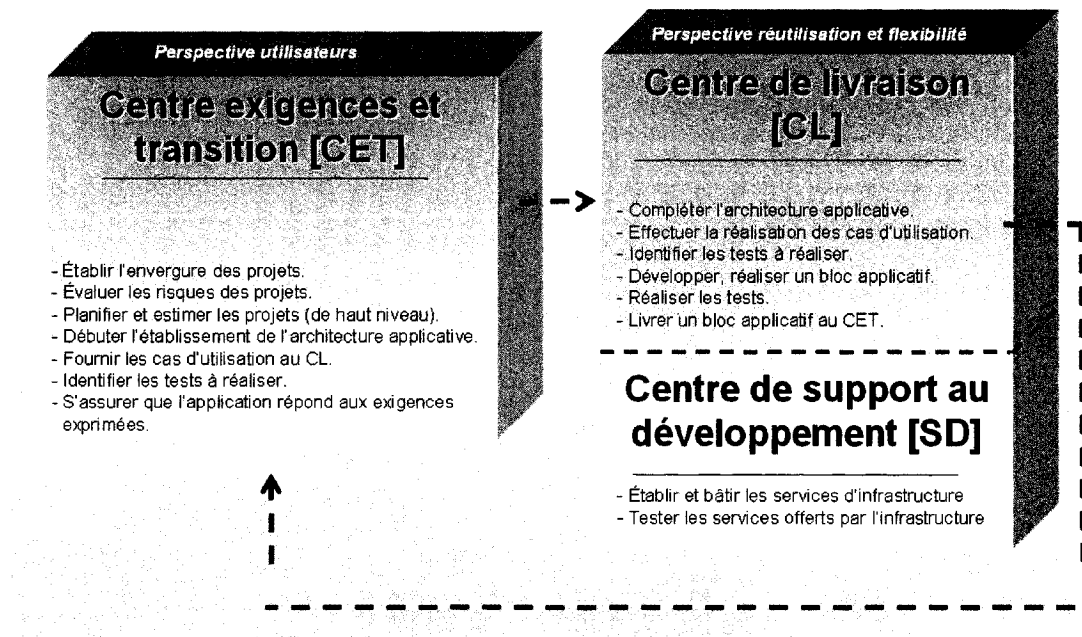


Figure 5.3 : Modèle de développement du projet²¹.

L'organisation est structurée en deux principales parties : un centre des exigences et transition (CET) et un centre de livraison (CL). Chaque centre est responsable d'enchaînements d'activité, sauf la discipline des tests qui est partagée par les deux. Le CET a pour tâche d'identifier les tests et le CL a celle de les réaliser. La figure illustre les responsabilités des deux centres.

On trouve les disciplines, dont est responsable le CET, à la figure 5.4 Ces disciplines sont la modélisation d'affaires, la définition des exigences et une partie de la discipline des tests.

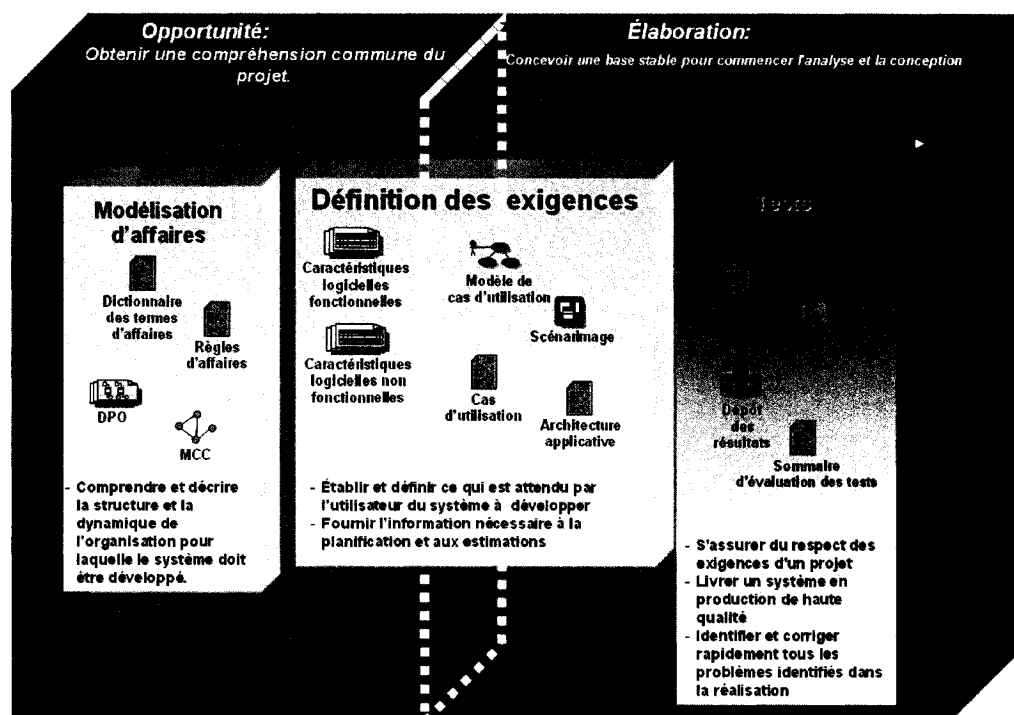
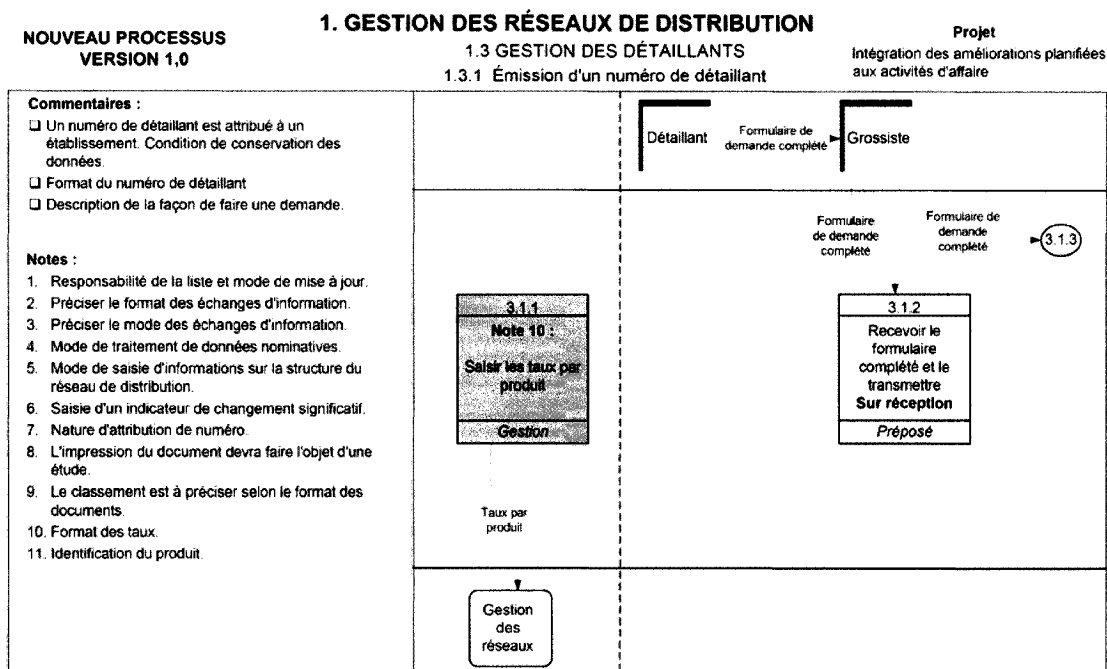


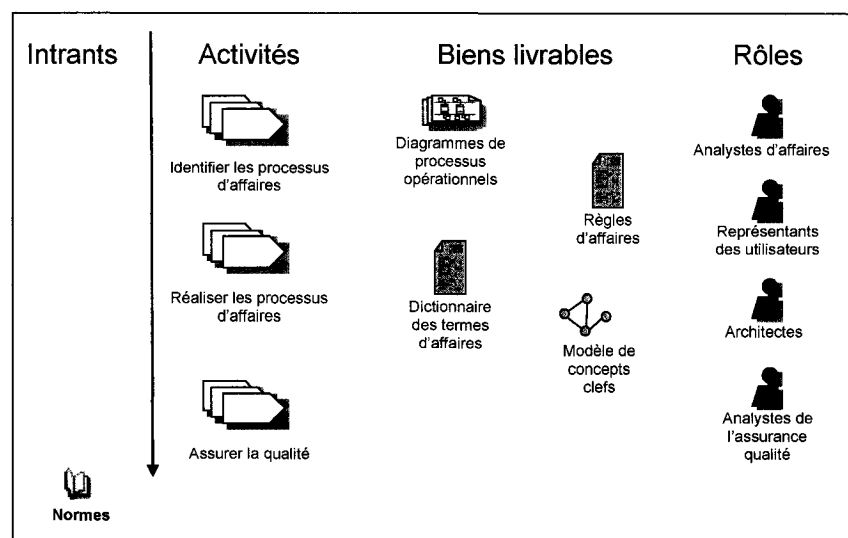
Figure 5.4 : Artefacts produits par le centre des exigences et transition²².

Les activités de la modélisation d'affaires se déroulent durant la phase d'opportunité. L'activité centrale de la modélisation d'affaires est la production des diagrammes de processus opérationnels (DPO) qui illustrent les processus d'affaires lorsque le nouveau système sera en opération. Ces diagrammes se font en deux étapes. On produit d'abord les diagrammes des processus d'affaires courants et, ensuite, les processus cibles. Ces derniers auront cours une fois le système implanté et servent d'intrants aux artefacts subséquents. Ils sont produits selon un formalisme propre à l'entreprise²³. La définition des exigences, qui se déroule lors des phases d'opportunité et d'élaboration, a la production des cas d'utilisation pour aboutissement principal.

Les DPO sont produits par des analystes d'affaires en collaboration avec les représentants des utilisateurs. La figure 5.5 illustre un exemple d'un DPO : « Émission d'un numéro de détaillant ».

Figure 5.5 : Exemple d'un DPO²⁴.

Les architectes sont responsables de la production du modèle des concepts clés (MCC)²⁵. La figure 5.6 illustre la discipline, les activités, les artefacts qui leurs sont associés et les travailleurs qui y contribuent.

Figure 5.6 : Adaptation du cheminement d'activités de la modélisation d'affaire²⁶.

Le dictionnaire des termes d'affaire et les règles d'affaires sont alimentés parallèlement aux DPO par les représentants des utilisateurs²⁷.

La définition des exigences se déroule en deux étapes. Les caractéristiques logicielles sont d'abord définies durant la phase d'opportunité et les cas d'utilisation sont produits durant la phase d'élaboration. La figure 5.7 illustre les activités, les artefacts et les rôles de ce cheminement d'activités.

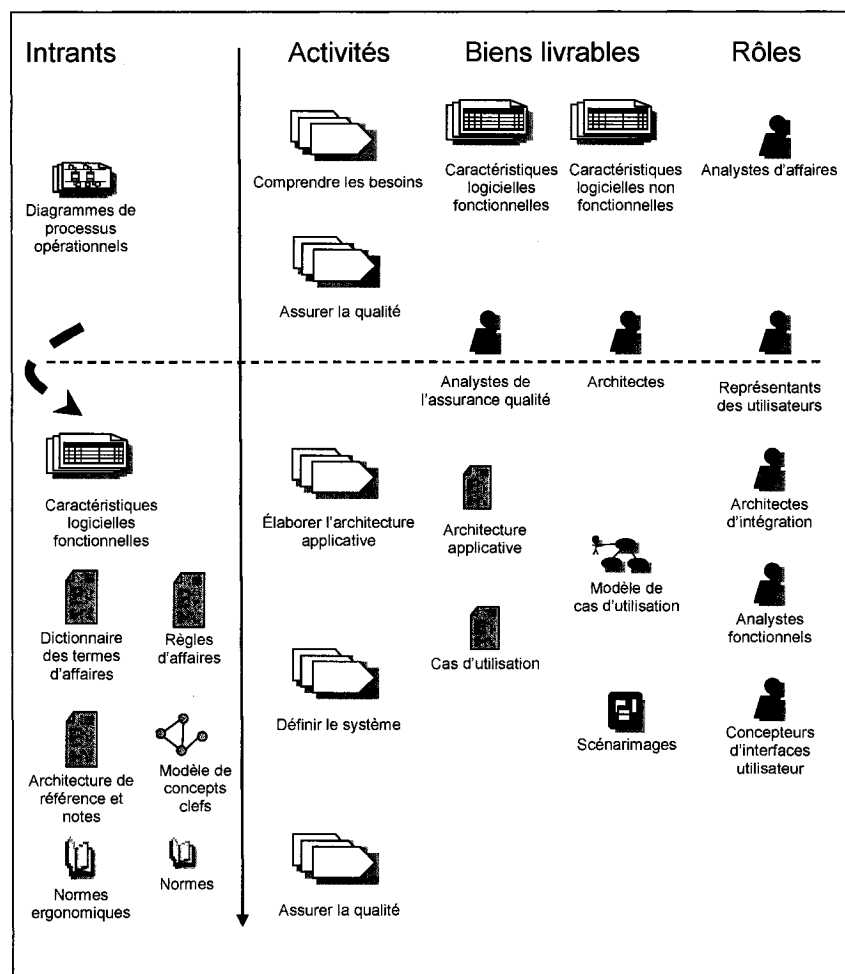


Figure 5.7 : Adaptation du cheminement d'activités de la définition des exigences²⁸.

Les caractéristiques logicielles fonctionnelles sont produites par les analystes d'affaires. Elles sont extraites des DPO²⁹. Outre une description, elles comportent un objectif, une priorité et une évaluation de la complexité à la réaliser³⁰. Les caractéristiques logicielles non fonctionnelles sont

produites par les analystes d'affaires³¹. Elles ont pour objectif, d'énumérer les exigences des parties prenantes soit celles des utilisateurs, de la sécurité, de l'exploitation, de la commercialisation et de l'architecture, qui ne sont pas décrites par les DPO, d'énumérer les contraintes imposées au système en développement et de fournir un point d'entrée pour la traçabilité³². Elles comportent notamment une description et une catégorie. Les catégories sont au nombre de 14. Une catégorie peut être, par exemple, l'intégrité ou la portabilité³³.

Le modèle des cas d'utilisation est élaboré par les analystes fonctionnels³⁴. La figure 5.8 donne un exemple de la présentation graphique UML d'un extrait de cas d'utilisation.

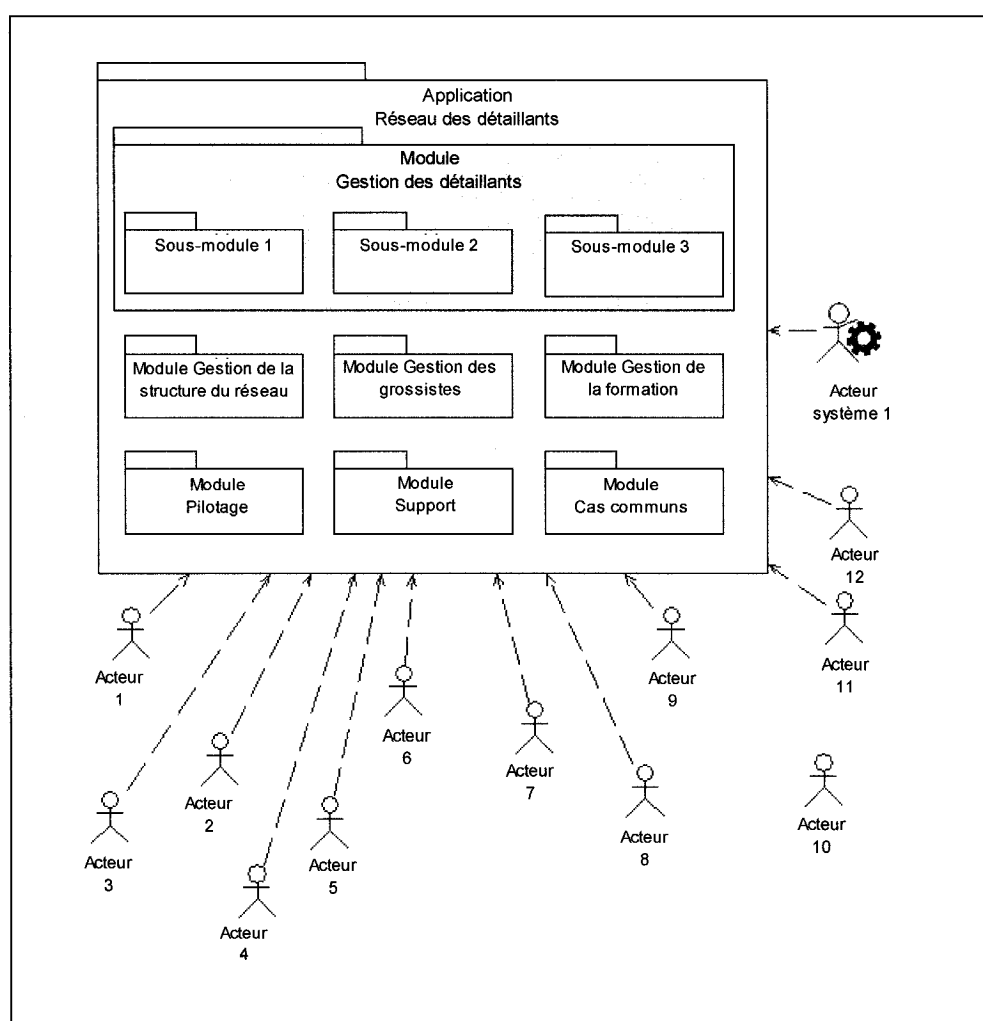


Figure 5.8 : Diagramme UML du modèle des cas d'utilisation de l'application³⁵.

Les intrants du modèle de cas sont le dictionnaire des termes d'affaires, les caractéristiques logicielles fonctionnelles, les diagrammes des processus opérationnels et le modèle des concepts clés³⁶. Pour l'équipe du programme, l'objectif du modèle des cas d'utilisation consiste à faire ressortir les fonctionnalités en identifiant ce qui va interagir avec le système, en définissant les limites du système, comment sont traduites les caractéristiques logicielles fonctionnelles et en définissant la structuration des fonctionnalités du système³⁷. Le modèle des cas comporte une présentation graphique des relations entre les cas d'utilisation et les acteurs, une courte description de chacun d'eux³⁸.

Ensuite, les cas d'utilisation sont eux-mêmes rédigés. Outre le nom des acteurs et la séquence d'actions réalisée par le système, qu'on appelle ici le déroulement du cas d'utilisation, on y inscrit les contraintes et exigences non fonctionnelles liées au cas ainsi que les règles d'affaire et les exigences de volume et de performance. On y inclut les valeurs par défaut proposées par le système et les messages d'erreurs³⁹. Le tableau 5.3 contient la table des matières d'un cas d'utilisation.

Tableau 5.3 : Gabarit de l'adaptation de l'artefact des cas d'utilisation⁴⁰.

1. INTRODUCTION
1.1. Application
1.2. Contexte du cas d'utilisation
1.3. Description sommaire
1.4. Envergure et limitations
2. CADRE DU CAS D'UTILISATION
2.1. Liste des acteurs
2.1.1. Acteurs primaires
2.1.2. Acteurs secondaires
2.2. Événements déclencheurs
2.3. Préconditions
2.4. Postconditions
2.4.1. Postconditions du cas en général
2.4.2. Postconditions internes
2.4.2.1. Post-conditions - <Nom du chemin principal>
2.5. Garantie minimale
3. DÉROULEMENT DU CAS D'UTILISATION
3.1. Chemin principal – <Nom du chemin principal>
3.1.1. Étapes du chemin principal
3.1.2. Chemins alternatifs
3.1.2.1. Chemin alternatif – <Nom du chemin alternatif>
4. VOLUME ET PERFORMANCE
5. ÉLÉMENTS DE SUPPORT AU CAS D'UTILISATION
5.1. Règles d'affaires
5.2. Messages
5.3. Domaines de valeurs

L'objectif du cas d'utilisation, pour l'équipe du programme, consiste à « définir le comportement du système, offrir une base contractuelle entre les utilisateurs et l'équipe de développement et à fournir une unité de base pour :

- la conception des interfaces utilisateur ;
- la gestion et le suivi du projet ;
- les interactions avec l'utilisateur ;
- la rédaction des cas de test ;
- la rédaction de la documentation ;
- l'élaboration de l'architecture applicative. »⁴¹

Les intrants sont le dictionnaire des termes d'affaires, les caractéristiques logicielles fonctionnelles, les DPO et le modèle des cas d'utilisation⁴². L'analyste fonctionnel est responsable du cas d'utilisation, mais le représentant des utilisateurs doit participer à l'établissement du détail de cas d'utilisation et doit l'approuver. Un architecte assure la coordination entre les cas d'utilisation et oriente leur rédaction⁴³.

Le scénarimage est un autre artefact de la phase d'élaboration de la définition des exigences⁴⁴. Cependant, il est prévu que sa production se poursuive lors de la discipline d'analyse et de conception lors de la phase de construction⁴⁵. Le scénarimage a pour objectifs d'offrir une base contractuelle pour les interfaces-utilisateurs, les écrans ou rapports, qui supportent un cas d'utilisation, de définir la navigation entre les interfaces-utilisateurs du cas, de déterminer les informations rendues disponibles par l'interface et de fournir une maquette des interfaces-utilisateurs⁴⁶.

Le tableau 5.4 reprend la table des matières d'un scénarimage.

Tableau 5.4 : Gabarit de l'adaptation du scénarimage⁴⁷.

1. INTRODUCTION
2. DIAGRAMME DE NAVIGATION
3. INTERFACES
3.1. Interface – <Nom de l'interface>
3.2. Éléments de données
3.3. Règles de présentation
4. RAPPORTS
4.1. Rapport – <Nom du rapport>
4.2. Éléments de données

Le diagramme de navigation est un diagramme d'activités selon le formalisme UML. Le diagramme d'activités est un type de diagramme d'état. Les activités sont ici des écrans, des rapports ou des connexions qui sont des liens vers d'autres diagrammes. L'activité correspond alors à l'exécution d'une instruction, d'une procédure ou au déroulement d'une opération d'un cas d'utilisation. La figure 5.9 est un exemple de diagramme d'activités qui sert de diagramme de navigation pour les scénarimages.

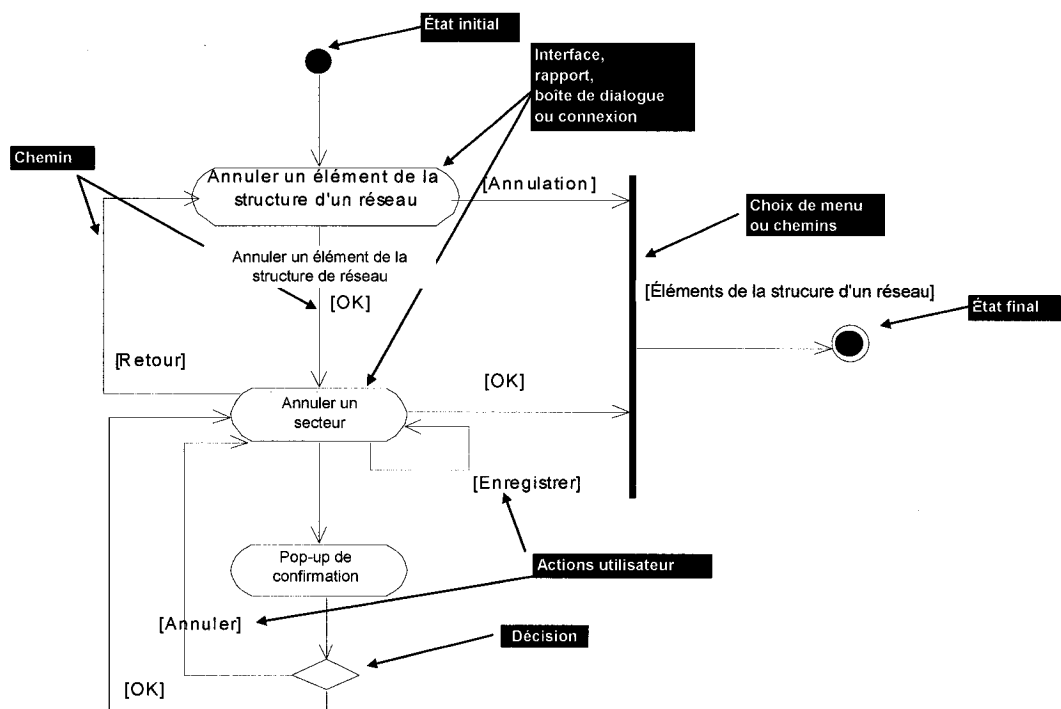


Figure 5.9 : Exemple d'un diagramme de navigation⁴⁸.

Les scénarimages sont produits par les analystes fonctionnels à la suite des cas d'utilisation lors de la discipline de définition des exigences. Ils sont complétés par des analystes-concepteurs lors de la discipline d'analyse et de conception.

Les activités pour la production d'un scénarimage consistent à « déduire du cas d'utilisation le tracé des interactions », concevoir les interfaces en définissant la navigation et en identifiant leur contenu et en produisant les maquettes d'interfaces-utilisateurs. La figure 5.10 illustre ces activités. Celles de l'assurance-qualité sont aussi prévues ; elles apparaissent à la figure 5.11.

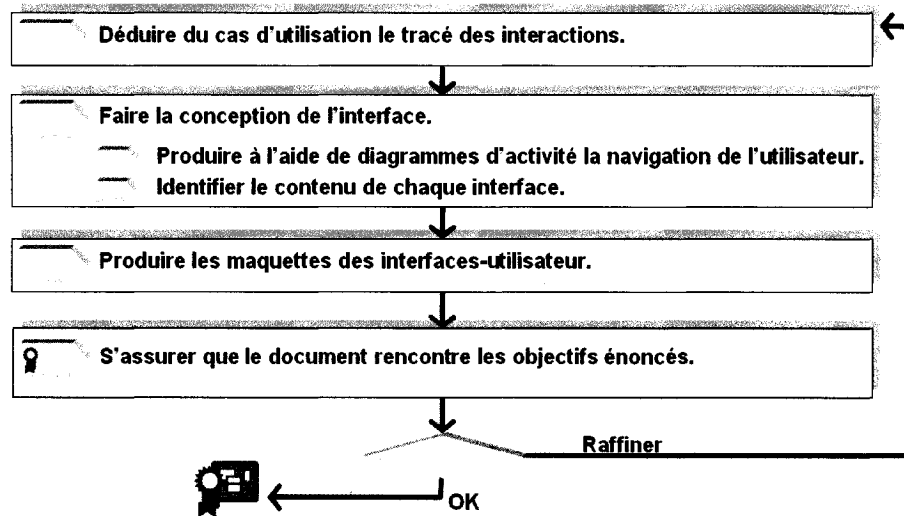


Figure 5.10 : Activités prescrites pour la production des scénarimages⁴⁹.

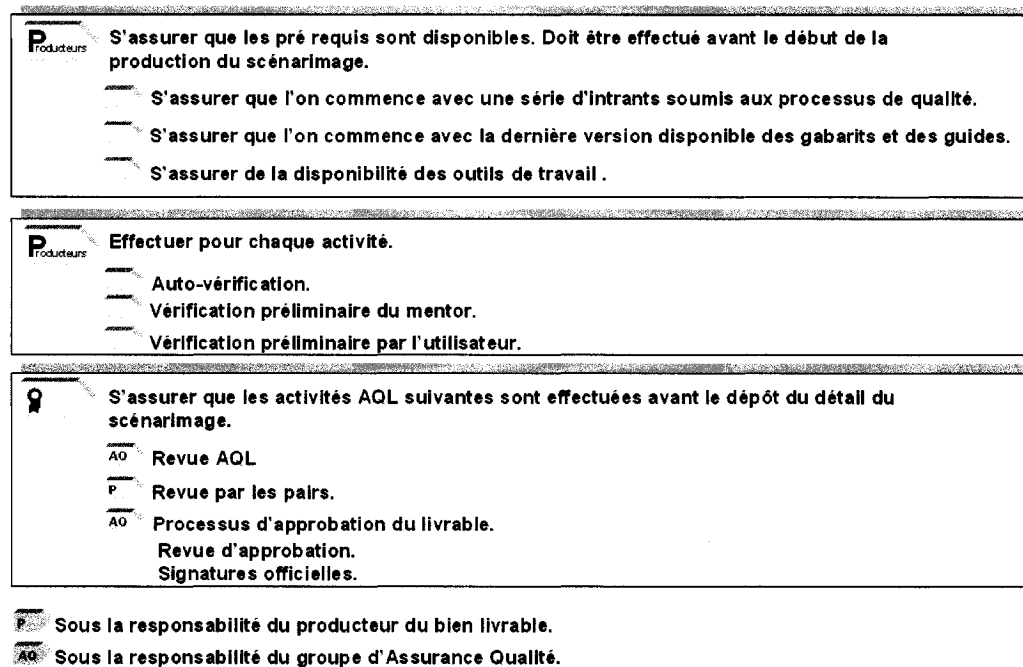


Figure 5.11 : Activités d'assurance-qualité des scénarimages⁵⁰.

Les activités de la discipline d'analyse et de conception sont, par ailleurs, plus proche de l'implémentation et sont donc hors de la portée de cette étude.

Les intrants de la production des scénarimages sont le dictionnaire des termes d'affaires, les caractéristiques logicielles non fonctionnelles, le modèle des cas d'utilisation, le cas d'utilisation, les lignes directrices de conception des interfaces-utilisateurs et les gabarits associés⁵¹.

Les rôles prévus par l'équipe du programme pour l'application du processus, dans le cadre de la modélisation d'affaires et de la définition des exigences, sont définis dans le tableau 5.5.

Tableau 5.5 : Rôles selon l'équipe du programme⁵².

Rôles	Profils	Responsabilités	Biens Livrables
Analyste	D'affaires	- Faire le pont entre l'informatique et les utilisateurs. - Documenter les processus d'affaires ainsi que les besoins détaillés des utilisateurs. - Définir les liens de traçabilité.	- DPO - Caractéristiques logicielles
	Fonctionnel	- Structurer les besoins des utilisateurs sous la forme de cas d'utilisation et d'interfaces utilisateurs. - Définir les liens de traçabilité.	- Modèles de cas d'utilisation - Cas d'utilisation - Scénarimage
Utilisateur		- Documenter la terminologie d'affaires dans un dictionnaire des termes d'affaires. - Exprimer ses attentes sous forme de besoins (fonctionnels et non fonctionnels)	Dictionnaire des termes d'affaires
Architecte	Applicatif	Définir l'architecture fonctionnelle applicative ainsi que les éléments manipulés par chacune des applications qui supportent les processus d'affaires. Cette architecture repose sur l'architecture de référence.	- Architecture applicative - Modèles des concepts clefs
	Architecte	Définir l'architecture et l'infrastructure qui permettront de supporter les processus d'affaires via les applications.	- Architecture de référence - Notes d'architecture
Assurance qualité		- S'assurer que les biens livrables produits soient uniformes et consistants avec les normes et façons de faire. - S'assurer que des activités d'assurance-qualité soient mises en place dans le cadre des disciplines. - Vérifier l'utilisation appropriée de l'outil de traçabilité.	Plan de qualité d'un projet

5.3 L'application : la gestion des réseaux de distribution

La gestion des réseaux de distribution est une application qui fait l'objet d'un projet. C'est le premier sous-système implanté. L'étude de cas porte sur ce sous-système et plus particulièrement sur la modélisation d'affaires et la définition des exigences de l'application.

L'intervention en ergonomie fait partie de la définition des exigences. Le cycle ergonomique a porté sur une partie de cette application.

5.3.1 Description de l'application

L'application, dont l'étude de cas fait l'objet, est définie par 53 DPO et 194 caractéristiques logicielles fonctionnelles⁵³. Les caractéristiques logicielles, formulées dans ce cas-ci pour l'ensemble du programme, sont au nombre de 109. Le tableau 5.6 en dresse l'inventaire, par catégorie.

Tableau 5.6 : Caractéristiques logicielles non fonctionnelles⁵⁴.

Catégorie	Total
Conformité aux standards	18
Disponibilité	11
Documentation	1
Exploitation	7
Extensibilité	2
Flexibilité	6
Intégrité	8
Performance	17
Portabilité	18
Reprise après pannes	5
Sécurité	7
Tolérance aux pannes	2
Utilisabilité	7
Total	109

Le modèle des cas d'utilisation de l'application est présenté au tableau 5.7 ; la figure 5.12 illustre le diagramme UML du sous-module 1 de la gestion des détaillants.

Tableau 5.7 : Sommaire du modèle des cas d'utilisation de l'application⁵⁵.

Catégorie de cas	Nombre de cas
Cas de support	7
Gestion de la formation	1
Gestion des grossistes	6
Gestion de la structure du réseau	7
Sous-module 1	5
Pilotage	2
Gestion des détaillants	
Sous-module 1	6
Sous-module 2	3
Sous-module 3	5
Cas communs	6
Total	48

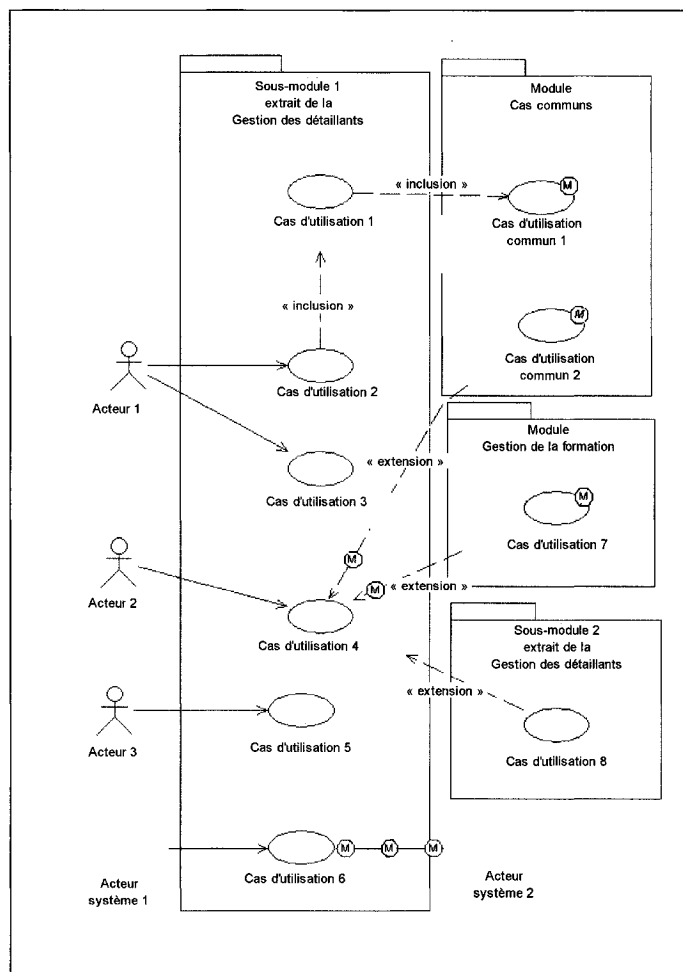


Figure 5.12 : Exemple de diagramme UML du modèle de cas d'utilisation d'un sous-module⁵⁶

Le nombre de cas a évolué en passant de 41 à 50, puis à 48 selon les versions⁵⁷. Les acteurs de l'application sont au nombre de 67, dont 28 sont des acteurs-systèmes⁵⁸. On estime à 80 utilisateurs le nombre d'employés qui vont utiliser d'une façon ou l'autre l'application qui fait l'objet de l'étude de cas⁵⁹.

5.3.2 Partie de l'application traitée par le cycle ergonomique

Six cas d'utilisation de l'application ont été traités par le cycle ergonomique. Les attributs de ces cas d'utilisation sont présentés dans le tableau 5.8.

Tableau 5.8 : Cas d'utilisation traités par le cycle ergonomique⁶⁰.

Attribut	Cas d'utilisation 1	Cas d'utilisation 2	Cas d'utilisation 3	Cas d'utilisation 4	Cas d'utilisation 5	Cas d'utilisation 6
Chemins	10	8	1	4	6	1
Étapes	118	19	10	49	45	3
Sous-étapes	259	29	0	127	85	6
Cas d'inclusion	17	0	0	2	25	0
Règles d'affaire	90	16	14	47	44	5
Messages	20	0	7	18	7	1
Domaines de valeurs	20	19	6	11	1	1
Acteurs primaires	1	1	s.o.	1	s.o.	0
Acteurs secondaires	17	s.o.	s.o.	15	s.o.	5
Préconditions	3	1	s.o.	1	2	s.o.
Postconditions	6	1	1	4	2	1

Le plus important cas d'utilisation comporte 10 chemins et 90 règles d'affaires. Les étapes et sous-étapes correspondent aux gestes posés par les acteurs et les réponses du système. Les cas d'inclusion sont des séquences d'actions qui se répètent. Ils correspondent aux appels faits à d'autres cas durant l'exécution du cas d'utilisation principal⁶¹. Les messages sont ceux qui sont affichés par le système lorsque la situation le commande. Quant aux domaines de valeurs, ce sont les différents choix de valeurs présentés à l'utilisateur lorsque le contexte le nécessite. Ces choix se traduisent habituellement par des listes combinées dans une interface-utilisateur⁶². Les scénarimages correspondants, qui ont été produits suite au cycle ergonomique, apparaissent dans le tableau 5.9. Un total de 34 interfaces-utilisateurs comprenant 727 champs ont été définies.

Tableau 5.9 : Scénarimages du cycle ergonomique⁶³.

Attribut	Scénarimage 1	Scénarimage 2	Scénarimage 3	Scénarimage 4	Scénarimage 5	Scénarimage 6
Interfaces	9	3	8	3	9	2
Champs	241	106	36	131	194	19

La spécification d'interface-utilisateur correspondante compte 24 écrans pour 189 spécifications⁶⁴.

Le personnel qui a participé à la production des artefacts de la définition des exigences qui ont été utilisés et traités lors du cycle ergonomique, était au nombre de 9. Le tableau 5.10 en fait l'énumération.

Tableau 5.10 : Personnel ayant produit des artefacts traités par le cycle ergonomique⁶⁵.

Fonction	Personnel
Analyste Ergonomie	1
Analyste fonctionnel	4
Atelier logiciel	1
Ergonome	2
Gestion du changement	1
Total	9

Les représentants des utilisateurs ont plus de 20 ans d'expérience dans l'entreprise. Les analystes fonctionnels et les analystes d'affaires ont tous plus de 10 ans d'expérience en informatique. Ils ont cependant peu ou pas d'expérience du domaine d'affaires⁶⁶.

Le personnel qui a participé à la révision des artefacts qui ont été utilisés et traités lors du cycle ergonomique était au nombre de 9. Le tableau 5.11 en fait l'énumération.

Tableau 5.11 : Personnel ayant révisé des artefacts⁶⁷

Fonction	Personnel
Analyste d'affaires	2
Analyste Ergonomie	1
Analyste fonctionnel	4
Analyste sécurité	1
Architecte	1
Architecte applicatif	1
Ergonome	2
Gestion du changement	1
Représentant utilisateur	3
Directeur Architecture	1
Analyste Groupe Atelier	1
Centre de livraison - Infrastructure	2
Centre de livraison - Réalisation	1
Total	21

5.4 La nature de l'intervention en conception centrée sur l'utilisateur

C'est la direction Fonctionnel de l'équipe du programme qui a recruté des ergonomes par appel d'offres au début des activités de définition des exigences de la phase d'élaboration pour l'intervention en conception centrée sur l'utilisateur. La direction Fonctionnel est responsable des analystes d'affaires et des représentants des utilisateurs. Elle est en quelque sorte la

représentante du client au sein de l'équipe du programme⁶⁸. Le mandat des ergonomes, défini par elle consiste à :

- organiser l'information et la navigation dans une application ;
- intervenir pour les aspects relevant de l'ergonomie cognitive ;
- selon sa compréhension des capacités humaines, perception, traitement de l'information, capacités et limites de la mémoire, etc., appliquer les règles qui orientent les choix des éléments graphiques, sonores, animés ainsi que les fonctions de navigation ;
- s'assurer que tous ces éléments respectent les critères et les normes en ergonomie cognitive et que les interfaces soient conviviales, cohérentes et faciles d'utilisation.

Leurs tâches sont :

- participer à la définition des spécifications de l'application ;
- participer au montage des maquettes préliminaires et à la scénarisation interactive ;
- participer à la mise en place de groupes cibles et à la définition des critères adaptés au type d'application et à l'environnement physique ;
- monter les grilles d'évaluation pour les techniciens attitrés aux tests et aux validations, et faire le suivi des problèmes et des corrections auprès du responsable de la qualité⁶⁹.

L'intervention ergonomique a donc porté sur une application du programme « Gestion des réseaux de distribution ». La modélisation d'affaires de cette application a commencé en septembre 2002 avec les premiers DPO. Les dernières versions des cas d'utilisation ont été approuvées en novembre 2003. Cependant, l'essentiel des efforts de production des cas d'utilisation et de leurs scénarimages correspondants ont été faits de janvier à mai 2003.

L'intervention ergonomique a porté sur la conception des interfaces-utilisateurs qui touchaient 6 des 48 cas d'utilisation de l'application. Il en est ressorti une spécification d'interfaces-utilisateurs de 24 écrans. Pour obtenir ces résultats, 2 séries de tests d'utilisabilité ont été tenues. Ces séries de tests comportaient chacune une séance de pré-test d'utilisabilité avec les représentants des utilisateurs, une séance de tests d'utilisabilité avec des utilisateurs finaux et une séance de pas à pas faite avec les représentants des utilisateurs et les analystes fonctionnels concernés.

Les utilisateurs finaux participant aux tests étaient au nombre de 4 incluant les représentants des utilisateurs étant eux-mêmes des utilisateurs finaux. Les écrans de saisie et de modification étaient pour leur usage dans les opérations courantes de l'entreprise. Par contre, les écrans de consultation étaient susceptibles d'être utilisés par 30 utilisateurs finaux.

5.4.1 La conception des interfaces-utilisateurs par les analystes

Les analystes fonctionnels produisent les scénarimages qui correspondent aux interfaces-utilisateurs. La conception initiale s'inspire de lignes directrices, le « cadre de conception des interfaces Web » rédigé pour les besoins du projet⁷⁰. Un scénarimage correspond à un cas d'utilisation. Les scénarimages sont produits à l'occasion de rencontres avec les représentants des utilisateurs⁷¹. La figure 5.13 illustre le processus.

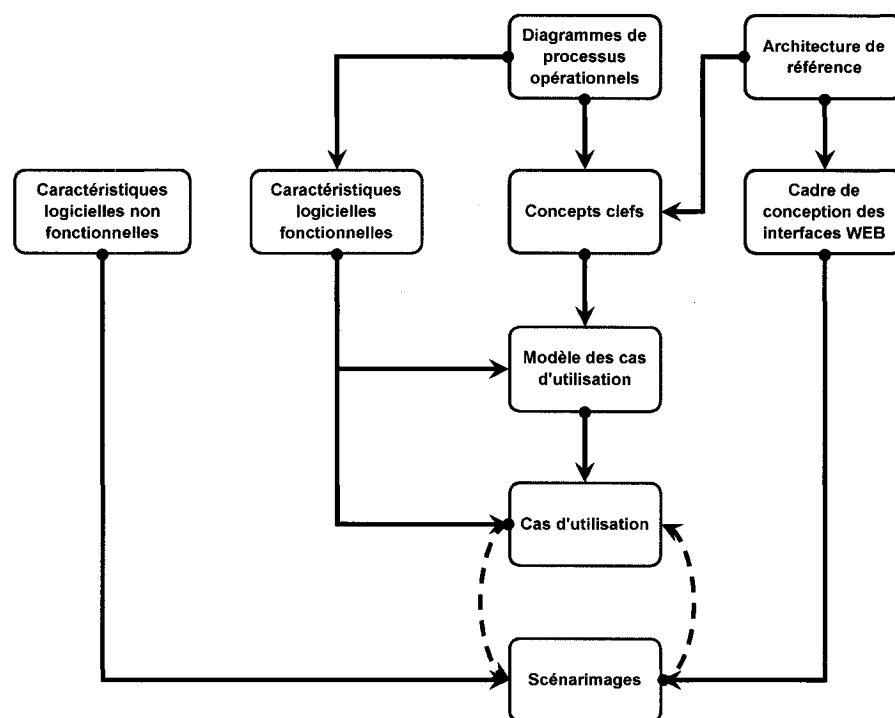


Figure 5.13 : Processus de production des artefacts avant l'intervention⁷².

5.4.2 La conception des interfaces-utilisateurs par les ergonomes

Le processus appliqué lors de l'intervention est inspiré du cycle d'ingénierie de l'utilisabilité défini par Deborah Mayhew (1999) et illustré à la figure 5.14.

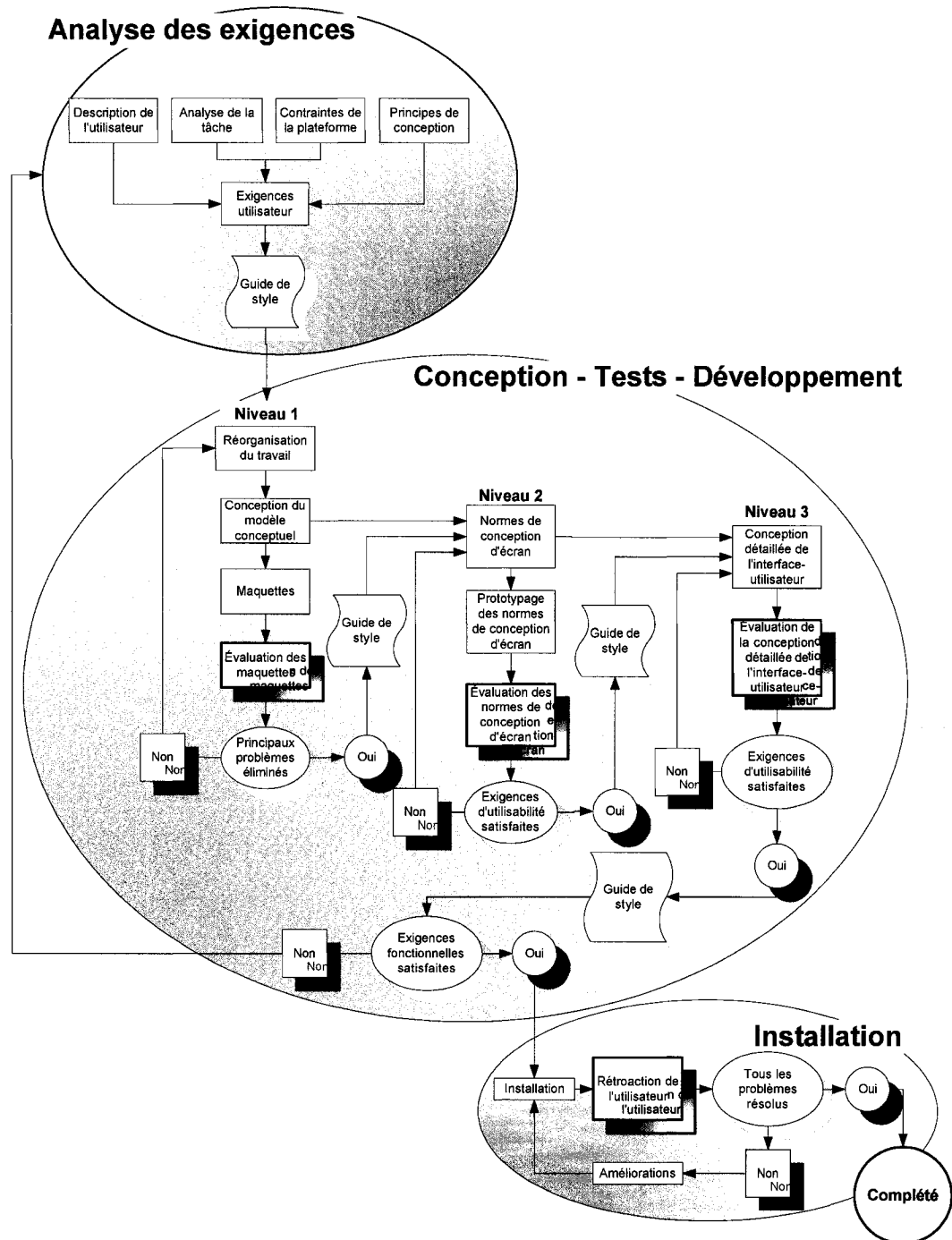


Figure 5.14 : Cycle de vie de l'ingénierie de l'utilisabilité (Mayhew, 1999).

Une analyse de contexte d'utilisation décrit d'abord le cadre dans lequel est produite l'application. L'analyse de contexte d'utilisation décrit le contexte environnemental soit : l'entreprise, le domaine d'affaires et le projet lui-même qui comprend l'environnement technologique⁷³.

Puis, à partir de l'analyse de contexte d'utilisation, les tâches les plus critiques et les cas d'utilisation correspondants sont identifiés. La criticité des tâches est établie selon l'importance organisationnelle qu'elles représentent et la fréquence de leur exécution.

À partir de l'analyse de la tâche, des caractéristiques logicielles fonctionnelles et non fonctionnelles ainsi que des cas d'utilisation, une maquette papier des interfaces-utilisateurs est conçue et soumise à des tests d'utilisabilité auprès des représentants des utilisateurs et des utilisateurs finaux⁷⁴. Un guide de style est aussi produit au fur et à mesure que les tests confirment la validité des règles à y inclure⁷⁵. La figure 5.15 illustre ce processus.

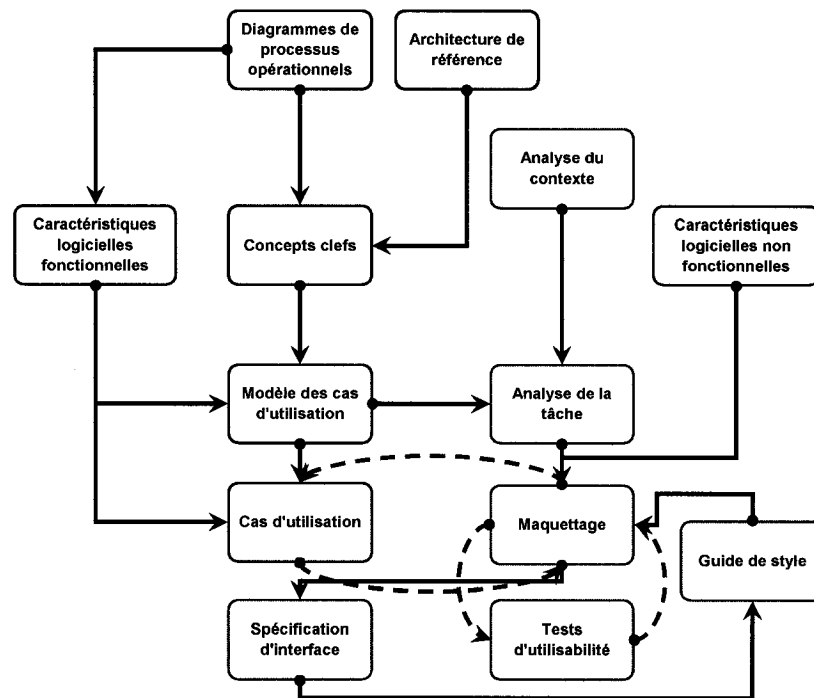


Figure 5.15 : Processus de production des artefacts lors de l'intervention ergonomique⁷⁶.

Un deuxième cycle est ensuite exécuté avec le ou les analystes fonctionnels concernés par les cas d'utilisation⁷⁷. Des spécifications d'interface sont ensuite produites. Les interfaces-

utilisateurs correspondent aux cas d'utilisation du premier cycle auxquels s'ajoutent d'autres cas d'utilisation. C'est la tâche et la validation par les tests d'utilisabilité qui dictent cette correspondance⁷⁸. Parallèlement, un prototype d'interface-utilisateur dynamique est produit et soumis à des tests d'utilisabilité auprès des représentants des utilisateurs et des utilisateurs finaux⁷⁹.

5.4.3 Les activités complémentaires à l'intervention

D'autres activités se sont ajoutées au processus de conception d'interfaces. La figure 5.16 illustre l'ensemble des activités de l'intervention. Les interfaces-utilisateurs des applications informatiques avec lesquelles le personnel de l'entreprise était familier avaient été conçues en mode texte ou en mode *Windows, icons, menus and pointers* (WIMP). Une présentation sur les différences entre les interfaces WIMP et les interfaces-utilisateurs a été produite et faite au personnel responsable de la réalisation technique du programme⁸⁰.

Une présentation sur le déroulement du premier cycle ergonomique a été faite à l'ensemble du personnel du programme impliqué dans les phases d'opportunité et d'élaboration du programme⁸¹. Des propositions de modifications au processus de création de spécification d'interface ont été soumises⁸².

L'en-tête des pages Web des interfaces-utilisateurs comporte certaines informations et des fonctions, telles une fonction de recherche et des menus de navigation. La spécification de cet en-tête a fait l'objet d'une démarche de conception⁸³.

Afin de s'assurer de la faisabilité technique de la spécification des interfaces-utilisateurs, deux actions ont été entreprises. D'une part, le prototype dynamique a été réalisé par le personnel responsable du développement de l'application⁸⁴. D'autre part, un échange d'informations a été instauré afin de s'assurer que les spécifications de l'interface-utilisateur qui présentaient une difficulté technique soient rencontrées. C'est ce à quoi a été consacrée l'étude des questions techniques et ergonomiques⁸⁵.

Des discussions avec l'équipe d'assurance-qualité se sont traduites par des recommandations pour le contrôle de la qualité des interfaces-utilisateurs⁸⁶. Enfin, une formation pour produire les spécifications d'interface-utilisateur a été transmise aux groupes concernés, notamment aux analystes fonctionnels, aux analystes de l'assurance-qualité et aux représentants des utilisateurs⁸⁷.

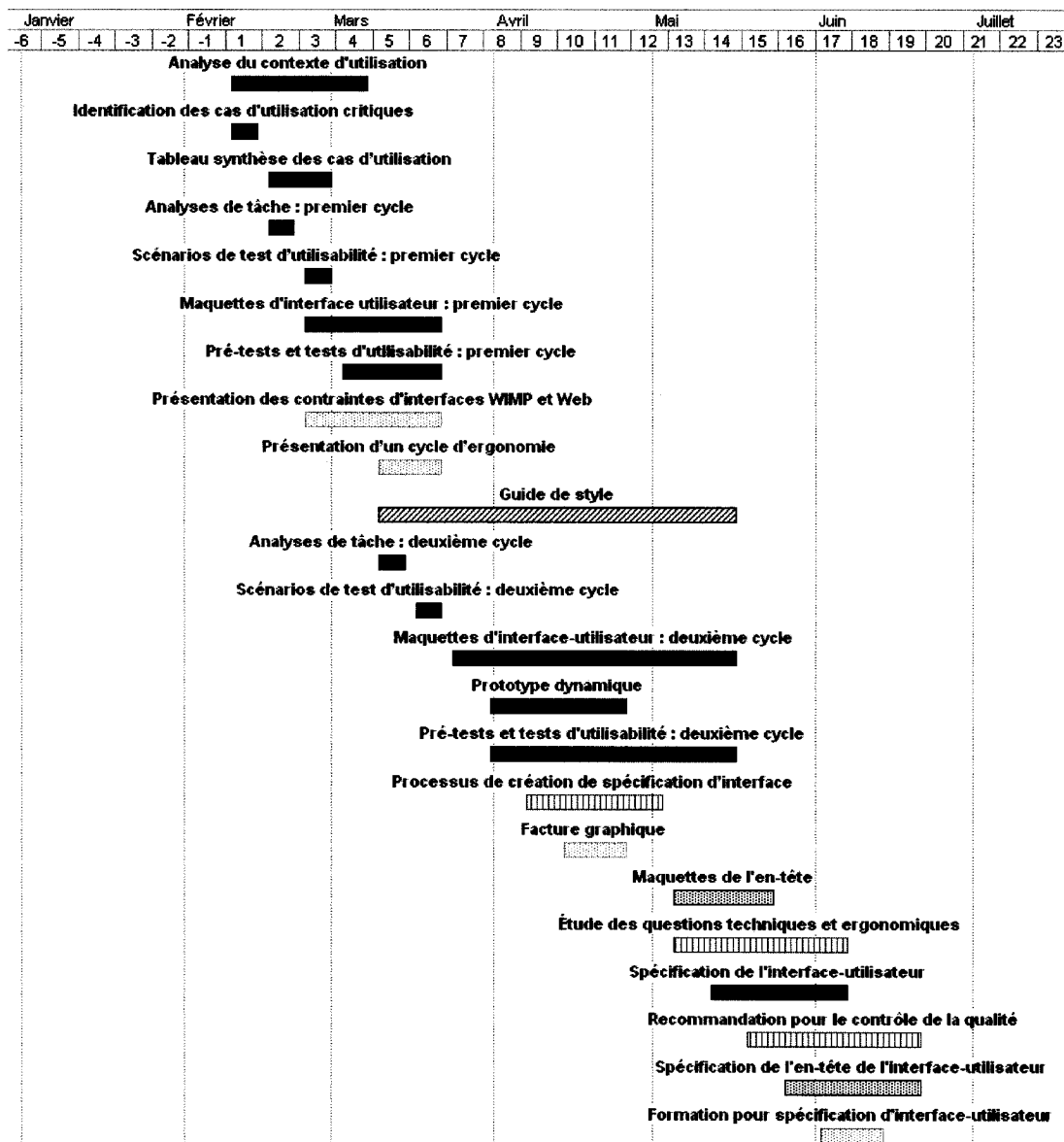


Figure 5.16 : Activités de l'intervention ergonomique⁸⁸.

Une planification des activités d'ergonomie régulièrement mise à jour a été déposée. Des rapports hebdomadaires visant à faire le point sur l'évolution de l'intervention ont été présentés⁸⁹.

Notes et références

- ¹ Rapports annuels de l'entreprise.
- ² États financiers inclus dans les rapports annuels de l'entreprise.
- ³ Énoncé de mission inscrit dans les rapports annuels de l'entreprise.
- ⁴ Rapports annuels de l'entreprise.
- ⁵ Nos observations, *Analyse de contexte d'utilisation* et *Compte rendu d'entrevue avec analyste d'affaires et représentant d'utilisateur*.
- ⁶ *Compte rendu d'entrevue avec analyste d'affaires et représentant d'utilisateur*, p. 1.
- ⁷ Document intitulé Besoins d'affaires cité par *Analyse de contexte d'utilisation*.
- ⁸ Les Bulletins hebdomadaires du projet, *Architecture de référence*, p. 11, et *Analyse de contexte d'utilisation*, p. 12.
- ⁹ Modèles des cas et *Analyse de contexte d'utilisation*, p. 24.
- ¹⁰ Nos observations, tel qu'annoncé dans *Bulletin du projet vol.1 n° 9*, p. 1 et *Compte rendu d'entrevue avec analyste d'affaires et représentant d'utilisateur*.
- ¹¹ *Analyse de contexte d'utilisation*, p. 14.
- ¹² Rapports annuels de l'entreprise.
- ¹³ Organigramme du projet apparaissant dans l'artefact *Analyse de contexte d'utilisation*, p. 14.
- ¹⁴ Organigramme du projet et nos observations.
- ¹⁵ Version expurgée de l'organigramme du projet.
- ¹⁶ Organigramme du projet et nos observations.
- ¹⁷ Analyse des documents définissant les lignes directrices du projet, notamment les Fiches signalétiques, Rational (2002) et Kruchten (2001).
- ¹⁸ Nos observations et les documents disponibles, notamment les gabarits, Fiches signalétiques et guides.
- ¹⁹ *Façon de faire - Vue globale*, p. 24.

- ²⁰ Nos observations et *Façon de faire - Vue globale*, p. 24.
- ²¹ *Façon de faire - Vue globale*, p. 8, produite par l'équipe responsable de la discipline Environnement du processus RUP©.
- ²² *Façon de faire - Vue globale*, p. 13.
- ²³ *Compte rendu d'entrevue avec analyste d'affaires et représentant d'utilisateur*, p. 3.
- ²⁴ *DPO - Émission d'un numéro de détaillant*.
- ²⁵ *Fiche signalétique - Modèle des concepts-clés*, p. 10.
- ²⁶ *Façon de faire - Vue globale*, p. 7.
- ²⁷ *Dictionnaire des termes d'affaires*.
- ²⁸ *Façon de faire - Vue globale*, p. 18.
- ²⁹ *Fiche signalétique - Registre des caractéristiques logicielles fonctionnelles*, p. 3, et *Traçabilité Gestion des Réseaux*.
- ³⁰ *Registre des caractéristiques logicielles - Réseau de distribution*.
- ³¹ *Fiche signalétique - Registre des caractéristiques logicielles non fonctionnelles*, p. 6.
- ³² *Ibidem*, p. 3.
- ³³ *Registre des caractéristiques logicielles non fonctionnelles*.
- ³⁴ *Fiche signalétique - Modèle de cas d'utilisation*, p. 9.
- ³⁵ *Modèle des cas d'utilisation - Réseau de distribution - Version 5*, version expurgée, p. 4.
- ³⁶ *Ibidem*, p. 7.
- ³⁷ *Ibidem*, p. 3.
- ³⁸ *Modèle des cas d'utilisation - Réseau de distribution - Versions 2, 4 et 5*.
- ³⁹ *Gabarit de cas d'utilisation*.
- ⁴⁰ *Ibidem*, p. 2,
- ⁴¹ *Fiche signalétique - Détail de cas d'utilisation*, p. 3.

⁴² *Ibidem*, p. 7.

⁴³ *Ibidem*, p. 9.

⁴⁴ *Fiche signalétique – Scénarimage*, p. 3.

⁴⁵ *Ibidem*, p. 4.

⁴⁶ *Ibidem*, p. 5.

⁴⁷ *Gabarit de cas d'utilisation*.

⁴⁸ *Fiche signalétique – Scénarimage*, p. 14.

⁴⁹ *Ibidem*, p. 6.

⁵⁰ *Ibidem*, p. 8.

⁵¹ *Ibidem*, p.11.

⁵² *Rôles- profils et responsabilités*.

⁵³ Analyse de la version du *Registre des caractéristiques logicielles fonctionnelles* produite durant l'étude de cas.

⁵⁴ Analyse de la version du *Registre des caractéristiques logicielles non fonctionnelles* produite durant l'étude de cas.

⁵⁵ *Modèle des cas d'utilisation - Réseau de distribution - Version 5*.

⁵⁶ *Modèle des cas d'utilisation - Réseau de distribution - Version 5*, p. 22.

⁵⁷ Analyse des versions 2, 4 et 5 du modèle des cas.

⁵⁸ Analyse du rapport *Liste des acteurs - Réseau de distribution*.

⁵⁹ *Compte rendu d'entrevue avec analyste d'affaires et représentant d'utilisateur*, p. 1.

⁶⁰ Version révisée des artefacts Cas d'utilisation pertinents.

⁶¹ *Guide de notation des cas d'utilisation*, p.18 et 19.

⁶² Analyse du contenu des artefacts Cas d'utilisation pertinents.

⁶³ Analyse du contenu des artefacts Scénarimages pertinents.

- ⁶⁴ *Spécification d'interface - Dossier détaillant.*
- ⁶⁵ *Analyse du contexte d'utilisation.*
- ⁶⁶ *Compte rendu d'entrevue avec analyste d'affaires et représentant d'utilisateur, p. 1.*
- ⁶⁷ *Analyse de l'ensemble des artefacts qui ont été utilisés durant le cycle ergonomique.*
- ⁶⁸ *Organigramme du projet et nos observations.*
- ⁶⁹ *Appel de propositions pour les services d'ergonomes, p. 15.*
- ⁷⁰ *Cadre de conception des interfaces WEB.*
- ⁷¹ *Fiche signalétique – Scénarimage.*
- ⁷² *Observations et fiches signalétiques pertinentes.*
- ⁷³ *Analyse du contexte d'utilisation.*
- ⁷⁴ *Ibidem.*
- ⁷⁵ *Guide de style des interfaces Web.*
- ⁷⁶ *Nos observations.*
- ⁷⁷ *Rapports hebdomadaires de l'ergonomie des interfaces.*
- ⁷⁸ *Spécification d'interface - Dossier détaillant.*
- ⁷⁹ *Rapports hebdomadaires de l'ergonomie des interfaces, p.11.*
- ⁸⁰ *Distinctions WIMP – Web.*
- ⁸¹ *Intervention en ergonomie - Bilan du premier cycle.*
- ⁸² *Proposition de modification de scénarimage.*
- ⁸³ *Spécification d'interface - Zone de navigation primaire.*
- ⁸⁴ *Rapports hebdomadaires de l'ergonomie des interfaces.*
- ⁸⁵ *Étude des questions techniques et ergonomiques.*
- ⁸⁶ *Nos observations et Formation - Scénarimage et ergonomie.*
- ⁸⁷ *Formation - Scénarimage et ergonomie.*

⁸⁸ Inclus dans *Rapports hebdomadaires de l'ergonomie des interfaces*.

⁸⁹ *Rapports hebdomadaires de l'ergonomie des interfaces*.

CHAPITRE VI

ANALYSE DES DONNÉES

Les données abordées ici correspondent aux phénomènes observés qui ont une pertinence avec les questions de recherche élaborées pus tôt. Les données sont présentées alternativement selon l'application faite par l'équipe du programme et selon la perspective de l'intervention ergonomique pour chaque unité d'analyse, telle que définie par les questions de recherches inspirées des principes et des activités de ISO 13407. On pourra ainsi établir, par la suite, la comparaison entre les prescriptions des processus et l'application qui en a été faite.

6.1 (1) Est-ce que l'application du RUP© soutient la participation active des utilisateurs et définit le contexte d'utilisation selon ISO 13407 ?

La participation des utilisateurs devait être diversifiée et directe dans le processus de développement pour ISO 13407. Le contexte d'utilisation doit être défini par l'identification et la description de l'utilisateur, de l'environnement et de la tâche. L'application faite par l'équipe du programme et l'approche utilisée lors de l'intervention ergonomique sont décrites.

6.1.1 Équipe du programme : architecture de référence, DPO et caractéristiques logicielles

Pour l'équipe du programme, l'implication des utilisateurs est importante à toutes les étapes de la modélisation d'affaires et de la définition des exigences. Le contexte d'utilisation n'est pas identifié. On doit plutôt consulter plusieurs artefacts pour retrouver les informations habituellement inscrites dans le document Vision. La participation des utilisateurs et ce qui pourrait correspondre à la description du contexte d'utilisation sont ici abordés.

6.1.1.1 Une participation systématique des utilisateurs

Les représentants des utilisateurs, appelés utilisateurs dans la description du processus, participent systématiquement à la définition des processus d'affaires et à la définition des exigences. Ainsi, il participent à l'élaboration et approuvent les DPO, les caractéristiques logicielles, les cas d'utilisation et les scénarimages. Il participe aussi à la vérification du modèle des concepts clés et des modèles des cas d'utilisation¹.

Ces représentants ne sont pas nécessairement des utilisateurs finaux du système et peuvent ne pas être des employés de l'organisation². Pour la partie du système faisant l'objet de l'étude de

cas, les représentants sont eux-mêmes des utilisateurs finaux du système. Au nombre de deux, ils sont au service de l'entreprise depuis au moins 20 ans. L'un des deux a été dans le domaine d'affaires touché par l'application pendant tout ce temps³.

La cueillette des informations se fait essentiellement par des rencontres avec les représentants des utilisateurs. Ces derniers consacrent de 2 à 30 heures à la production des cas d'utilisation selon la nature du cas. Pour la production des scénarimages, 25 % du temps correspondant au cas d'utilisation est ajouté⁴. L'accès aux utilisateurs finaux est possible, mais les analystes s'en prévalent généralement peu⁵.

6.1.1.2 L'environnement est partiellement défini

L'adaptation qui a été faite du RUP© ne prévoyait pas la production d'un document Vision⁶. Des caractéristiques logicielles fonctionnelles et des caractéristiques logicielles non fonctionnelles, qui sont respectivement l'équivalent des caractéristiques de produit et l'équivalent des contraintes, du document Vision ont cependant été définies⁷. Le contexte n'est donc pas décrit tel que le prescrit le RUP©⁸. L'environnement est partiellement défini ailleurs, mais les caractéristiques des utilisateurs ne sont pas relevées et il n'y a pas d'analyse de la tâche⁹.

L'environnement matériel et les équipements ne sont pas décrits. Une partie de l'environnement technique est défini dans un artefact : l'architecture de référence qui décrit les couches logicielles et les stratégies d'interface. Il n'y a pas de description de l'environnement physique. L'environnement organisationnel n'est pas non plus décrit¹⁰.

6.1.1.3 Les acteurs décrits par leur rôle

Comme il n'y a pas de document Vision, les caractéristiques des utilisateurs ne sont pas définies. Il n'y a pas non plus de mécanisme formel pour enregistrer les demandes des parties prenantes où peut apparaître la description d'utilisateurs¹¹. Les acteurs du système sont identifiés dans le modèle des cas d'utilisation et apparaissent dans ceux qui leur sont associés¹². Des acteurs d'affaires sont théoriquement identifiés lors de l'élaboration du modèle des concepts-clés qui vise à fournir une cartographie des processus d'affaires inspirée des DPO. On doit y trouver les groupes de l'organisation, les acteurs, internes et externes du domaine d'affaires, les concepts ou entités d'affaires ainsi que les principales responsabilités¹³.

Les acteurs du système sont énumérés et ne sont décrits que par leur rôle dans le modèle des cas d'utilisation. Ainsi, l'acteur « Responsable - Gestion groupe de détaillants » est ainsi décrit :

« Le Responsable - Gestion groupe de détaillants effectue la gestion relative aux groupes de détaillants : création du groupe, sélection des détaillants de ce groupe, mise à jour de la liste des détaillants faisant partie de ce groupe¹⁴. »

Il n'y a pas d'autre description de caractéristiques d'utilisateurs. En d'autres termes, la description de l'acteur prescrite par le RUP© est appliquée, mais les caractéristiques de l'utilisateur qui en découlent ne sont pas relevées.

6.1.1.4 Les tâches : DPO et cas d'utilisation

Il n'y a pas d'analyse de la tâche. Les cas d'utilisation répondent à différents objectifs, mais non à celui de modéliser la tâche d'un utilisateur. Ils sont dérivés des DPO et des caractéristiques logicielles¹⁵.

Les besoins d'affaires sont représentés par les DPO. Le découpage de ces processus est basé sur le découpage des applications du programme Neptune défini par l'architecture de référence. De ces DPO, sont extraits les activités qui nécessitent l'utilisation d'une application¹⁶.

Les caractéristiques logicielles fonctionnelles sont extraites par application à partir de ces activités¹⁷. Les objectifs des caractéristiques logicielles consistent « à énumérer les besoins détaillés de l'utilisateur par rapport aux DPO, à fournir une base pour la définition des exigences et une base pour la traçabilité¹⁸ ».

Un modèle des cas d'utilisation est produit à partir des caractéristiques logicielles fonctionnelles. Il doit répondre aux objectifs suivants :

- faire ressortir les fonctionnalités du système :
 - identifier qui ou quoi va interagir avec le système ;
 - définir les limites du système et supporter les caractéristiques logicielles fonctionnelles ;
 - définir la structuration des fonctionnalités du système ;
- s'assurer que l'on couvre les caractéristiques logicielles fonctionnelles exprimées par l'utilisateur¹⁹.

Le modèle de cas n'a donc pas pour mission d'identifier ou de structurer des tâches que l'utilisateur exercera avec le système, selon leur pertinence ou leur criticité. Il vise plutôt à décrire les fonctionnalités du système et à s'assurer que ces dernières sont toutes présentes.

Quant au cas d'utilisation lui-même qui a été décrit au chapitre précédent, son premier objectif est de « définir le comportement du système » et non celui de décrire la tâche de l'utilisateur. Le deuxième objectif était d'offrir une base contractuelle. Le troisième était de fournir une « unité de base » pour l'exécution de différentes activités du processus. Parmi, ces activités, il y avait la conception des interfaces-utilisateurs. Dans la pratique, l'interprétation qu'on en fera sera de lier les interfaces-utilisateurs au cas d'utilisation²⁰.

Le cas d'utilisation doit décrire le flot normal des événements et les flots alternatifs des événements. Ses intrants ne sont pas des analyses de la tâche, mais les DPO qui sont des processus d'affaires ou les caractéristiques logicielles fonctionnelles qui décrivent le système²¹.

Le cas d'utilisation est aussi défini comme une façon d'utiliser le système pour l'exécution d'une tâche d'affaires. Il se présente comme un dialogue entre le système et des acteurs. Et il est déclenché par un acteur qui peut être un humain ou un automate. Les acteurs sont énumérés dans la description du détail des cas d'utilisation, mais ne sont pas décrits²².

6.1.2 Intervention ergonomique : analyse de contexte

Les ergonomes ont produit une analyse du contexte qui regroupait les informations telles que le prescrivent les normes ISO. On y trouve la description des environnements, des utilisateurs et des tâches.

6.1.2.1 La participation des utilisateurs

Les ergonomes ont présentés aux responsables des représentants des utilisateurs des rapports hebdomadaires de l'évolution de leurs travaux²³. En diversifiant les types d'évaluation, ils ont fait varier la nature de la participation des utilisateurs²⁴.

6.1.2.2 Les environnements

L'environnement organisationnel a d'abord été défini à partir d'un document issu de l'étude de préféabilité, de rapports annuels et d'entrevues²⁵. L'équipement et l'environnement technique ont été identifiés à partir du document d'architecture de référence ou sur le terrain²⁶. Le type de

poste de travail et la taille des écrans utilisés par les utilisateurs ont, par exemple, été relevés. L'environnement physique, tel le bruit ambiant, l'éclairage et l'aménagement des postes de travail, a été inventorié²⁷.

6.1.2.3 La description des utilisateurs

L'identification et la description des utilisateurs ont été faites par les ergonomes. Les utilisateurs finaux ont été désignés par les représentants des utilisateurs. À partir d'entrevues faites avec les utilisateurs finaux, leurs caractéristiques ont été identifiées. Celles-ci comprenaient :

- leur titre d'emploi et la description de leur tâche ;
- leur âge et leur sexe ;
- leur ancienneté dans l'organisation ;
- leur qualification et leur statut académique ;
- leur expérience avec le système patrimonial ;
- leur expérience avec d'autres systèmes ou applications informatiques ;
- leur expérience dans l'exercice de la tâche ;
- leur statut dans l'organisation : cadre ou syndiqué, permanent ou occasionnel, temps complet ou temps partiel.

Les profils des autres employés, qui pouvaient être amenés à exercer la même tâche, ont été identifiés²⁸.

6.1.2.4 L'analyse contextuelle de la tâche

Les ergonomes ont produit des analyses de la tâche. Les représentants des utilisateurs ont été invités à désigner les tâches les plus critiques et les plus courantes que les utilisateurs finaux ont confirmées. L'exécution des tâches a été filmée. Ces films ont été visionnés et analysés. Des diagrammes hiérarchiques de la tâche et des scénarios d'utilisation ont été produits²⁹.

Les facteurs liés à la tâche, telles la criticité, la complexité, les conditions d'exécution, la concurrence avec d'autres tâches, comme la fréquence des appels téléphoniques, ainsi que l'impact juridique et financier de la tâche sur les opérations de l'entreprise ont été identifiés. Les

artefacts du système courant, tels les formulaires en papier, les copies d'écran du système patrimonial et autres éléments nécessaires à la tâche ont été inventoriés. Les volumes de données traitées, la fréquence d'exécution de la tâche et les temps-réponses du système ont été recueillis³⁰.

6.1.3 Synthèse de la question (1)

L'application faite par l'équipe de projet prévoit donc que les représentants des utilisateurs participent, lors d'ateliers, à la production des DPO, à la définition des caractéristiques logicielles, à la réalisation des cas d'utilisation et aux scénarimages correspondants. Il n'y a pas d'identification de contexte d'utilisation.

Pour leur part, les ergonomes sollicitent systématiquement les utilisateurs et leurs représentants sont pour la planification des activités d'ergonomie, la définition du contexte d'utilisation et l'évaluation des solutions de conception avec différents types d'activités

6.2 (2) L'application du RUP© permet-elle une bonne spécification des exigences-utilisateurs notamment par une bonne répartition des fonctions entre les utilisateurs et la technologie selon ISO 13407 ?

La spécification des exigences liées à l'utilisateur et à l'organisation suppose une répartition adéquate des fonctions qui tient compte du contexte de la tâche selon ISO 13407. Outre la conception des interfaces-utilisateurs, cela comprend aussi certaines qu'on désigne par exigences non fonctionnelles comme, par exemple, celles d'utilisabilité et de sécurité.

6.2.1 Équipe du programme : caractéristiques logicielles non fonctionnelles

La notion d'exigences-utilisateurs n'existe pas pour l'équipe du programme. Des exigences d'utilisabilité et de sécurité sont définies comme caractéristiques logicielles non fonctionnelles. Les caractéristiques logicielles non fonctionnelles sont les exigences formulées par les parties prenantes qui n'apparaissent pas dans les DPO³¹. Cependant, parmi les activités prévues pour leur formulation, on relève l'identification des activités des DPO qui nécessitent l'utilisation d'une application. Cela concorde avec la notion d'exigence d'automatisation définie par le RUP©³².

Les exigences d'utilisabilité y sont peu nombreuses, moins d'une dizaine, et plutôt d'ordre général. Ainsi, on y spécifie, entre autres, que l'interface-utilisateur du système est en mode Web, qu'une aide en ligne contextuelle doit être disponible ou que les rapports doivent être imprimables à partir d'une imprimante locale. Elles correspondent aux contraintes qu'on aurait retrouvées dans le document Vision s'il avait été produit. La nature des activités de l'organisation étudiée dans l'étude de cas a fait, de la sécurité des données, une préoccupation déterminante. Une équipe distincte en a assuré le suivi³³.

6.2.2 Intervention ergonomique : exigences incluses dans la spécification d'interface

Les exigences liées à l'utilisateur et à l'organisation ont été abordées à plusieurs occasions lors de l'intervention ergonomique, mais elles n'ont pas été formellement documentées. Ainsi, les spécifications incluses dans les cas d'utilisation et celles des interfaces-utilisateurs ont été influencées par les commentaires formulés par les ergonomes, mais les corrections qui en ont résultées ne leur ont pas été formellement associées à l'intervention ergonomique³⁴.

Un cas d'utilisation portant sur la saisie d'adresses, par exemple, a fait l'objet d'une telle démarche. L'analyste fonctionnel modélisait ses données en se référant à la ventilation des éléments d'une adresse telle que déterminée par le service des postes. Les ergonomes recommandaient plutôt une ventilation moins détaillée, compte tenu de l'effort que cela susciterait de la part des utilisateurs concernés et des besoins réels de l'organisation³⁵.

Les exigences d'utilisabilité et de sécurité étaient intégrées à la spécification d'interface.

6.2.3 Synthèse de la question 2

Pour l'équipe du programme, cette répartition n'est pas faite. Les exigences-utilisateurs n'existent pas. Les exigences d'utilisabilité sont définies à haut niveau. Pour les ergonomes, la répartition doit se faire et les exigences-utilisateurs sont incluses dans la spécification d'interface.

6.3 Itération des solutions de conception

L'itération signifie que le travail est repris jusqu'à ce qu'une solution satisfaisante soit atteinte. Pour ISO 13407, il faut une matérialisation d'une solution de conception, des évaluations de cette solution et une gestion de ces itérations.

6.3.1 (3) La matérialisation des solutions de conception se fait-elle par l'application du RUP© selon ISO 13407 ?

La matérialisation des solutions de conception doit d'abord s'inspirer du contexte d'utilisation pour ISO 13407. Ensuite, s'ajoutent l'état des techniques, l'expérience et les connaissances des participants et, enfin, les normes et guides.

6.3.1.1 Équipe du programme : un cadre de conception

Les analystes fonctionnels s'inspirent d'abord du cas d'utilisation auquel le scénarimage est associé. Le scénarimage et le cas d'utilisation sont d'ailleurs produits parallèlement. Bien qu'elle ne soit pas prescrite, certains analystes complètent par l'observation du fonctionnement des systèmes patrimoniaux. Le modèle des cas d'utilisation est une référence utilisée³⁶.

Le « cadre de conception des interfaces Web » fixant des lignes directrices de conception d'interface-utilisateur, expliquant l'utilisation des composants d'interface et présentant des gabarits d'écran générique, a été produit par différents intervenants sous la supervision d'un architecte. Ce guide a été rédigé suite à la publication du document d'architecture de référence lors de la phase d'inception. Il constitue une référence pour la production des scénarimages par les analystes. Des exemples d'utilisation des composants dans la conception d'interfaces Web sont présentés : différents types d'interfaces et de fenêtres, des fonctions utilitaires et autres. Il correspond aux guides spécifiques au projet qui doivent être produits selon les prescriptions du RUP©. Une interface-utilisateur doit correspondre à un cas d'utilisation³⁷.

6.3.1.2 Intervention ergonomique : plusieurs sources

L'analyse du contexte d'utilisation a permis de repérer et de décrire les tâches critiques de l'organisation. Les fonctionnalités liées à ces tâches sont identifiées ainsi que les cas d'utilisation correspondants. C'est à partir de ces analyses qu'ont été conçus les prototypes d'interface-utilisateur soumis à l'évaluation. Ces analyses ont aussi permis d'élaborer les scénarios nécessaires à l'exécution des tests d'utilisabilité³⁸. Les premières solutions ont aussi été élaborées à partir des normes génériques reconnues, les normes internationales et celles qui sont spécifiques à la plateforme utilisée. Les différentes formes d'évaluation de l'utilisabilité ont alimenté ensuite les corrections au fil des itérations³⁹.

Au fur et à mesure que des principes de conception d'interface sont confirmés par les tests d'utilisabilité, des règles et des normes de conception sont ajoutés à un guide de style. Le guide de style est présenté sous une forme graphique et comprend les règles propres au projet. Le guide de style n'est publié qu'une fois les validations faites avec les utilisateurs⁴⁰.

6.3.1.3 Synthèse de la question 3

Pour l'équipe du programme, la conception des interfaces-utilisateurs doit se conformer à un « cadre de conception » que l'équipe du programme a élaboré avant le début de la définition des exigences. Les ergonomes s'inspirent de différentes sources pour leur conception dont la première : l'analyse du contexte d'utilisation. Un guide de style est aussi produit, mais à partir d'interfaces-utilisateurs validées avec les utilisateurs.

6.3.2 (4) L'approche itérative de ISO 13407 est-elle respectée lors de l'application du RUP© ?

L'approche itérative suppose une gestion des itérations et au premier chef une planification.

6.3.2.1 Équipe du programme : pas d'itération planifiée

Pour l'équipe du programme, une itération est le développement d'une partie d'une application réalisée de bout en bout, de la phase d'opportunité à la transition. Cependant, ce principe n'est pas appliqué durant l'étude de cas. Le développement est modulaire, mais on n'associe pas cette approche à une itération. Une itération 0 est cependant réalisée pour faire une preuve de concept⁴¹. À partir de quelques cas d'utilisation, un prototype a été construit et présenté à l'ensemble des parties prenantes.

Il n'y a pas de notion d'itération pour la spécification des interfaces-utilisateurs. Cependant, la multiplication des rencontres entre les analystes fonctionnels et les représentants des utilisateurs, combinée aux modifications apportées aux scénarimages qui font office de prototypes d'interface-utilisateur, trahissent l'existence d'une évolution itérative. Mais elle n'est pas planifiée⁴². Pour les analystes fonctionnels, « le contenu des scénarimages est négocié à la pièce avec les représentants des utilisateurs⁴³ ». Toutefois, la procédure de rédaction des cas d'utilisation prévoit, elle, un processus itératif qui se poursuit jusqu'à la « satisfaction des représentants des utilisateurs⁴⁴ ».

6.3.2.2 Intervention ergonomique : plusieurs itérations

Un plan de travail pour la conception des interfaces-utilisateurs a été produit et déposé. Le plan prévoyait deux cycles ergonomiques comprenant plusieurs itérations. Le premier portait sur une interface-utilisateur qui incluait 3 cas d'utilisation⁴⁵. Le deuxième cycle ajoutait à cette interface-utilisateur celles qui découlaient d'autres cas d'utilisation correspondant au contexte de la tâche⁴⁶. Des éléments de la navigation ont ensuite été définis. Cela s'est traduit par la nature de l'écran de consultation de l'application et par l'entête des écrans qui prévoyaient des menus pour accéder à d'autres écrans⁴⁷.

Les itérations, déterminées par différents types d'évaluation, ont entraîné l'évolution des modifications des interfaces-utilisateurs. Une douzaine d'itérations ont été faites sur des interfaces-utilisateurs, certaines étaient incrémentales. Par exemple, un écran de saisie pouvait être produit, suivi d'un écran de modification et d'un écran de consultation. 24 écrans et 189 spécifications ont été ainsi produits⁴⁸.

6.3.2.3 Synthèse de la question 4

L'équipe du programme ne planifie pas de cycle itératif. Cependant, les analystes doivent multiplier les rencontres avec les utilisateurs avant d'arriver à une solution acceptée par ces derniers. Les ergonomes présentent la planification de deux cycles d'un processus itératif incrémental.

6.3.3 (5) L'évaluation des solutions de conception retenue lors de l'application du RUP© est-elle conforme à celle qui est prescrite par ISO 13407 ?

L'évaluation des solutions de conception consiste à valider les exigences et à assurer une surveillance. Elle comprend, entre autres, la simulation de tâches par les utilisateurs qui permettent de recueillir leurs commentaires ou d'observer les défaillances de la solution, selon ISO 13407.

6.3.3.1 Équipe du programme : des revues

Pour la conception même des interfaces-utilisateurs, il n'y a pas de processus d'évaluation autres que des revues avec les analystes fonctionnels et les représentants des utilisateurs. Les activités

d'assurance-qualité prévoient essentiellement des révisions des scénarimages : une revue par les pairs et une revue par l'assurance-qualité.

Concrètement, cela se traduit par des rencontres entre les analystes fonctionnels et les représentants des utilisateurs et les analystes d'affaires au rythme de deux séances par semaine. L'architecte applicatif l'accompagne parfois. Ces séances consistent à réviser le travail fait et à obtenir des informations complémentaires des représentants des utilisateurs. Elles se poursuivent jusqu'à ce que les représentants des utilisateurs soient satisfaits. Puis, une vérification préliminaire de l'utilisateur, qui consiste en une relecture du scénarimage, est faite par les représentants des utilisateurs qui soumettent des corrections. Ensuite, le scénarimage est l'objet d'une revue par les pairs où sont exclus les représentants des utilisateurs. Des architectes, des analystes fonctionnels et des analystes qualité y participent. Les représentants des utilisateurs disposent ensuite des corrections soumises lors de la revue par les pairs et approuvent le scénarimage⁴⁹.

6.3.3.2 Intervention ergonomique : plusieurs types d'évaluation

Durant le cycle ergonomique, plusieurs types d'évaluation ont été utilisés. Les premières versions du prototype sont soumises aux évaluations selon les lignes directrices et, ensuite, à des évaluations heuristiques faites par les ergonomes⁵⁰.

Des tests d'utilisabilité formels et des évaluations à rabais ont ensuite été faits avec les représentants des utilisateurs et avec les utilisateurs finaux à partir de maquettes papier et avec des prototypes logiciels jetables. Des évaluations pas à pas ont aussi été faites avec les représentants des utilisateurs seuls, puis en présence des analystes fonctionnels dont les cas d'utilisation étaient touchés par la conception d'interface. Ces évaluations ont eu pour conséquence, outre l'amélioration des interfaces, de permettre aux analystes fonctionnels et aux représentants des utilisateurs d'ajouter des cas d'utilisation manquant au modèle des cas. Elles ont aussi permis de corriger le contenu des cas d'utilisation en cours de réalisation⁵¹.

Les séances de tests d'utilisabilité ont été faites lors d'un premier cycle avec les représentants des utilisateurs en présence de l'analyste d'affaire, puis avec l'utilisateur final sans autre présence extérieure⁵². Lors du deuxième cycle, les tests d'utilisabilité et les révisions pas à pas ont été faites en présence de l'analyste d'affaire et des analystes fonctionnels concernés⁵³.

6.3.3.3 Synthèse de la question 5

L'évaluation, pour l'équipe du programme, consiste en des revues des scénarimages avec les utilisateurs et par les pairs. Pour les ergonomes, les évaluations peuvent être heuristiques, pas à pas, de conformité aux lignes directrices, à rabais ou des tests d'utilisabilité formels.

6.4 (6) L'application du RUP© reconnaît-elle la nécessité d'un personnel qualifié selon ISO 13407 ?

Pour ISO 13407, la conception centrée sur l'utilisateur doit être pluridisciplinaire. Cela suppose la participation d'intervenants provenant de différents horizons aux qualifications diverses. Des spécialistes des sciences cognitives sont nommément identifiés.

6.4.1 Équipe du programme : des analystes fonctionnels

L'application initiale du RUP© faite par l'équipe du programme ne prévoyait pas l'embauche d'ergonomes. C'est à l'initiative de la direction responsable des représentants des utilisateurs et des analystes d'affaires, celle qui représentait le client, que les ergonomes ont été recrutés et embauchés. Ces derniers ont d'ailleurs été sous leur juridiction durant l'intervention ergonomique⁵⁴.

Les lignes directrices de conception d'interface-utilisateur et le « Cadre de conception des interfaces WEB » ont été produits sous la responsabilité d'un architecte au début du projet⁵⁵. Les interfaces-utilisateurs étaient conçues par les analystes fonctionnels responsables de la rédaction des cas d'utilisation. Leurs qualifications correspondaient à celles du spécificateur de cas d'utilisation et du concepteur de test définies par le RUP©, puisqu'ils avaient à jouer ces deux rôles⁵⁶. Ils n'avaient pas de qualifications liées à la conception d'interfaces, quoique certains en aient réalisées par le passé. Les interfaces-utilisateurs sont créées par eux et soumises, lors de réunions, aux représentants des utilisateurs et aux analystes d'affaires⁵⁷.

6.4.2 Intervention ergonomique : qualifications en ergonomie et en informatique

L'intervention est faite par deux ergonomes qualifiés. Les deux ont une formation universitaire en informatique et en ergonomie du logiciel. L'ergonome principal a de plus une formation universitaire en psychologie⁵⁸. Les services d'un graphiste ont été retenus

afin de finaliser la facture graphique de l'application⁵⁹. Un informaticien s'est joint à l'équipe afin de vérifier, entre autres, la faisabilité technique des spécifications d'interfaces susceptibles de poser des problèmes de réalisation⁶⁰.

6.4.2.1 Synthèse de la question 6

Pour l'équipe du programme, les mêmes personnes produisent les cas d'utilisation et les interfaces-utilisateurs. Il n'y a pas de qualifications requises. Les ergonomes ont une formation spécialisée en ergonomie du logiciel et font appel à un graphiste pour l'apparence et un informaticien pour vérifier la faisabilité technique. Les connaissances et les qualifications des autres parties prenantes sont aussi sollicitées.

6.5 Comparaison entre ISO 13407 et les autres approches

ISO 13407 édicte des principes et des activités pour définir la conception centrée sur l'utilisateur. Ces principes et activités ont été ici reformulés en 6 questions de recherche. Le RUP©, un processus de développement centré sur l'architecture, évoque la question de la conception centrée sur l'utilisateur en associant certaines de ses activités aux principes de la norme ISO 13407. L'équipe du programme Neptune n'a pas cherché à être fidèle au principe de la conception centrée sur l'utilisateur mais elle a adapté le RUP© à ses besoins pour le développement de ses applications. L'intervention ergonomique s'est faite selon les règles de l'art fixées par la conception centrée sur l'utilisateur en respectant les contraintes de l'adaptation faite du RUP© lui fixait.

Une synthèse des caractéristiques de ces quatre approches apparaissent dans le tableau 6.1 qui récapitule les différences et similarités de conception centrée sur l'utilisateur entre ISO 13407, le RUP©, l'application faite par l'équipe du programme et l'intervention ergonomique pour chaque question de recherche.

Tableau 6.1 : Comparaison entre ISO 13407 et les différentes approches.

Question	ISO 13407	RUP©	Équipe du programme	Intervention ergonomique
(1) Est-ce que l'application du RUP© soutient la participation active des utilisateurs et définit le contexte d'utilisation selon ISO 13407 ?				
Participation des utilisateurs	• Obligatoire • Activités variées	• Souhaitable • Activités variées	• Obligatoire • Ateliers de travail	• Obligatoire • Activités variées
Contexte d'utilisation				
Utilisateur	• Description élaborée	• Décrits à haut niveau	• Non décrits	• Description élaborée
Environnement	• Description élaborée	• Description à haut niveau du système	• Partielle	• Description élaborée
Tâche	• Analyse contextuelle de la tâche	• Cas d'utilisation	• Cas d'utilisation	• Analyse contextuelle de la tâche
(2) L'application du RUP© permet-elle une bonne spécification des exigences-utilisateurs notamment par une bonne répartition des fonctions entre les utilisateurs et la technologie selon ISO 13407 ?				
	• Répartition des fonctions • Exigences-utilisateurs	• Pas de répartition des fonctions • Exigences d'automatisation • Exigences d'utilisabilité	• Pas de répartition des fonctions • Exigences d'utilisabilité à haut niveau	• Répartition des fonctions • Incorporés à la spécification d'interface
(3) La matérialisation des solutions de conception se fait-elle par l'application du RUP© selon ISO 13407 ?				
	• Contexte d'utilisation • Autres connaissances • Évolue avec évaluations	• Normes et guides	• Normes et guides • Interface-utilisateur associée à un cas d'utilisation	• Contexte d'utilisation • Autres connaissances • Évolue avec évaluations
(4) L'approche itérative de ISO 13407 est-elle respectée lors de l'application du RUP© ?				
	• Planifiée • Intégrée à la conception	• Non planifiée • Lorsque besoins changeants ou problème complexe	• Non planifiée	• Planifiée • Intégrée à la conception
(5) L'évaluation des solutions de conception retenue lors de l'application du RUP© est-elle conforme à celle qui est prescrite par ISO 13407 ?				
	• Liée à la conception • Surveillance à long terme • Planifiée • Plusieurs types	• Accord des parties prenantes • Pallier les défaillances • Plusieurs types • Participation d'expert	• Accord des parties prenantes • Pallier les défaillances • Revues	• Liée à la conception • Plusieurs types d'évaluation
(6) L'application du RUP© reconnaît-elle la nécessité d'un personnel qualifié selon ISO 13407 ?				
	• Obligatoire • Expert en facteur humain • Équipe distincte	• Facultatif mais recommandé • Graphiste facultatif • Équipe distincte pour grands projets	• Pas de qualification • Pas d'équipe distincte	• Obligatoire • Expert en facteur humain • Équipe distincte

Notes et références

¹ *Analyse des fiches signalétiques des artefacts pertinents et Compte rendu d'entrevue avec analyste d'affaires et représentant d'utilisateur.*

² C'était le cas pour un autre projet. Les représentants des utilisateurs étaient des surnuméraires embauchés pour leur qualification à comprendre le domaine d'affaires. Selon nos observations et le *Compte rendu d'entrevue avec analyste d'affaires et représentant d'utilisateur*.

³ *Compte rendu d'entrevue avec analyste d'affaires et représentant d'utilisateur.*

⁴ *Ibidem.*

⁵ *Ibidem.*

⁶ Nos observations. À titre d'ergonome, nous avons demandé aux directeurs de service de nous transmettre le document Vision afin de faciliter l'analyse du contexte d'utilisation. Ils nous ont confirmé ne pas avoir produit ce document. Le contexte dans les différentes fiches signalétiques ne font pas apparaître cet artefact.

⁷ *Fiche signalétique - Registre des caractéristiques logicielles fonctionnelles et Fiche signalétique - Registre des caractéristiques logicielles non fonctionnelles.*

⁸ Ni le document Vision, ni la définition du contexte n'apparaissent dans le guide *Façon de faire - Vue globale* qui identifie les artefacts du processus.

⁹ L'environnement technologique peut être déduit à partir du document *Architecture de référence*. Les autres éléments de l'environnement et l'analyse de la tâche étaient des notions étrangères à l'équipe de projet selon nos observations.

¹⁰ Dans *Architecture de référence*, la description des interfaces avec les autres systèmes donne un aperçu de l'environnement technologique.

¹¹ *Façon de faire - Vue globale*, p. 13. Les demandes des parties prenantes prévues par le RUP© n'apparaissent pas dans la liste des artefacts.

¹² *Fiche signalétique - Modèle de cas d'utilisation, Guide de détail de cas d'utilisation et Fiche signalétique - Détail de cas d'utilisation.* Ils apparaissent effectivement dans les artefacts.

¹³ La *Fiche signalétique - Modèle des concepts-clés* prévoit une identification des acteurs. Par contre, l'artéfact du *Modèle des concepts clés* n'identifie pas d'acteur.

¹⁴ *Modèle des cas d'utilisation - Réseau de distribution - Version 5.*

¹⁵ *Fiche signalétique - Détail de cas d'utilisation*, p. 7.

¹⁶ *Compte rendu d'entrevue avec analyste d'affaires et représentant d'utilisateur.*

¹⁷ *Fiche signalétique - Registre des caractéristiques logicielles fonctionnelles.*

¹⁸ *Ibidem*, p. 3.

¹⁹ *Fiche signalétique - Modèle de cas d'utilisation*, p. 3.

²⁰ Nos observations, le *Compte rendu d'entrevue avec analyste d'affaires et représentant d'utilisateur* et le *Guide de scénarimage* le confirme.

²¹ *Fiche signalétique - Détail de cas d'utilisation*, p. 7.

²² *Guide de détail de cas d'utilisation*, p. 5. On entend ici, par tâche d'affaires, le DPO qui est un processus d'affaires et non une tâche d'un utilisateur. Les DPO, *DPO - Émission d'un numéro de détaillant*, *DPO - Annulation d'un numéro de détaillant*, *DPO - Consultation du dossier détaillant* et *DPO - Modification du dossier détaillant* en sont des exemples.

²³ *Rapport hebdomadaire de l'ergonomie des interfaces, Planification des activités ergonomiques des quatre premières semaines, Planification des activités ergonomiques 2e cycle.*

²⁴ *Rapport hebdomadaire de l'ergonomie des interfaces et Intervention en ergonomie - Bilan du premier cycle.*

²⁵ *Analyse de contexte d'utilisation*, p. 7-16.

²⁶ *Ibidem*, p. 17-22.

²⁷ *Ibidem*, p. 24.

²⁸ *Ibidem*, p. 23.

²⁹ *Ibidem*, p. 24.

³⁰ *Ibidem.*

- ³¹ *Fiche signalétique - Registre des caractéristiques logicielles fonctionnelles et Fiche signalétique - Registre des caractéristiques logicielles non fonctionnelles.*
- ³² *Fiche signalétique - Registre des caractéristiques logicielles fonctionnelles, p. 4.*
- ³³ *Registre des caractéristiques logicielles non fonctionnelles et nos observations.*
- ³⁴ *Nos observations.*
- ³⁵ *Rapports hebdomadaires de l'ergonomie des interfaces, p. 6-7 et nos observations.*
- ³⁶ *Fiche signalétique – Scénarimage, p. 5, et Compte rendu d'entrevue avec analyste fonctionnel, p. 3.*
- ³⁷ *Cadre de conception des interfaces WEB.*
- ³⁸ *Intervention en ergonomie - Bilan du premier cycle, p. 5 et 9.*
- ³⁹ *Ibidem, p. 12.*
- ⁴⁰ *Guide de style des interfaces Web.*
- ⁴¹ *Compte rendu d'entrevue avec chef d'équipe de gestion du changement, p. 1.*
- ⁴² *Fiche signalétique – Scénarimage, p. 6, et Compte rendu d'entrevue avec analyste fonctionnel, p. 3.*
- ⁴³ *Rencontre avec analystes fonctionnels sur bilan de cycle ergonomique, p. 1.*
- ⁴⁴ *Compte rendu d'entrevue avec analyste fonctionnel.*
- ⁴⁵ *Planification des activités ergonomiques des quatre premières semaines p. 1 et Intervention en ergonomie - Bilan du premier cycle, p. 7.*
- ⁴⁶ *Planification des activités ergonomiques 2^e cycle.*
- ⁴⁷ *Spécification d'interface - Zone de navigation primaire.*
- ⁴⁸ *Spécification d'interface - Dossier détaillant et Intervention en ergonomie - Bilan du premier cycle, p. 12 et 14.*
- ⁴⁹ *Compte rendu d'entrevue avec analyste fonctionnel et Fiche signalétique – Scénarimage, p. 8.*
- ⁵⁰ *Rapports hebdomadaires de l'ergonomie des interfaces, p. 3 et observations.*
- ⁵¹ *Intervention en ergonomie - Bilan du premier cycle, p. 12 et 14, Compte rendu du pré test d'utilisabilité Cycle 2 Itération 1, p. 1, Compte rendu du test d'utilisabilité Cycle 2 Itération 1, p. 1 et 2,*

⁵² *Intervention en ergonomie - Bilan du premier cycle*, p. 12, et *Analyse de contexte d'utilisation*, p. 36.

⁵³ *Rapports hebdomadaires de l'ergonomie des interfaces* p. 12, 13 et 14.

⁵⁴ Organigramme du projet et nos observations.

⁵⁵ *Cadre de conception des interfaces WEB*.

⁵⁶ *Compte rendu d'entrevue avec analyste d'affaires et représentant d'utilisateur* et selon l'analyse des auteurs des artefacts produits. L'équipe de projet a d'ailleurs demandé aux ergonomes de faire une formation de conception d'interfaces-utilisateurs pour tenter de combler la carence : *Formation - Scénarimage et ergonomie*.

⁵⁷ *Rapports hebdomadaires de l'ergonomie des interfaces*, p. 10. À l'occasion de rencontres avec les analystes les insuffisances de qualifications en conception d'interfaces-utilisateurs étaient non équivoques. Les notions en ergonomie étaient inconnues.

⁵⁸ *Intervention en ergonomie - Bilan du premier cycle*, p 1.

⁵⁹ *Rapports hebdomadaires de l'ergonomie des interfaces*, p. 10.

⁶⁰ *Étude des questions techniques et ergonomiques*.

CHAPITRE VII

DISCUSSION

Les explications et tendances qui peuvent être extraites des données présentées au précédent chapitre sont abordées ici. Pour chaque question de recherche, les aspects pertinents relevant de la norme ISO 13407, du RUP©, de l'application faite par l'équipe du programme et par l'intervention ergonomique. On pourra ainsi apprécier dans quelle mesure une application faite du RUP© se rapproche ou s'éloigne de la conception centrée sur l'utilisateur.

7.1 (1) Est-ce que l'application du RUP© soutient la participation active des utilisateurs et définit le contexte d'utilisation selon ISO 13407 ?

La participation active des utilisateurs doit être directe et la spécification du contexte d'utilisation suppose une description de l'environnement, des utilisateurs et de la tâche.

Le RUP© prévoit que des éléments du contexte, la description de l'environnement et des utilisateurs sont identifiés dans les artefacts Vision métier et Vision. Cependant, ces deux documents servent à définir ce qu'on appelle en génie logiciel le domaine du problème et à fournir une spécification des exigences à haut niveau. Ces éléments du contexte qu'ils décrivent peuvent être pertinents à la conception centrée sur l'utilisateur même s'ils sont définis pour identifier le contexte du problème et non le contexte d'utilisation. Cependant, l'équipe du programme n'a pas jugé pertinent de produire ces documents. Cela a eu pour conséquence qu'aucune forme de contexte n'a été identifiée.

7.1.1 La participation des utilisateurs

Pour le RUP©, la présence des utilisateurs est souhaitable. Pour l'équipe du programme, la participation des utilisateurs a été systématique. Par contre, la nature de la participation consistait en des séances d'information et à valider les artefacts produits par des révisions conjointes de document. C'est cela qui faisait dire à Gulliksen, Göransson et Lif (2001) que le RUP© pouvait provoquer une confusion sur le rôle du représentant de l'utilisateur entre expert du domaine et utilisateur final.

L'intervention ergonomique, pour sa part, a varié la nature des activités par leur participation à la définition du contexte d'utilisation, aux analyses de la tâche, la planification et au suivi des

activités de conception ou par les types d'évaluation utilisés. Les participants ont ainsi été amenés à envisager les exigences sous différentes perspectives, ce qui a facilité les échanges et la compréhension des positions de chacun en enrichissant la participation et l'implication des utilisateurs.

7.1.2 L'environnement

L'absence de document Vision métier et Vision a fait que l'environnement n'a pas été identifié par l'équipe du programme. Certains éléments se retrouvaient dans un document d'architecture. Ce n'est cependant pas au document d'architecture de décrire l'environnement qui apparaît normalement dans les documents Vision. Les analystes responsables de la production des interfaces-utilisateurs n'ont donc pas identifié l'environnement dans lequel le système devait évoluer. Pour les ergonomes, il n'y a pas eu de problème à colliger les informations nécessaires à l'identification de l'environnement et des équipements.

7.1.3 Les utilisateurs

L'absence de document Vision métier et Vision a limité l'identification des utilisateurs à celle des acteurs sans autre description qu'un résumé de leur rôle comme le prévoit l'activité qui consiste à trouver les acteurs et les cas d'utilisation pour obtenir le modèle de cas d'utilisation. Les acteurs sont des catégories d'utilisateur. Ces données sont évidemment insuffisantes pour réaliser les interfaces-utilisateurs. Les ergonomes ont donc relevé toutes les informations nécessaires pour identifier les utilisateurs et leurs caractéristiques.

On peut s'interroger par ailleurs, sur la pertinence d'inscrire la description détaillée des utilisateurs dans un document Vision qui, pour les autres aspects, vise à décrire à haut niveau le système à construire. Demander au concepteur d'interface d'y inscrire ces caractéristiques est plutôt dérisoire, d'autant qu'il n'a pas la responsabilité de l'artefact.

7.1.4 La tâche

Le contexte de la tâche est une notion qui n'existe pas pour le RUP© pour qui le cas d'utilisation pourrait en faire office. Le cas d'utilisation peut être interprété comme une modélisation de la tâche d'utilisation du nouveau système, mais pas comme un contexte de la tâche de l'utilisateur. Or, pour ISO (1999), « il est primordial de comprendre et d'identifier ce contexte en détail, afin de guider les premières décisions de conception et définir une base pour l'évaluation ». Pour

l'équipe du programme, le contexte de la tâche n'a donc jamais été défini. L'intervention ergonomique a donc dû définir le contexte d'utilisation, notamment celui de la tâche, indépendamment du processus de développement.

Ainsi, l'équipe du programme a été respectueuse du processus proposé par le RUP©. Elle a produit des cas d'utilisation inspirés des caractéristiques logicielles fonctionnelles qui correspondent aux caractéristiques de produit du document Vision et des DPO qui correspondent au cas d'utilisation-métier. L'équipe du programme prévoyait ainsi que les interfaces-utilisateurs devaient correspondre aux cas d'utilisation sans égard à la tâche de l'utilisateur.

Pour le RUP©, la tâche correspond aux cas d'utilisation et les scénarimages constituent une modélisation de la tâche. Dans les faits, le cas d'utilisation correspond à l'utilisation du système en élaboration faite par un acteur qui est une catégorie d'utilisateur proposée par l'analyste fonctionnel. C'est une description dynamique du système qui prend sa source dans les caractéristiques de produit définies dans le document Vision et dans les cas d'utilisation-métier. Or, les cas d'utilisation-métier sont, eux, des processus d'affaires.

ISO 13407 (1999) jugeait pertinent de préciser que la description de la tâche devait distinguer « la répartition des activités et des phases opérationnelles entre les composantes humaines et techniques ». Et la norme ajoutait : « Il convient de ne pas décrire les tâches uniquement en termes de fonctions ou d'attributs d'un produit ou système. » C'est ce que ne fait pas le cas d'utilisation. Quant aux scénarimages, ils constituent une représentation graphique des flots d'événements décrits dans les cas d'utilisation et une identification à haut niveau des éléments du prototype d'interface-utilisateur. On peut parler de modélisation de la tâche, mais c'est la tâche d'utilisation du système en élaboration. Ce n'est pas la tâche courante de l'utilisateur. C'est ce que soulignaient Gulliksen, Göransson et Lif (2001) lorsqu'ils déploraient le manque de soutien du processus pour l'analyse de la tâche de l'utilisateur.

Pour les ergonomes, les cas d'utilisation sont certes pertinents pour décrire le système. Mais l'analyse contextuelle de la tâche est la référence constante à partir de laquelle sont prises les décisions de conception de l'interface-utilisateur. Ainsi, une interface-utilisateur peut correspondre à plusieurs cas d'utilisation et certaines autres à une partie de cas d'utilisation. Et, parfois, l'analyse de la tâche met en évidence certaines exigences qui nécessitaient l'ajout de nouveaux cas d'utilisation.

7.2 (2) L'application du RUP© permet-elle une bonne spécification des exigences-utilisateurs notamment par une bonne répartition des fonctions entre les utilisateurs et la technologie selon ISO 13407 ?

La répartition des exigences entre les utilisateurs et la technologie est une notion qui n'existe pas pour le RUP© et que l'équipe projet ignorait aussi. Les exigences-utilisateurs ou exigences liées à l'utilisateur et à l'organisation ne sont pas une notion reconnue par le RUP©. Par contre, les exigences d'automatisation sont celles qui guident la modélisation des cas d'utilisation. Le RUP© retient une méthode d'analyse « orientée problème » qui consiste à modéliser le « domaine » avec des cas d'utilisation-métier à partir desquels on extrait les exigences d'automatisation qui constituent la solution.

L'équipe du programme a reproduit cette approche. L'exemple de la conception d'une interface-utilisateur pour la saisie d'adresse démontre que l'adaptation est d'abord définie en fonction de la technologie disponible plutôt qu'en fonction du contexte d'utilisation. L'analyste souhaitait fournir ainsi les fonctionnalités définies par des normes nationales guidant le format des adresses plutôt qu'en fonction de la pratique des utilisateurs, de leur capacité de saisir l'information correspondante et des besoins réels de l'entreprise. Cela confirme la constatation que Gulliksen, Göransson et Lif (2001) font lorsqu'ils déplorent le manque d'engagement à l'utilisabilité du RUP© par delà la conception des interfaces utilisateurs.

7.3 (3) La matérialisation des solutions de conception se fait-elle par l'application du RUP© selon ISO 13407 ?

Les solutions de conception doivent s'alimenter d'abord du contexte d'utilisation et des autres connaissances acquises. Or, l'équipe du programme ne prévoyait pas d'analyse du contexte d'utilisation.

L'équipe du programme a produit durant la phase d'opportunité le guide appelé « Cadre de conception des interfaces Web » comme le prescrit le RUP©. Ce guide propose des solutions d'interface-utilisateur tellement détaillées qu'elles forcent le concepteur à nier le contexte dans lequel cet interface-utilisateur doit être définie. Il présupait ainsi des solutions qui seraient éventuellement retenues et qui n'avaient pas été évaluées avec les utilisateurs. En l'absence d'analyse de contexte, les analystes devaient s'en remettre à ce guide et à leur expérience passée en conception d'interface-utilisateur, s'ils en avaient une. Enfin, ce type de guide laisse croire au

concepteur qu'il existe des solutions toutes faites à la conception des interfaces-utilisateurs. Gulliksen, Göransson et Lif (2001) y voyait là une faiblesse importante du RUP©

Il y a, par ailleurs, une confusion sur la nature des scénarimages. Göransson, Gulliksen et Lif (2001) croient qu'il est absolument nécessaire de compléter les cas d'utilisation avec une quelconque forme de prototype d'interface-utilisateur pour illustrer comment le système peut soutenir les utilisateurs dans leur travail.

Scénarimager est une méthode d'explicitation des exigences proposé tant par Kruchten (2001) que par Leffingwell et Widrig (2000) pour illustrer celles-ci à l'utilisateur afin de faciliter les échanges. Par contre, Kruchten (2001) et Ahlqvist, Bylund et Kruchten (2001) utilisent le même terme, le scénarimage, pour désigner une modélisation de l'interface-utilisateur sans qu'il n'y ait, incidemment, d'illustration d'écran.

Dans l'étude de cas, on peut constater que la solution retenue pour l'adaptation du RUP© est une hybridation des deux approches. Les scénarimages étaient systématiquement associés à un cas d'utilisation sans égard au contexte de la tâche, comme l'avaient constaté, par ailleurs, Gulliksen, Göransson et Lif (2001). Il servait à la fois de prototype d'interface d'utilisateur et de modèle d'interface.

Les itérations sur un prototype d'interface répondent parfaitement à l'objectif d'illustrer les exigences tel que scénarimager lors de l'explicitation. L'expérience du cas démontre que l'utilisateur en est pleinement satisfait. C'est à ce propos que Gulliksen, Göransson et Lif (2001) insistaient sur la nécessité de compléter le cas d'utilisation avec un prototype d'interface-utilisateur pour représenter la complexité du système. Gulliksen et al. (2005) rappelle de plus que le langage UML est souvent inaccessible aux utilisateurs et que le prototype d'interface-utilisateur est beaucoup plus apprécié.

7.4 (4) L'approche itérative de ISO 13407 est-elle respectée lors de l'application du RUP© ?

RUP© reconnaît l'importance des itérations. C'est d'ailleurs le fondement du processus. Il reconnaît aussi plusieurs types d'itérations dont celui qui est prescrit par ISO 13407. Pour la conception d'interface-utilisateur, notamment pour la conception de prototype d'interface, il reconnaît une évolution itérative du prototype. Cependant, la notion d'itération pour le RUP©

correspond aux approfondissements successifs ou à la couverture : croître de façon incrémentale d'itération en itération (John, Bass et Adams, 2003).

Dans l'article *User Interface Design in the Rational Unified Process* (Kruchten, Ahlqvist et Bylund, 2001), on reconnaît un processus itératif entre le cas d'utilisation, le scénarimage et le prototype d'interface-utilisateur. Mais le RUP© ne prescrit pas la planification formelle de ces itérations. De la même façon, l'équipe du programme ne prévoyait pas de cycles itératifs : le cas d'utilisation était produit, puis le scénarimage. Dans les faits, les analystes fonctionnels ont produit les scénarimages parallèlement aux cas d'utilisation. Par contre, les ergonomes présentaient une planification des itérations qui correspondait au travail à faire et à la durée réelle du processus.

Gulliksen, Göransson et Lif (2001) mentionnaient qu'il pourrait être raisonnable d'avoir des cycles itératifs de 5 jours : 2 jours d'analyse, 2 jours de conception et 1 jour d'évaluation. Lorsqu'il y a plus de temps alloué à ce sous-processus, il y a un grand risque de syndrome du bébé : une défense exagérée d'une solution proposée. Le plus grand défi qui se pose aux spécialistes de l'utilisabilité est d'enseigner et de promouvoir une conception itérative de l'interface-utilisateur qui est effective.

7.5 (5) L'évaluation des solutions de conception retenue lors de l'application du RUP© est-elle conforme à celle qui est prescrite par ISO 13407 ?

Bien que le RUP© évoque un certain nombre de modes d'évaluation d'interface-utilisateur, il ne propose pas une planification de ces tests. Si on s'en tient à la façon qu'ils sont présentés, ils sont assimilables à d'autres tests conçus par un concepteur de tests et exécutés par un testeur. Certes, la notion de rôle joué par un travailleur n'exclut pas que ce soit le concepteur d'interface-utilisateur qui assume ces responsabilités lors de la production des solutions de conception, mais cela n'est pas précisé clairement. Par ailleurs, on ne distingue pas clairement les tests d'utilisabilité liés à la conception de l'interface à ceux qui sont liés à la surveillance. Ces derniers tests existent si on en juge par les gabarits proposés pour l'artefact du plan de test.

ISO 13407 évoque le prototypage, les simulations, la modélisation et les maquettes à améliorer par les tests avec les utilisateurs à différentes étapes de la conception (John, Bass et Adams, 2003). Pour l'équipe du programme, l'évaluation des interfaces-utilisateurs reposait sur des revues : des rencontres des analystes fonctionnels avec les représentants des utilisateurs et des

des tests d'utilisabilité ont été évoqués par les ergonomes, l'équipe de l'assurance qualité y a vu une intrusion dans sa juridiction. Une description de la nature et du rôle des évaluations de conception en conception centrée sur l'utilisateur a été nécessaire pour rassurer les parties prenantes.

7.6 (6) L'application du RUP© reconnaît-elle la nécessité d'un personnel qualifié selon ISO 13407 ?

L'équipe du programme ne reconnaissait pas la nécessité d'avoir une équipe de personnes qualifiées pour produire les interfaces-utilisateurs. Ce sont d'ailleurs les utilisateurs qui ont fait appel aux ergonomes, pas les informaticiens. Pour, l'équipe du programme la conception des interfaces-utilisateurs étaient la responsabilité d'analystes fonctionnels. C'est exactement la lacune qu'avaient identifiée Gulliksen, Göransson et Lif (2001).

Le RUP© reconnaît l'existence d'experts en conception d'interface-utilisateur et concède même qu'un non-informaticien puisse être utile à cette activité. Dans ce dernier cas, cependant, il évoque des qualifications en graphisme plutôt qu'en ergonomie. Mais il n'insiste pas sur la diversification des compétences comme l'avaient John, Bass et Adams (2003). Il est donc normal que, lors de l'application du processus, les équipes de projet n'y voient pas là une nécessité.

Selon, John, Bass et Adams (2003), ISO 13407 réclame l'assurance qu'un personnel qualifié soit embauché pour exécuter les activités requises et que des méthodes appropriées soient utilisées. Or, toujours selon John, Bass et Adams (2003), le RUP© ne pose pas de conditions sur les qualifications du personnel ou sur les méthodes, en presumant que toute l'équipe de développement est qualifiée pour exécuter toutes les activités nécessaires.

Le personnel impliqué dans l'intervention ergonomique était qualifié en informatique et en sciences cognitives; les services d'un graphiste ont été retenus. Et lors de l'intervention ergonomique, les parties prenantes ayant un intérêt dans les interfaces-utilisateurs ont été invités à participer à la conception.

CONCLUSION

Une étude de cas unique invite à une certaine prudence sur les conclusions qu'on peut tirer des observations faites. On ne peut pas se fonder sur cette étude pour confirmer ou infirmer la valeur de la conception centrée sur l'utilisateur, par exemple. On ne peut pas généraliser à tout le génie logiciel les constatations faites pour le RUP©. Les conclusions qu'on peut en tirer pour le processus RUP© doivent être limitées à l'application qu'en a faite une équipe du programme.

Le RUP© est un processus centré sur l'architecture. Il n'a pas la prétention d'être centré sur l'utilisateur. On constate néanmoins qu'il évoque les principes de la conception centrée sur l'utilisateur. Cependant, il n'en fait pas un enchaînement d'activités distinct. Certains principes sont mal compris et d'autres mal intégrés.

L'étude de cas démontre qu'une équipe de projet peut appliquer le processus, tel que formulé, sans avoir à rencontrer les principes de conception centrée sur l'utilisateur. L'intervention ergonomique de l'étude de cas démontre aussi, par contre, qu'une conception centrée sur l'utilisateur peut être compatible avec le RUP©.

Göransson, Gulliksen et Lif (2001) signalent que les organisations qui font du développement à l'interne décident d'acheter le RUP© pour d'autres raisons que la conception centrée sur l'utilisateur. Ces entreprises évoquent plutôt que le RUP© est devenu une norme, qu'il facilite l'embauche de consultants externes, qu'il bénéficie d'un support technique, qu'il repose sur des techniques orientées objet dont l'utilisation se justifie par plusieurs raisons et que des outils et de la formation sont disponibles.

La conception centrée sur l'utilisateur offre une méthode de validation des exigences-utilisateurs comme l'ont souligné John, Bass et Adams (2003). La gestion des exigences est un problème majeur qui hypothèque la réussite de plusieurs projets logiciels. Comme dans le cas des systèmes interactifs, les exigences-utilisateurs constituent une part substantielle des exigences d'un système, la conception centrée sur l'utilisateur devient un atout à la réussite d'un projet. Il y aurait lieu d'approfondir la question de la validation des exigences faite lors de la définition des exigences sous la perspective de la conception centrée sur l'utilisateur.

L'introduction des cas d'utilisation dans les processus d'analyse orientée objet peut laisser croire que ces derniers peuvent éliminer la distance qui existe entre les informaticiens et les

utilisateurs. Les cas d'utilisation sont une description détaillée et dynamique d'un système et en donnent certes une image plus concrète à l'utilisateur. Mais ils ne correspondent pas au contexte d'utilisation nécessaire à la conception centrée sur l'utilisateur. Ivar Jacobson, lui-même, a reconnu que les cas d'utilisation ne suffisaient pas et a manifesté une grande sensibilité aux principes de la conception centrée sur l'utilisateur. L'objectif des cas d'utilisation est de décrire toutes les fonctionnalités d'un système afin d'obtenir des résultats. L'analyse du contexte de la tâche a plutôt pour objectif de hiérarchiser les tâches selon leur criticité afin d'optimiser la performance de l'utilisateur. Sans être incompatibles, ces deux objectifs ne sont pas toujours convergents. Il apparaît souhaitable de conserver les deux approches : le contexte du problème et le contexte d'utilisation comme le proposent Sousa et Furtado (2003b). Enfin ces cas d'utilisation deviennent une obsession qui, comme le soulignaient Gulliksen et al. (2005), éloignent le développeur des besoins réels de l'utilisateur.

L'intervention ergonomique de l'étude de cas démontre qu'il est possible d'introduire une conception centrée sur l'utilisateur sans trop modifier le processus. Il y a certes quelques artefacts à modifier ou à remplacer. Il faut une équipe distincte et qualifiée. Il faudrait que le RUP© reconnaisse l'enchaînement d'activités de conception d'interface-utilisateur plus formellement. Dans une architecture mult niveau, la couche présentation se distingue des autres. Ainsi, la couche présentation qui comprend les interfaces utilisateurs se distingue, par exemple, de la couche des traitements ou de la couche de persistance des données. Il faudrait qu'on reconnaisse que sa spécification et sa conception se distingue autant.

John, Bass et Adams (2003) indiquent qu'il y a aussi des problèmes de terminologie. Ainsi, les tests d'utilisabilité de la conception centrée sur l'utilisateur ne sont pas les tests qu'on conçoit en génie logiciel. La conception centrée sur l'utilisateur relève plus de la spécification que la conception telle que comprise en génie logiciel. L'itération a un sens précis. Le contexte de la tâche ne doit pas être confondu avec la modélisation de la tâche. Une harmonisation des terminologies ou, à tout le moins, une reconnaissance de ces distinctions serait appropriée.

Gloria Gery (1997) constate que les développeurs ont une vision des systèmes qui est orientée sur les données. Selon elle, même lorsque les analystes se rendent sur le terrain, c'est avec la perspective du traitement des données. De la tâche, ils ne retiennent que l'ordre dans lequel sont saisies les données. Gulliksen et al. (2005) abondaient dans le même sens en soulignant que les développeurs étaient plus préoccupés par l'implémentation de toutes les fonctionnalités plutôt

que par l'utilisabilité. Selon Lévesque (1998), l'analyse de système ne doit pas se limiter à l'inventaire des données et aux traitements, elle doit aussi tenir compte du contexte de la tâche et des connaissances de l'utilisateur.

La conception centrée sur l'utilisateur proposée par ISO 13407 ne se limite pas à la seule conception de l'interface-utilisateur comme le laisse croire le RUP®. Toute la gestion des exigences d'un système interactif doit être envisagée pour répondre aux prescriptions de la norme comme le rappelait la norme comme le rappelaient Sousa et Furtado (2003) et Earthy, Sherwood-Jones et Bevan (2001).

Ainsi, les techniques d'analyse proposées par le génie logiciel ne sont pas suffisamment complètes pour tenir compte de l'importance de l'utilisabilité d'un système. À ce titre, il y aurait lieu d'identifier ce qui distingue l'analyse du domaine du problème en génie logiciel de la compréhension du contexte d'utilisation en conception centrée sur l'utilisateur comme le relevait Judy Brown (1997). Il n'est pas suffisant de s'assurer qu'un système ait toutes les fonctions désirées par un client, encore faut-il que ces fonctions soient utilisables.

RÉFÉRENCES

- ABRAN, Alain, MOORE, James W., BOURQUE, Pierre, DUPUIS, Robert, TRIPP, Leonard. 2004. "The guide to the Software Engineering Body of Knowledge". *The guide to the Software Engineering Body of Knowledge, 2004 version, Swebok* . Décembre. IEEE Computer Society: Los Alamitos, CA.
- BOEHM, Barry W. 1988. "A Spiral Model of Software Development and Enhancement". *IEEE Computer* . Mai. IEEE Computer Society: Los Alamitos, CA. 21(5).61-72.
- BROWN, Judy. 1997. "HCI and Requirements Engineering - Exploring Human-Computer Interaction and Software Engineering Methodologies for the Creation of Interactive Software". *ACM SIGCHI Bulletin*. Janvier. ACM Special Interest Group on Computer-Human Interaction: Minneapolis. Janvier. 32 - 35 .
- COCKBURN, Alistair. 2001. *Writing effective use cases*. Addison-Wesley: Boston.
- CONSTANTINE, Larry L., LOCKWOOD, Lucy A. D. 2001. "Structure and Style in Use Cases for User Interface Design". *Object modeling and user interface design*. Eds. Addison-Wesley: Boston. 245-279.
- CUNNINGHAM, J. Bart. 1997. "Case study principles for different types of cases". *Quality and quantity*. Novembre. Kluwer Academic Publishers: Dordrecht, Pays-Bas. 31(4).401-423.
- DIAPER, Dan. 1997. "Integrating HCI and Software Engineering Requirements Analysis: A Demonstration of Task Analysis Supporting Entity Modeling HCI and Requirements Engineering". *ACM SIGCHI Bulletin*. Janvier. ACM Special Interest Group on Computer-Human Interaction: Minneapolis. Janvier. 41-50.
- EARTHY, Jonathan. 2000. *Human Factors Integration Capability Maturity Model - reference model*. Juin. ISO (The International Organization for Standardization): London, UK.
- EARTHY, Jonathan, SHERWOOD-JONES, Brian, BEVAN, Nigel. 2001. "The improvement of human-centred processes-facing the challenge and reaping the benefit of ISO 13407". *International Journal of Human Computer Studies*. Octobre. Academic Press: London, UK. 553-585.
- ELKOUTBI, Mohammed, KHRISS, Ismaïl, KELLER, Rudolf K. 1999. "Generating user interface prototypes from scenarios". *Proceedings. IEEE International Symposium Requirements Engineering*. IEEE Computer Society: Piscataway, NJ. 150 -158.

- EMAM, Khaled El, BIRK, Andreas. 2000. "Validating the ISO/IEC 15504 Measure of Software Requirements Analysis Process Capability". *IEEE Transactions On Software Engineering*. Juin. IEEE Computer Society: Minneapolis. 541-566.
- GAGNON, Yves-Chantal. 2005. *L'étude de cas comme méthode de recherche : guide de réalisation*. Presses de l'Université du Québec: Sainte-Foy, QC.
- GERY, Gloria. 1997. "Granting three wishes through performance-centered design". *Communications of the ACM*. Juillet. ACM Press: New York, NY. 0407.54-59.
- GIBBS, Wayt W. 1997. "Taking Computers to Task". *Scientific American*. Juillet. Scientific American: New York. 82-89.
- GOULD, J. D., BOIES, S. J., UKELSON, J. 1997. "How to Design Usable Systems". *Handbook of Human-computer Interaction, second edition*. Eds. Elsevier: Amsterdam. 231-254.
- GULLIKSEN, Jan, GÖRANSSON, Bengt, BOIVIE, Inger, PERSSON, Jenny, BLOMKVIST, Stefan, CAJANDER, Åsa. 2005. "Key Principles for User-Centred Systems Design". *Human-Centered Software Engineering*. Eds. Kluwer Academic Publishers: Boston, Ma. 17-36.
- GULLIKSEN, Jan, GÖRANSSON, Bengt, LIF, Magnus. 2001. "A User-Centered Approach to Object-Oriented User Interface Design". *Object modeling and user interface design*. Eds. Addison-Wesley: Boston. 288-312.
- HLADY RISPAL, Martine. 2002. *La méthode des cas : application à la recherche en gestion*. De Boeck Université: Paris.
- HOFMAN, Huber F., LEHNER, Franz, HOFMAN, Huber F. 2001. "Requirements engineering as a success factor in software projects". *IEEE Software*. Juillet/Août. IEEE Computer Society: Minneapolis. 184.58-66.
- HOMRIGHAUSEN, Andreas, SIX, Hans-Werner WINTER, Mario. 2002. "Round-Trip Prototyping Based on Integrated Functional and User Interface Requirements Specifications". *Requirements Engineering*. Springer: London, UK. 34-45.
- IEEE. 1998a. *IEEE guide for information technology - system definition - Concept of Operations (ConOps) document*. Mars. IEEE Computer Society: New York, NY. IEEE Std 1362-1998.

- IEEE. 1998b. *IEEE guide for developing system requirements specifications*. Décembre. IEEE Computer Society: New York, NY. IEEE Std 1233, 1998 Edition.
- IEEE. 1998c. *IEEE recommended practice for software requirements specifications*. Juin. IEEE Computer Society: New York, NY. IEEE Std 830-1998.
- IEEE. 2005. *Guide to the software engineering body of knowledge: 2004 version*. Février. Eds. IEEE Computer Society: Los Alamitos, CA. SWEBOK.
- ISO. 1998. *Exigences ergonomiques pour travail de bureau avec terminaux à écrans de visualisation (TEV) Partie 11: Lignes directrices relatives à l'utilisabilité*. Mars. ISO (Organisation internationale de normalisation): Genève, Suisse. ISO 9241-11:1998(F).
- ISO. 1999. *Processus de conception centrée sur l'opérateur humain pour les systèmes interactifs*. Août. ISO (The International Organization For Standardization): 29p. Norme internationale ISO Suisse ISO 13407:1999.
- ISO. 2000. *Ergonomics -- Ergonomics of human-system interaction -- Human-centred lifecycle process descriptions*. Avril. ISO (The International Organization for Standardization): Genève, Suisse. ISO/TR 18529:2000(E).
- ISO. 2004. *Principes ergonomiques de la conception des systèmes de travail*. Février. ISO (Organisation internationale de normalisation): Genève, Suisse. 6385:2004(F).
- ISO/IEC 1995. *Information Technology Software Life Cycle Processes*. Février. IEEE Computer Society: New York, NY. ISO/IEC 12207:1994(E).
- JACOBSON, Ivar, BOOCH, Grady, RUMBAUGH, James. 1999. *The unified software development process*. Addison-Wesley: Reading, MA. 463.
- JACOBSON, Ivar. 1994. "Basic Use Case Modeling". *Report on Object Analysis and Design*. Juillet/Août. SIGS Publications: New York. 15-19.
- JACOBSON, Ivar. 1995. "The Use-Case Construct in Object-Oriented Software Engineering". *Scenario-based design : envisioning work and technology in system development*. Eds. John Wiley and Sons: New York. 309-336.
- JOHN, Bonnie E., BASS, Len J., ADAMS, Robert J. 2003. "Communication across the HCI/SE divide: ISO 13407 and the Rational Unified Process®". *In proceedings of HCI International*,. Juin. Elsevier: Crête. 5.

- JOHNSON, J. 2000. "First Principles". *GUI Bloopers: Don'ts and Do's for Software Developers and Web Designers*. Academic Press: San Francisco. 1-37.
- KRUCHTEN, Philippe, AHLQVIST, Stefan, BYLUND, Stefan. 2001. "User Interface Design in the Rational Unified Process". *Object modeling and user interface design*. Eds. Addison-Wesley: Boston. 131-196.
- KRUCHTEN, Philippe. 2001. *Introduction au Rational Unified Process*. Éditions Eyrolles: Paris, France.
- LAMOUREUX, Andrée. 1992. Collaborateurs : ARCHAMBAULT, Jean-Pierre, BERTHIAUME, François, FRÉCHETTE, Nathalie, *Une démarche scientifique en sciences humaines : méthodologie*. Études vivantes: Laval, Qc.
- LANDAUER, T. K. 1995. *The trouble with computers : usefulness, usability, and productivity*. MIT Press: Cambridge, Mass.
- LEFFINGWELL, Dean, WIDRIG, Don. 2000. *Managing software requirements : a unified approach*. Addison-Wesley: Reading, MA.
- LEMEUNIER, Thierry. 2000. "Principe de la méthode". *L'intentionnalité communicative dans le dialogue homme-machine en langue naturelle*. Université du Maine: Le Mans, France.
- LÉVESQUE, Ghislain. 1998. *Analyse de système orientée objet et génie logiciel : concept, méthodes et applications*. Chenelière/McGraw-Hill: Montréal, QC.
- MAGUIRE, Martin. 2001. "Methods to support human-centred design". *International Journal of Human Computer Studies*. Octobre. Academic Press: London, UK. 55 4.587-634.
- MAYHEW, D. J. 1999. "Introduction". *The Usability engineering lifecycle*. Morgan Kaufman: San Francisco. 1-31.
- NAUR, Peter, RANDELL, Brian. OTAN. 1969. "Software Engineering: Report on a conference sponsored by the NATO Science Committee". *Software Engineering: Report on a conference sponsored by the NATO Science Committee*. Octobre. Eds. Organisation du traité de l'Atlantique nord: Garmisch, Germany Allemagne.
- PIMENTA, Marcelo Soares, FAUST, Richard. 1997. "Eliciting Interactive Systems Requirements in a Language-Centred User-Designer Collaboration: A Semiotic Approach". *ACM SIGCHI Bulletin*. Janvier. ACM Special Interest Group on Computer-Human Interaction: Minneapolis.

- PREECE, Jennifer, ROGERS, Yvonne, SHARP, Helen. 2002. *Interaction design : beyond human-computer interaction*. New York. : John Wiley and Sons. 519p.
- RATIONAL. 2002. *Rational Unified Process*. Version 2002.05.01.305.000. Rational Software Corporation: Cupertino, Ca.
- RATIONAL. 2003. *Rational Unified Process*. Version 2003.06.00.65. Rational Software Corporation: Cupertino, Ca.
- RAVIDA, A., BERRY, Daniel M. 2000. "A Method for Extracting and Stating Software Requirements that a User Interface Prototype Contains". *Requirements Engineering*. Springer: London, UK. 225-241.
- ROSSON, Mary Beth, CARROLL, John M.. 2002. *Usability Engineering : Scenario-based Development of Human-Computer Interaction*. San Francisco. : Morgan Kaufman. 422p.
- SAWYER, Peter, KOTONYA, Gerald, SAWYER, Peter. 2001. "Software requirements". *Guide to the software engineering body of knowledge : trial version SWEBOK : a project of the Software Engineering Coordinating Committee*. Mai. Eds. IEEE Computer Society: Los Alamitos, CA. 2-1 à 2-26.
- SEFFAH, Ahmed, DESMARAIS, Michel C., METZKER, Eduard. 2005. "Hci, Usability and Software Engineering Integration: Present and Future". *Human-Centered Software Engineering* . Eds. Kluwer Academic Publishers: Boston, Ma. 37-57.
- SOUSA, Kênia Soares, FURTADO, Elizabeth. 2003. "RUPi, A Unified Process that Integrates Human-Computer Interaction and Software Engineering". *Bridging the Gaps Between Software Engineering and Human-Computer Interaction*. Mai. International Federation for Information Processing: Portland, Or. 41-48.
- SOUSA, Kênia Soares, FURTADO, Elizabeth. 2003b. "A Unified Process for Interactive Systems". *CLIHC 2003 - Workshop Integrating Human-Computer Interaction and Software Engineering Models and Processes* . Août. Semiotic engineering research group: Rio de Janeiro, BR. 1-6.
- SOUSA, Kênia Soares, FURTADO, Elizabeth. 2003c. "An Approach To Integrate Hci And Se In Requirements Engineering". *Closing the Gap : Software Engineering and Human-Computer Interaction* . Septembre. International Federation for Information Processing: Louvain-la- Neuve, Belgique. 81-88.

- TAYLOR, Andrew. 2000. " IT projects: sink or swim ?". *The Computer Bulletin*. Janvier. The British Computer Society: Croydon, UK. 24-26.
- THAYER, Richard H. 2002. "Software system engineering: a tutorial". *Computer*. Avril. IEEE Computer Society: Piscataway, NJ. 68-73.
- THE STANDISH GROUP INTERNATIONAL, INC., .1995. The CHAOS Report. Décembre. The Standish Group International, Inc.: West Yarmouth, MA
- WASSERMAN, Anthony I. 1996. "Toward a discipline of software engineering". *IEEE Software*. Novembre. IEEE Computer Society: Piscataway, NJ. 23-31.
- YIN, Robert K. 1989. "Analyzing Case Study Evidence". *Case study research : design and methods*. Sage Publications: Newbury Park, CA.
- YIN, Robert K. 2003. *Applications of case study research 2nd ed*. Sage Publications: Thousand Oaks, CA.
- YOUNG, Ralph Rowland. 2001. "Define the Real Customer Needs". *Effective requirements practices*. Addison-Wesley: Boston, Ma. 57-95.
- ZELKOWITZ, Marvin V., WALLACE, Dolores R.. 1998. "Experimental Models for Validating Technology". *Computer*. Mai. IEEE Computer Society: Piscataway, NJ. 031:5.23-31.

ANNEXE I

SOURCES DE DONNÉES TRIÉES PAR ORDRE ALPHABÉTIQUE

- Analyse de contexte d'utilisation, 2003-04-10, pp. 37.
- Appel de propositions, 2003-01-31, pp. 27.
- Architecture de référence, 2002-11-28, pp. 145.
- Bulletin du projet vol.1 n° 1, 2003-04-11, pp. 2.
- Bulletin du projet vol.1 n° 2, 2003-04-17, pp. 2.
- Bulletin du projet vol.1 n° 3, 2003-04-24, pp. 2.
- Bulletin du projet vol.1 n° 4, 2003-04-30, pp. 2.
- Bulletin du projet vol.1 n° 5, 2003-05-07, pp. 2.
- Bulletin du projet vol.1 n° 6, 2003-05-21, pp. 2.
- Bulletin du projet vol.1 n° 7, 2003-05-29, pp. 2.
- Bulletin du projet vol.1 n° 8, 2003-06-05, pp. 2.
- Bulletin du projet vol.1 n° 9, 2003-06-12, pp. 2.
- Cadre de conception des interfaces WEB, 2003-01-14, pp. 54.
- Cas d'utilisation - Gérer un dossier détaillant, 2003-04-14, pp. 72.
- Cas d'utilisation - Gérer un dossier détaillant, 2003-10-31, pp. 72.
- Cas d'utilisation - Consulter un dossier détaillant, 2003-10-31, pp. 23.
- Cas d'utilisation - Définir une adresse, 2003-09-04, pp. 15.
- Cas d'utilisation - Gérer le dossier équipement, 2003-02-18, pp. 48.
- Cas d'utilisation - Gérer le dossier équipement, 2003-08-29, pp. 43.
- Cas d'utilisation - Gérer les responsables d'un dossier détaillant, 2003-01-14, pp. 23.
- Cas d'utilisation - Gérer les responsables d'un dossier détaillant, 2003-10-31, pp. 33.
- Cas d'utilisation - Gérer un dossier détaillant, 2003-02-26, pp. 31.

Cas d'utilisation - Gérer une personne, 2003-11-18, pp. 24.

Cas d'utilisation - Identifier un détaillant, 2003-06-29, pp. 9.

Compte rendu d'entrevue avec analyste d'affaires et représentant d'utilisateur, 2005-10-15, pp. 5.

Compte rendu d'entrevue avec analyste fonctionnel, 2005-11-18, pp. 3.

Compte rendu d'entrevue avec chef d'équipe de gestion du changement, 2005-11-03, pp. 1.

Compte rendu du pré test d'utilisabilité Cycle 2 Itération 1, 2003-04-22, pp. 2.

Compte rendu du test d'utilisabilité Cycle 2 Itération 1, 2003-05-23, pp. 2.

Diagramme de contexte - Réseau de distribution, 2002-12-03, pp. 1.

Dictionnaire des termes d'affaires, 2003-05-28, pp. 43.

Distinctions WIMP - Web, 2003-03-23, pp. 10.

DPO - Annulation d'un numéro de détaillant, 2003-11-26, pp. 1.

DPO - Consultation du dossier détaillant, 2003-11-26, pp. 1.

DPO - Émission d'un numéro de détaillant, 2003-11-26, pp. 3.

DPO - Modification du dossier détaillant, 2003-11-26, pp. 1.

Étude des questions techniques et ergonomiques, 2003-05-27, pp. 24.

Façon de faire - Vue globale, 2003-03-25, pp. 25.

Fiche signalétique - Détail de cas d'utilisation, 2003-02-10, pp. 10.

Fiche signalétique - Modèle de cas d'utilisation, 2003-02-10, pp. 11.

Fiche signalétique - Modèle des concepts-clés, 2003-04-01, pp. 13.

Fiche signalétique - Registre des caractéristiques logicielles fonctionnelles, 2003-02-10, pp. 9.

Fiche signalétique - Registre des caractéristiques logicielles non fonctionnelles, 2003-02-10, pp. 8.

Fiche signalétique - Scénarimage, 2003-04-03, pp. 16.

Formation - Scénarimage et ergonomie, 2003-06-11, pp. 18.

Gabarit de cas d'utilisation, 2003-11-17, pp. 8.

Gabarit de scénarimage, 2003-05-20, pp. 5.

Guide de détail de cas d'utilisation, 2003-07-06, pp. 29.

Guide de notation des cas d'utilisation, 2003-06-02, pp. 25.

Guide de scénarimage, 2003-01-27, pp. 9.

Guide de style des interfaces Web, 2003-11-19, pp. 9.

Intervention en ergonomie - Bilan du premier cycle, 2003-03-17, pp. 17.

Liste des acteurs - Réseau de distribution, 2003-02-11, pp. 37.

Message annonçant la présentation de l'itération 0, 2003-04-30, pp. 1.

Modèle des cas d'utilisation - Réseau de distribution - Version 2, 2002-12-20, pp. 30.

Modèle des cas d'utilisation - Réseau de distribution - Version 4, 2003-04-22, pp. 31.

Modèle des cas d'utilisation - Réseau de distribution - Version 5, 2003-05-30, pp. 31.

Modèle des concepts clés, 2002-12-10, pp. 26.

Planification des activités ergonomiques 2e cycle, 2003-06-17, pp. 9.

Planification des activités ergonomiques des quatre premières semaines, 2003-02-16, pp. 4.

Proposition de modification de scénarimage, 2003-04-22, pp. 8.

Rapport annuel 2002, 2002-06-17, pp. 70.

Rapport annuel 2003, 2003-06-17, pp. 64.

Rapport annuel 2004, 2004-06-10, pp. 66.

Rapport annuel 2005, 2005-06-16, pp. 74.

Rapport hebdomadaire de l'ergonomie des interfaces, pp. 19.

Registre des caractéristiques logicielles - Réseau de distribution, 2003-03-21, pp. 10.

Registre des caractéristiques logicielles non fonctionnelles, 2003-02-25, pp. 11.

Rencontre avec analystes fonctionnels sur bilan de cycle ergonomique, 2003-04-09, pp. 2.

Rôles- profils et responsabilités, 2003-02-10, pp. 5.

Scénarimage - Consulter un dossier détaillant, 2003-10-31, pp. 30.

Scénarimage - Définir une adresse, 2003-09-04, pp. 24.

Scénarimage - Gérer les équipements, 2003-08-29, pp. 36.

Scénarimage - Gérer une personne, 2003-11-18, pp. 28.

Scénarimage - Identifier un détaillant, 2003-06-30, pp. 13.

Scénarimage - Gérer un dossier détaillant, 2003-11-03, pp. 70.

Scénarimage - Gérer les responsables d'un dossier détaillant, 2003-10-31, pp. 55.

Spécification d'interface - Dossier détaillant, 2003-06-02, pp. 103.

Spécification d'interface - Zone de navigation primaire, 2003-06-11, pp. 7.

Traçabilité Gestion des réseaux, 2002-12-06, pp. 7.

ANNEXE II

SOURCES DE DONNÉES GROUPEES PAR TYPE DE DOCUMENT

Les sources de données sont regroupées par type de document et triées par date de parution à l'intérieur de chaque groupe.

Biens livrables

- Architecture de référence, 2002-11-28, pp. 145.
- Diagramme de contexte - Réseau de distribution, 2002-12-03, pp. 1.
- Modèle des concepts clés, 2002-12-10, pp. 26.
- Modèle des cas d'utilisation - Réseau de distribution - Version 2, 2002-12-20, pp. 30.
- Cas d'utilisation - Gérer les responsables d'un dossier détaillant, 2003-01-14, pp. 23.
- Cadre de conception des interfaces WEB, 2003-01-14, pp. 54.
- Liste des acteurs - Réseau de distribution, 2003-02-11, pp. 37.
- Cas d'utilisation - Gérer le dossier équipement, 2003-02-18, pp. 48.
- Registre des caractéristiques logicielles non fonctionnelles, 2003-02-25, pp. 11.
- Cas d'utilisation - Gérer un dossier détaillant, 2003-02-26, pp. 31.
- Registre des caractéristiques logicielles - Réseau de distribution, 2003-03-21, pp. 10.
- Distinctions WIMP - Web, 2003-03-23, pp. 10.
- Analyse de contexte d'utilisation, 2003-04-10, pp. 37.
- Cas d'utilisation - Gérer un dossier détaillant, 2003-04-14, pp. 72.
- Modèle des cas d'utilisation - Réseau de distribution - Version 4, 2003-04-22, pp. 31.
- Étude des questions techniques et ergonomiques, 2003-05-27, pp. 24.
- Dictionnaire des termes d'affaires, 2003-05-28, pp. 43.
- Modèle des cas d'utilisation - Réseau de distribution - Version 5, 2003-05-30, pp. 31.
- Spécification d'interface - Dossier détaillant, 2003-06-02, pp. 103.

- Spécification d'interface - Zone de navigation primaire, 2003-06-11, pp. 7.
- Cas d'utilisation - Identifier un détaillant, 2003-06-29, pp. 9.
- Scénarimage - Identifier un détaillant, 2003-06-30, pp. 13.
- Scénarimage - Gérer les équipements, 2003-08-29, pp. 36.
- Cas d'utilisation - Gérer le dossier équipement, 2003-08-29, pp. 43.
- Cas d'utilisation - Définir une adresse, 2003-09-04, pp. 15.
- Scénarimage - Définir une adresse, 2003-09-04, pp. 24.
- Cas d'utilisation - Gérer un dossier détaillant, 2003-10-31, pp. 72.
- Cas d'utilisation - Gérer les responsables d'un dossier détaillant, 2003-10-31, pp. 33.
- Cas d'utilisation - Consulter un dossier détaillant, 2003-10-31, pp. 23.
- Scénarimage - Gérer les responsables d'un dossier détaillant, 2003-10-31, pp. 55.
- Scénarimage - Consulter un dossier détaillant, 2003-10-31, pp. 30.
- Scénarimage - Gérer un dossier détaillant, 2003-11-03, pp. 70.
- Cas d'utilisation - Gérer une personne, 2003-11-18, pp. 24.
- Scénarimage - Gérer une personne, 2003-11-18, pp. 28.
- DPO - Consultation du dossier détaillant, 2003-11-26, pp. 1.
- DPO - Annulation d'un numéro de détaillant, 2003-11-26, pp. 1.
- DPO - Émission d'un numéro de détaillant, 2003-11-26, pp. 3.
- DPO - Modification du dossier détaillant, 2003-11-26, pp. 1.

Bulletins internes du projet

- Bulletin du projet vol.1 n° 1, 2003-04-11, pp. 2.
- Bulletin du projet vol.1 n° 2, 2003-04-17, pp. 2.
- Bulletin du projet vol.1 n° 3, 2003-04-24, pp. 2.
- Bulletin du projet vol.1 n° 4, 2003-04-30, pp. 2.

Bulletin du projet vol.1 n° 5, 2003-05-07, pp. 2.

Bulletin du projet vol.1 n° 6, 2003-05-21, pp. 2.

Bulletin du projet vol.1 n° 7, 2003-05-29, pp. 2.

Bulletin du projet vol.1 n° 8, 2003-06-05, pp. 2.

Bulletin du projet vol.1 n° 9, 2003-06-12, pp. 2.

Comptes rendus

Rencontre avec analystes fonctionnels sur bilan de cycle ergonomique, 2003-04-09, pp. 2.

Compte rendu du pré test d'utilisabilité Cycle 2 Itération 1, 2003-04-22, pp. 2.

Compte rendu du test d'utilisabilité Cycle 2 Itération 1, 2003-05-23, pp. 2.

Compte rendu d'entrevue avec analyste d'affaires et représentant d'utilisateur, 2005-10-15, pp. 5.

Compte rendu d'entrevue avec chef d'équipe de gestion du changement, 2005-11-03, pp. 1.

Compte rendu d'entrevue avec analyste fonctionnel, 2005-11-18, pp. 3.

Courriel

Message annonçant la présentation de l'itération 0, 2003-04-30, pp. 1.

Documents de gestion

Appel de propositions, 2003-01-31, pp. 27.

Gabarits de document

Gabarit de scénarimage, 2003-05-20, pp. 5.

Gabarit de cas d'utilisation, 2003-11-17, pp. 8.

Guides

Guide de scénarimage, 2003-01-27, pp. 9.

Guide de notation des cas d'utilisation, 2003-06-02, pp. 25.

Guide de détail de cas d'utilisation, 2003-07-06, pp. 29.

Guide de style des interfaces Web, 2003-11-19, pp. 9.

Lignes directrices

Fiche signalétique - Modèle de cas d'utilisation, 2003-02-10, pp. 11.

Rôles- profils et responsabilités, 2003-02-10, pp. 5.

Fiche signalétique - Registre des caractéristiques logicielles non fonctionnelles, 2003-02-10, pp. 8.

Fiche signalétique - Registre des caractéristiques logicielles fonctionnelles, 2003-02-10, pp. 9.

Fiche signalétique - Détail de cas d'utilisation, 2003-02-10, pp. 10.

Façon de faire - Vue globale, 2003-03-25, pp. 25.

Fiche signalétique - Modèle des concepts-clés, 2003-04-01, pp. 13.

Fiche signalétique - Scénarimage, 2003-04-03, pp. 16.

Planifications

Planification des activités ergonomiques des quatre premières semaines, 2003-02-16, pp. 4.

Planification des activités ergonomiques 2e cycle, 2003-06-17, pp. 9.

Présentations

Intervention en ergonomie - Bilan du premier cycle, 2003-03-17, pp. 17.

Formation - Scénarimage et ergonomie, 2003-06-11, pp. 18.

Rapports

Traçabilité Gestion des réseaux, 2002-12-06, pp. 7.

Rapports annuels de l'entreprise

Rapport annuel 2002, 2002-06-17, pp. 70.

Rapport annuel 2003, 2003-06-17, pp. 64.

Rapport annuel 2004, 2004-06-10, pp. 66.

Rapport annuel 2005, 2005-06-16, pp. 74.

Rapports hebdomadaires

Rapports hebdomadaires de l'ergonomie des interfaces, 2003, pp. 19.

Recommandations

Proposition de modification de scénarimage, 2003-04-22, pp. 8.