

# **Pratique recommandée par IEEE pour la préparation de spécifications d'exigences de logiciel**

*Circuits et dispositifs*

*Technologie des communications*

**Société d'informatique IEEE**

**Commandité par  
le Comité des normes de l'ingénierie de logiciels**

*Électromagnétique et rayonnements*

*Énergie et puissance*

*Applications industrielles*

*Signaux et applications*

*Comités de coordination des normes*

# **Pratique recommandée par IEEE pour la préparation de spécifications d'exigences de logiciel**

Commanditaire :  
**le Comité des normes de l'ingénierie de logiciels  
de la Société d'informatique IEEE**

Approuvé le 2 décembre 1993  
**Conseil des normes IEEE**

**Résumé :** Le présent document décrit ce que devrait contenir toute spécification d'exigences de logiciel (SEL) et quelles sont les caractéristiques d'une SEL bien rédigée. Il présente également le plan de plusieurs exemples de SEL. Cette pratique recommandée vise en premier lieu à spécifier les exigences relatives aux logiciels, mais elle peut aussi aider à choisir des logiciels internes ou commerciaux.

---

The Institute of Electrical and Electronics Engineers, Inc.  
345 East 47<sup>th</sup> Street, New York, NY 10017-2394, USA

Copyright © Institute of Electrical and Electronics Engineers, Inc.  
Tous droits réservés. Publié en 1994. Imprimé aux États-Unis.

ISBN 1-55937-395-4

*Toute reproduction d'un extrait quelconque de cette publication par quelque procédé que ce soit, et notamment par un système d'extraction électronique, est strictement interdite sans autorisation écrite de l'éditeur.*

Les normes IEEE sont élaborées par les comités techniques des sociétés IEEE et par les comités de coordination des normes du IEEE Standards Board. Bénévoles, les membres des comités ne sont pas rémunérés. Ils doivent obligatoirement être membres de l'Institut. Toute norme élaborée au IEEE représente un consensus des vastes compétences sur le sujet réunies au sein de l'Institut, ainsi que des activités extérieures au IEEE qui ont souhaité participer à l'élaboration de la norme.

L'utilisation des normes IEEE est volontaire. L'existence d'une norme IEEE n'implique pas qu'il n'existe aucune autre façon de produire, de tester, de mesurer, d'acheter, de mettre en marché ou de fournir les biens et services qu'englobe la norme IEEE. De plus, le point de vue exprimé au moment de l'approbation et de la publication d'une norme est susceptible de changer à la suite de l'avancement des connaissances ainsi que des commentaires reçus des utilisateurs de la norme. Chaque norme IEEE est révisée ou reconduite au moins tous les cinq ans. Lorsqu'un document date de plus de cinq ans et n'a pas été reconduit, il est raisonnable de conclure que son contenu, même s'il possède encore une certaine valeur, ne reflète pas les connaissances les plus récentes. Les utilisateurs devraient vérifier s'ils ont en main la version la plus récente de la norme.

Aux fins de la révision de ses normes, l'Institut accepte les commentaires de toute partie intéressée, quelle que soit son affiliation au IEEE. Les suggestions de modification à un document devrait prendre la forme d'une proposition de texte modifié, accompagnée des commentaires pertinents.

Interprétations : À l'occasion, le sens de certaines parties d'une norme peut prêter à interprétation, par rapport à une application spécifique. Lorsqu'une demande d'interprétation est formulée au IEEE, celui-ci lance le processus de préparation de la réponse. Comme les normes IEEE sont le résultat d'un consensus entre toutes les parties intéressées, il faut s'assurer que toute interprétation reçoit leur consentement, en fonction d'un équilibre des intérêts. C'est pourquoi l'Institut et les membres de ses comités techniques ne sont pas en mesure de répondre immédiatement aux demandes d'interprétation, sauf dans les cas où la question a déjà fait l'objet d'un examen formel.

Envoyer les commentaires sur les normes et les demandes d'interprétation à l'adresse suivante :

Secretary, IEEE Standards Board  
443 Hoew Lane  
P.O. Box 1331  
Piscataway, NJ 08855-1331  
USA

Les normes IEEE peuvent entraîner l'utilisation de technologies brevetées. Leur approbation par l'Institute of Electrical and Electronics Engineers ne signifie pas que l'utilisation d'une technologie en vue de se conformer à une norme est autorisée par le détenteur du brevet. Il incombe à l'utilisateur d'obtenir toutes les permissions nécessaires à son utilisation.

## Table des matières

ARTICLE	PAGE
1. Vue d'ensemble .....	1
1.1 Portée .....	1
2. Références .....	2
3. Définitions .....	3
4. Facteurs à considérer dans la préparation d'une SEL.....	4
4.1 Nature de la SEL.....	4
4.2 Contexte de la SEL .....	4
4.3 Caractéristiques d'une SEL bien rédigée .....	5
4.4 Préparation conjointe de la SEL .....	9
4.5 Évolution de la SEL.....	9
4.6 Réalisations de prototypes .....	9
4.7 Incorporation des critères de conception .....	10
4.8 Incorporation des exigences du projet .....	10
5. Les parties de la SEL .....	11
5.1 Introduction (chapitre 1 de la SEL) .....	11
5.2 Description générale (chapitre 2 de la SEL) .....	12
5.3 Exigences spécifiques (chapitre 3 de la SEL).....	15
5.4 Renseignements d'appui .....	20
Annexe A .....	21
A.1 Modèle du chapitre 3 organisé par mode (version 1) .....	21
A.2 Modèle du chapitre 3 organisé par mode (version 2) .....	21
A.3 Modèle du chapitre 3 organisé par classe d'utilisateur.....	22
A.4 Modèle du chapitre 3 organisé par objet.....	22
A.5 Modèle du chapitre 3 organisé par caractéristique .....	23
A.6 Modèle du chapitre 3 organisé par stimulus .....	24
A.7 Modèle du chapitre 3 organisé par hiérarchie de fonctions.....	24
A.8 Modèle du chapitre 3 montrant des organisations multiples .....	26

## Introduction

(Cette introduction ne fait pas partie de la norme IEEE 830-1993 – Pratique recommandée par IEEE pour la préparation de spécifications d'exigences de logiciel.)

Le présent document décrit les approches recommandées dans la préparation de spécifications d'exigences de logiciel. La pratique se fonde sur un processus dont le résultat est un document de spécification non ambigu et complet. Elle devrait aider :

- a) les acheteurs de logiciels à décrire précisément ce qu'ils recherchent;
- b) les fournisseurs de logiciels à comprendre exactement ce que veut le client;
- c) les personnes qui veulent atteindre les objectifs suivants :
  - 1) élaborer un plan normalisé de spécification d'exigences de logiciel (SEL) à l'intention de leur entreprise
  - 2) déterminer le format et le contenu de leurs spécifications d'exigences de logiciel particulières
- 3) préparer d'autres outils d'aide, comme une liste de contrôle ou un guide de rédaction.

Une SEL bien rédigée devrait apporter plusieurs avantages à ces personnes, notamment :

- a) *Établir les fondements de l'entente à conclure entre le client et le fournisseur, sur les fonctions du logiciel.* La description complète des fonctions que doit réaliser le logiciel aide l'utilisateur soit à déterminer si le logiciel satisfait ses besoins, soit à préciser comment le modifier.
- b) *Réduire les tâches de développement.* La préparation d'une SEL force les divers groupes intéressés dans l'entreprise à prendre rigoureusement en compte tous les besoins avant d'entreprendre la conception; elle réduit ainsi la nécessité de rajuster la conception, de reprendre le codage et de faire de nouveaux essais. En revoyant avec soin les exigences de la SEL, on peut détecter les omissions, les ambiguïtés et les incohérences, très tôt dans le cycle de développement, lorsque ces problèmes sont plus faciles à corriger.
- c) *Fournir une base pour l'estimation des coûts et du calendrier.* La description du produit à développer, tel qu'elle est fournie par la SEL, constitue un fondement réaliste pour l'estimation des coûts du projet et peut être utilisée pour faire approuver les offres ou les estimations de prix.
- d) *Fournir une mesure de base pour la validation et la vérification.* L'entreprise peut élaborer un plan de validation et de vérification beaucoup plus efficace si elle se fonde sur la SEL. Si elle est intégrée au contrat de développement, la SEL offre une base qui permet de mesurer la conformité du résultat.
- e) *Faciliter le transfert aux nouveaux utilisateurs ou aux nouvelles machines.* Ainsi, il est plus facile pour le client de transférer le logiciel ailleurs dans l'entreprise et pour le fournisseur de le transférer à de nouveaux clients.
- f) *Servir de base pour une mise à niveau ou une amélioration.* Puisque la SEL traite du produit et non du projet, la SEL peut servir de base à toute mise à niveau du produit fini. Elle peut devoir être modifiée, mais elle offre un fondement pour l'évaluation continue de la production.

Au moment de l'achèvement de cette pratique, le groupe de travail sur les spécifications d'exigences de logiciel était formé des membres suivants :

**Edward R. Byrne**, président

Fletcher J. Buckley  
A. M. Davis  
J.W. Horch  
Motti Klein

Thomas M. Kurihara  
John R. Matras  
Robert Nation  
William C. Sasso  
Carl S. Seddio

Carl A. Singer  
Richard H. Thayer  
Leonard L. Tripp  
A. Yonda

# Pratique recommandée par IEEE pour la préparation de spécifications d'exigences de logiciel

## 1. VUE D'ENSEMBLE

Le présent document décrit les approches recommandées dans la préparation de spécifications d'exigences de logiciel. Il comporte cinq chapitres. Le premier explique la portée du document. Le deuxième donne le nom des autres normes auquel renvoie le texte. Le troisième définit les termes utilisés. Le quatrième donne des conseils de nature générale sur la rédaction d'une SEL. Le cinquième aborde les diverses parties de la SEL. On trouvera également, en annexe, des modèles de divers formats.

### 1.1 Portée

Le présent document recommande une pratique relative à la rédaction de spécifications d'exigences de logiciel (SEL). Elle décrit le contenu et les qualités d'une bonne SEL et donne plusieurs exemples de plans de SEL.

La pratique recommandée vise à spécifier les exigences de logiciels à développer, mais elle peut également s'appliquer à la sélection de logiciels internes ou commerciaux. Toutefois, en l'appliquant à des logiciels existants, on pourrait aller à l'encontre des buts recherchés.

Si le logiciel est intégré à un système plus vaste, comme de l'équipement médical, il faut parfois aborder des questions qui dépassent celles décrites dans cette norme.

Cette pratique recommandée décrit le processus de création d'un produit et le contenu de celui-ci. Le produit consiste dans la spécification des exigences de logiciel. Elle peut être employée en tant que telle pour créer une spécification ou elle peut servir de modèle pour la conception d'une norme plus spécifique.

Cette pratique recommandée ne précise pas de méthodes, de nomenclatures ou d'outils spécifiques pour la préparation d'une SEL.

## 2. RÉFÉRENCES

- ASTM 1340-90, Standard Guide for Rapide Prototyping of Computerized Systems.<sup>1</sup>
- Norme IEEE 610.12-1990, IEEE Standard Glossary of Software Engineering Terminology (ANSI)<sup>2</sup>
- Norme IEEE 730-1989, IEEE Standard for Software Quality Assurance Plans (ANSI)
- Norme IEEE 828-1990, IEEE Standard for Software Configuration Management Plans (ANSI)
- Norme IEEE 982.1-1988, IEEE Standard Dictionary of Measures to Produce Reliable Software (ANSI)
- Norme IEEE 982.2-1988, IEEE Guide for the Use of IEEE Standard Dictionary of Measures to Produce Reliable Software (ANSI)
- Norme IEEE 983-1986, IEEE Guide to Software Quality Assurance Planning<sup>3</sup>
- Norme IEEE 1002-1987, IEEE Standard Taxonomy for Software Engineering Standards (ANSI)
- Norme IEEE 1012-1986, IEEE Standard for Software Verification and Validation Plans (ANSI)
- Norme IEEE 1016-1987, IEEE Recommended Practice for Software Design Descriptions (ANSI)
- Norme IEEE 1028-1988, IEEE Standard for Software Reviews and Audits (ANSI)
- Norme IEEE 1042-1987, IEEE Guide to Software Configuration Management (ANSI)
- Norme IEEE 1058.1-1987, IEEE Standard for Project Management Plans (ANSI)
- Norme IEEE 1074-1991, IEEE Standard for Developing Software Life Cycle Processes (ANSI)
- IEEE P1233, octobre 1993, Draft Guide to Developing Systems Requirements Specifications<sup>4</sup>

---

<sup>1</sup> On peut obtenir les publications de l'ASTM à l'adresse suivante : Customer Service Department, American Society for Testing and Materials, 1916 Race Street, Philadelphia, PA 19103, États-Unis

<sup>2</sup> On peut obtenir les publications de l'IEEE à l'adresse suivantes : Institute of Electrical and Electronics Engineers, 445 Hoes Lanes, P.O. Box 1331, Piscataway, NJ 08844-1331, États-Unis

<sup>3</sup> Cette norme a été retirée; cependant, on peut en obtenir un exemplaire à l'adresse suivante : IEEE Standards Department, 445 Hoes Lanes, P.O. Box 1331, Piscataway, NJ 08844-1331, États-Unis

<sup>4</sup> Ce projet autorisé de rédaction de norme n'avait pas encore été approuvé par le comité des normes de l'IEEE au moment de mettre sous presse; on peut toutefois en obtenir un exemplaire à l'adresse suivante : IEEE Standards Department, 445 Hoes Lanes, P.O. Box 1331, Piscataway, NJ 08844-1331, États-Unis

### 3. DÉFINITIONS

En général, les termes utilisés dans ce document sont conformes aux définitions de la norme IEEE 610.12-1990<sup>5</sup>. Les termes clés sont définis ci-dessous.

- 3.1 **contrat** : un document juridique exécutoire, sur lequel s'entendent le client et le fournisseur, et qui comprend les exigences techniques et organisationnelles, les coûts et le calendrier de réalisation d'un produit. Le contrat peut également contenir des renseignements informels mais utiles, tels que les attentes ou les engagements des parties.
- 3.2 **client** : personne ou groupe qui paie le produit et qui (habituellement) en précise les exigences. Dans le contexte de ce document, le client et le fournisseur peuvent faire partie de la même entreprise.
- 3.3 **fournisseur** : personne ou groupe qui fournit le produit au client. Dans le contexte de ce document, le client et le fournisseur peuvent faire partie de la même entreprise.
- 3.4 **utilisateur** : personne qui utilise le produit ou qui est en interaction avec lui.

---

<sup>5</sup> Voir les références à l'article 2.



## 4. FACTEURS À CONSIDÉRER DANS LA PRÉPARATION D'UNE SEL

Ce chapitre donne de l'information générale sur les facteurs à considérer dans la rédaction d'une SEL :

- a) Nature de la SEL
- b) Contexte de la SEL
- c) Caractéristiques d'une SEL bien rédigée
- d) Préparation conjointe de la SEL
- e) Évolution de la SEL
- f) Réalisation de prototypes
- g) Incorporation des critères de conception
- h) Incorporation des exigences du projet

### 4.1 Nature de la SEL

La SEL sert à spécifier les exigences d'un logiciel, d'un programme ou d'un progiciel en particulier, qui exécute certaines fonctions dans un environnement précis. Elle peut être rédigée par des représentants du fournisseur, du client, ou des deux à la fois. À la section 4.4, c'est cette dernière option qui est recommandée.

Les questions fondamentales que doivent aborder les rédacteurs de la SEL sont :

- a) *Les fonctions* : que doit faire le logiciel?
- b) *Les interfaces externes* : quelle types de liens doit-il y avoir entre le logiciel et les utilisateurs, le matériel du système, les autres matériels et les autres logiciels?
- c) *Performance* : quelle doit être la vitesse, le degré de disponibilité, le délai de réponse et le délai de récupération des diverses fonctions logicielles, etc.?
- d) *Attributs* : de quoi faut-il tenir compte sur le plan de la transférabilité, de la facilité d'exécution de la maintenance, de sécurité, etc.?
- e) *Contraintes imposées sur l'implantation* : y a-t-il des contraintes dont il faut tenir compte (normes, langages d'implantation, politiques visant l'intégrité des bases de données, ressources limitées, cadre d'exploitation, etc.)?

Les rédacteurs de SEL devraient éviter d'y inclure des exigences relatives à la conception ou au projet.

Les contenus recommandés sont précisés au chapitre 5.

### 4.2 Contexte de la SEL

Il est important d'envisager le rôle que joue la SEL dans l'ensemble du plan de projet (défini à la norme 610.12-1990). Le logiciel peut englober toutes les fonctions prévues au projet, ou il peut faire partie d'un système plus vaste. Dans ce dernier cas, on prépare habituellement une SEL qui décrit les interfaces entre le système et le logiciel et qui impose des exigences externes à ce dernier, sur le plan de la performance et des fonctions.

La norme IEEE 1974-1991 décrit les étapes du cycle de vie d'un logiciel et les intrants à fournir à chaque étape. D'autres normes (voir chapitre 2) abordent d'autres éléments du cycle de vie et pourraient donc compléter les exigences de logiciel.

Comme la SEL a un rôle précis à jouer dans le processus de développement des logiciels, les rédacteurs ne devraient pas dépasser les limites de ce rôle. Cela signifie que la SEL :

- a) Devrait décrire correctement toutes les exigences relatives au logiciel. Une exigence pourrait être nécessaire en raison de la nature de la tâche à exécuter ou d'une caractéristique particulière du projet.

- b) Ne devrait pas fournir de détails sur la conception ou l'implantation, qui doivent plutôt être fournis à l'étape de conception du projet.
- c) Ne devrait pas imposer de contraintes supplémentaires au logiciel. Celles-ci doivent plutôt être précisées dans des documents comme le plan d'assurance-qualité du logiciel.

Ainsi, une SEL bien rédigée restreint les conceptions acceptables, mais n'impose pas une conception en particulier.

### 4.3 Caractéristiques d'une SEL bien rédigée

Une SEL devrait être

- a) Exacte
- b) Non ambiguë
- c) Complète
- d) Cohérente
- e) Hiérarchisée en fonction de l'importance et/ou de la stabilité
- f) Vérifiable
- g) Modifiable
- h) Traçable

#### 4.3.1 Exacte

Une SEL est exacte si et seulement si chaque exigence qui y est formulée doit être respectée par le logiciel.

Il n'y a aucun outil ni procédure qui permet d'assurer l'exactitude. On peut confronter la SEL aux normes applicables de rang supérieur, telle une spécification d'exigences de système, aux autres documents de projet et à d'autres normes applicables, pour s'assurer de leur concordance. Par ailleurs, le client ou l'utilisateur peut vérifier si la SEL reflète les besoins réels. La traçabilité facilite cette procédure et la rend moins sujette à erreur (voir 4.3.8).

#### 4.3.2 Non ambiguë

Une SEL est non ambiguë si et seulement si chaque exigence qui y est formulée n'a qu'une interprétation possible. Au minimum, chaque caractéristique du produit final doit donc être décrite au moyen d'une expression unique. Si, dans un contexte particulier, cette expression pouvait avoir diverses significations, elle devrait être définie plus précisément dans un glossaire.

La SEL est une partie importante du processus de définition des exigences, qui fait partie du cycle de vie d'un logiciel. Elle sert à diverses étapes : conception, implantation, surveillance du projet, vérification/validation, formation, comme le précise la norme IEEE 1074-1991. Elle devrait être non ambiguë tant pour ses créateurs que pour ses utilisateurs. Cependant, ces groupes n'ont pas souvent le même historique, et n'ont donc pas tendance à décrire de la même façon les exigences relatives aux logiciels. Une formulation qui améliore la spécification des exigences pour les développeurs pourrait aller à l'encontre du but recherché, si elle a pour conséquence de diminuer la compréhension des utilisateurs (et vice versa).

Les paragraphes suivants font des recommandations destinées à lever les ambiguïtés.

##### 4.3.2.1 Pièges qui touchent la langue

Les exigences sont souvent écrites dans une langue naturelle (le français ou l'anglais). Toute langue est fondamentalement ambiguë. Une SEL en langue naturelle devrait être examinée par un tiers, qui signalera tout terme ambigu.

#### 4.3.2.2 Langages conçus pour la spécification des exigences

Une façon d'éviter les ambiguïtés inhérentes à la langue naturelle est de rédiger la SEL dans une langue conçue à cet effet. Ses processeurs de langage repèrent automatiquement les erreurs terminologiques, syntaxiques ou sémantiques.

L'utilisation de ces langages comporte des inconvénients : il faut beaucoup de temps pour les apprendre et ils sont incompréhensibles pour de nombreux utilisateurs. De plus, comme elles sont plus efficaces pour exprimer certains types d'exigences, dans le contexte de certains systèmes, elles risquent d'avoir une influence subtile sur la formulation des exigences.

#### 4.3.2.3 Outils de représentation

En général, les méthodes et langages utilisés pour formuler les exigences, et les outils qui les soutiennent, se divisent en trois catégories, selon qu'ils se fondent sur les objets, les processus ou les comportements. Les approches fondées sur les objets organisent les exigences en fonction des objets, de leurs attributs et des services qu'ils fournissent. Les approches fondées sur les processus organisent les exigences en hiérarchies de fonctions, qui communiquent entre elles par des organigrammes. Les approches comportementales décrivent le comportement extérieur du système en fonction d'une notion abstraite (comme le calcul à prédicats), de fonctions mathématiques ou de machines à états.

L'utilité de ces outils et de ces méthodes dans la rédaction d'une SEL dépend de la grosseur et de la complexité du programme. Le présent document ne vise ni à décrire ni à approuver aucun outil en particulier.

Si on utilise l'une ou l'autre de ces approches, il vaut mieux opter pour la description en langue naturelle, pour permettre aux clients qui ne connaissent pas les notations de comprendre la SEL.

### 4.3.3 Complète

Une SEL est complète si et seulement si elle contient les éléments suivants :

- a) *Toutes les exigences significatives, qu'elles se rapportent aux fonctions, à la performance, aux contraintes de conception, aux attributs ou aux interfaces externes. Il faut traiter en particulier de toute exigence externe imposée par des spécifications de système.*
- b) *La description des réponses du logiciel à tous les types possibles de données d'entrée dans toutes les situations réalisables. À noter qu'il est important de préciser les réponses aux données invalides aussi bien qu'aux données valides.*
- c) *Des titres complets et des références exhaustives pour la totalité des figures, tableaux et diagrammes de la SEL et une définition de tous les termes et unités de mesure.*

#### 4.3.3.1 Utilisation de l'expression « à déterminer »

Toute SEL qui comporte l'expression « à déterminer » reste incomplète. Cependant, son utilisation est parfois nécessaire; dans ce cas, elle devrait être accompagnée :

- a) des raisons pour lesquelles on utilise « à déterminer » (par exemple, pourquoi on n'a pas la réponse), de façon à savoir comment régler la situation
- b) des mesures à prendre pour l'éliminer, du nom de la personne qui en est responsable et de l'échéance.

### 4.3.4 Cohérente

La qualité à laquelle on fait allusion ici est la cohérence interne. Si la SEL ne concorde pas avec un document de niveau supérieur, comme une spécification relative aux exigences du système, alors elle n'est pas exacte (voir 4.3.1).

#### 4.3.4.1 Cohérence interne

On peut attribuer à la SEL la qualité de cohérence interne si et seulement si aucune série d'exigences individuelles ne sont en conflit. Il y a trois types de conflits probables :

- a) Les caractéristiques d'objets réels peuvent être en conflit. Par exemple :
  - 1) Le même rapport peut être décrit dans une exigence comme étant de format tableau et dans une autre comme étant de format texte.
  - 2) Une exigence peut énoncer que les voyants doivent être verts, tandis qu'une autre, qu'ils doivent être bleus.
- b) Il peut y avoir des conflits entre deux actions, sur le plan de la logique ou du temps. Par exemple :
  - 1) Une exigence peut préciser que le programme doit additionner deux données, alors que l'autre précise qu'il doit les multiplier.
  - 2) Une exigence peut préciser que A doit toujours suivre B, alors que l'autre précise que A et B sont simultanées.
- c) Deux ou plusieurs exigences peuvent décrire les mêmes objets réels, mais en utilisant des termes différents. Par exemple, si un programme fait une demande à l'utilisateur, dans certains cas cette demande peut être appelée une instruction, dans d'autres, une demande. L'utilisation d'une terminologie et de définitions normalisées favorise la cohérence.

#### 4.3.5 Hiérarchisée en fonction de l'importance et/ou de la stabilité

Une SEL est hiérarchisée en fonction de l'importance et/ou de la stabilité lorsque chaque exigence est dotée d'un identificateur qui désigne son importance ou sa stabilité;

En général, les exigences relatives à un logiciel n'ont pas toutes la même importance. Certaines exigences sont essentielles, surtout pour les applications critiques pour la survie, alors que d'autres ne sont que souhaitables.

Ces nuances devraient être claires et explicites pour chaque exigence d'une SEL. Il est utile d'identifier les exigences pour les raisons suivantes :

- a) Les clients portent une attention plus poussée à chaque exigence, ce qui leur permet de clarifier toute présupposition cachée qu'ils pourraient nourrir.
- b) Les concepteurs prennent les bonnes décisions et déploient les efforts appropriés à chaque partie du logiciel.

##### 4.3.5.1 Degré de stabilité

Une des méthodes utilisée pour identifier les exigences s'appuie sur la notion de stabilité. On entend par là le nombre de modifications que l'exigence devrait subir, selon les prévisions que nous permettent d'établir soit notre expérience, soit notre connaissance des événements à venir qui pourraient avoir des conséquences sur l'entreprise, les fonctions et les personnes touchées par le logiciel.

##### 4.3.5.2 Degré de nécessité

Une autre façon de classer les exigences consiste à les distinguer selon qu'elles sont essentielles, conditionnelles ou facultatives.

- a) Essentielles : le logiciel ne sera pas acceptable si ces exigences ne sont pas fournies comme elles ont été demandées.
- b) Conditionnelles : ces exigences améliorerait le logiciel, mais ne le rendrait pas inacceptable si elles n'étaient pas fournies.
- c) Facultatifs : un type d'exigences qui pourrait ou non valoir la peine. Ainsi, le fournisseur a l'occasion de proposer une solution qui dépasse les exigences de la SEL.

### 4.3.6 Vérifiable

Une SEL est vérifiable si et seulement si chaque exigence qui y est énoncée est vérifiable. Une exigence est vérifiable si et seulement s'il existe un processus déterminé et rentable grâce auquel une personne ou une machine est en mesure de vérifier que le logiciel respecte l'exigence. En général, une exigence ambiguë n'est pas vérifiable.

Les exigences non vérifiables comprennent des énoncés comme « fonctionne bien », « bon interface personne-machine » ou « se produit habituellement ». Ces exigences ne sont pas vérifiables parce qu'il est impossible de définir les termes « bien », « bon » ou « habituellement ». L'énoncé « le programme n'entrera jamais dans un boucle infinie » n'est pas vérifiable parce qu'il est théoriquement impossible de tester cette qualité.

Voici un exemple d'un énoncé vérifiable :

*60% des fois, le programme fournit un résultat dans les 20 s suivant l'événement; 100% des fois, il fournit un résultat dans les 30 s suivant l'événement.*

Cet énoncé peut être vérifié parce qu'il utilise des termes concrets et des quantités mesurables.

Si on ne peut pas mettre au point de méthode pour déterminer si le logiciel respecte une exigence, il faut la retirer ou la réviser.

### 4.3.7 Modifiable

Une SEL est modifiable si et seulement si sa structure et son style sont tels que toute modification aux exigences peut être faite facilement, complètement et de façon cohérente, tout en préservant la structure et le style de la spécification. Pour être modifiable, la SEL doit généralement :

- a) Posséder une structure cohérente et facile à utiliser, avec une table des matières, un index et des renvois explicites.
- b) Ne pas être redondante. C'est-à-dire qu'une exigence ne doit figurer qu'une seule fois dans la SEL.
- c) Exprimer chaque exigence séparément, plutôt que la confondre avec les autres exigences.

La redondance n'est pas en soi une erreur, mais elle peut facilement entraîner des erreurs. Elle peut parfois aider à rendre la SEL plus lisible, mais il peut y avoir des problèmes au moment de la mise à jour du document. Par exemple, l'exigence peut être modifiée à un seul des endroits où elle figure. La SEL devient alors incohérente. Lorsqu'il est nécessaire d'être redondant, il faut inclure des renvois explicites pour faciliter la modification de la SEL.

### 4.3.8 Traçable

Une SEL est traçable si l'origine de chaque exigence est claire et qu'on peut retracer chaque exigence au moment d'un développement à venir ou lorsqu'on veut ajouter à la documentation. Deux types de possibilités de suivi sont recommandés.

- a) Possibilité de suivi antérieur (c'est-à-dire aux étapes précédentes du développement). Pour ce faire, chaque exigence doit renvoyer explicitement à ses sources dans d'autres documents.
- b) Possibilité de suivi postérieur (c'est-à-dire à tous les documents engendrés par la SEL). Pour ce faire, toute exigence de la SEL doit posséder un nom ou un numéro de référence unique.

La possibilité de suivi postérieur est particulièrement importante lorsque le logiciel est à l'étape du fonctionnement ou de la maintenance. Lorsque les codes et les documents de conception subissent des modifications, il est essentiel de pouvoir identifier la totalité des exigences qui risquent d'être touchés par ces modifications.

## 4.4 Préparation conjointe de la SEL

Au moment d'entreprendre le processus de développement d'un logiciel, le fournisseur et le client doivent s'entendre sur les fonctions que le logiciel doit remplir. Il est important que cette entente, qui prend la forme d'une SEL, soit préparée en collaboration, parce qu'en général ni le client ni le fournisseur n'est qualifié pour rédiger une bonne SEL par lui-même.

- a) Habituellement, le client ne comprend pas le processus de conception et de développement de logiciels assez bien pour rédiger une SEL utile.
- b) Habituellement, le fournisseur ne comprend pas le problème et le champ d'activités du client assez bien pour préciser de manière satisfaisante les exigences du système.

Donc, le client et le fournisseur devraient collaborer à la production d'une SEL bien rédigée et parfaitement comprise.

Il existe une situation particulière : la description du système et de ses logiciels se font en même temps. Dans ce cas, les fonctions, les interfaces, la performance, les autres attributs et contraintes du logiciel ne sont pas prédéterminés, mais doivent être décrits conjointement, et sont sujets à la négociation et au changement. Il est alors plus difficile, mais non moins important, de respecter les critères mentionnés en 4.3. Signalons tout particulièrement qu'une SEL qui ne satisfait pas aux exigences de son système parent est incorrecte.

Ce document ne fait aucune recommandation quant aux techniques de rédaction, à la langue ni au style. Il est important, cependant, qu'une SEL soit bien rédigée. Pour obtenir de l'aide, on peut se reporter aux ouvrages généraux sur la rédaction technique.

## 4.5 Évolution de la SEL

La SEL peut avoir à évoluer pendant le développement du logiciel. Il peut être impossible de préciser certains détails au début du projet; par exemple, de décrire les formats d'écran pendant la phases de description des exigences. D'autres modifications peuvent s'imposer lorsqu'on découvre des lacunes, des manques ou des inexactitudes dans la SEL.

Il faut prendre en considération deux facteurs principaux :

- a) Les exigences devraient être spécifiées aussi complètement et aussi exactement qu'il est possible de le faire au moment de la rédaction, même si on prévoit qu'il sera inévitable d'y faire des révisions dues à l'évolution. Il faut noter le fait qu'elles sont incomplètes.
- b) Il faut mettre en œuvre un processus officiel de modification, qui vise l'identification, le contrôle, le suivi et l'enregistrement des modifications projetées. Lorsque les modifications aux exigences ont été approuvées, elles devraient être incorporées à la SEL de manière à :
  - 1) fournir une piste de vérification complète et exacte des modifications
  - 2) permettre d'étudier les parties en vigueur et les parties annulées de la SEL.

## 4.6 Réalisation de prototypes

Les prototypes s'utilisent fréquemment pendant la description des exigences d'un projet. Il existe de nombreux outils qui permettent de réaliser, rapidement et facilement, des prototypes qui démontrent certaines des caractéristiques du système. Voir aussi la norme ASTM 1340-90.

Les prototypes servent à trois fins :

- a) Le client est plus susceptible d'étudier le prototype et d'offrir ses réactions que de lire la SEL et d'y réagir. Ainsi, le prototype assure une rétroaction rapide.
- b) Le prototype met en évidence des aspects imprévus du comportement du système. Ainsi, non seulement il fournit des réponses, mais il soulève des questions. Cela permet de parachever la SEL.
- c) Une SEL fondée sur un prototype tend à subir moins de modifications pendant l'étape de développement, ce qui raccourcit les délais à cette étape.

Un prototype devrait servir à mettre au jour les exigences du logiciel. Certaines caractéristiques, comme les formats d'écran, peuvent provenir directement du prototype. D'autres exigences peuvent être découvertes en faisant des expériences sur le prototype.

## **4.7 Incorporation des critères de conception**

L'exigence impose une fonction ou un attribut visible du système. Le critère de conception, pour sa part, décrit un sous-composant particulier du système ou ses interfaces avec les autres sous-composants. Les rédacteurs de la SEL devrait faire une distinction claire entre identifier les contraintes de conception et imposer des critères spécifiques de conception. Il est à noter que chaque exigence de la SEL restreint les possibilités de conception. Ce qui ne veut pas dire, toutefois, que chaque exigence équivaut à un critère de conception.

La SEL devrait préciser les fonctions qui doivent être exécutées sur tel type de données pour produire tel résultat, à tel endroit, pour tel groupe. Elle devrait mettre l'accent sur les services à réaliser. Elle ne devrait généralement pas préciser les critères de conception tels que :

- a) La séparation du logiciel en modules
- b) La répartition des fonctions entre les modules
- c) La description du flux d'information ou de contrôle entre les modules
- d) Le choix de la structure des données

### **4.7.1 Exigences de conception nécessaires**

Dans des cas très particuliers, certaines exigences peuvent restreindre très strictement la conception. Par exemple, les exigences de sûreté ou de sécurité peuvent se refléter directement dans la conception, comme le besoin de :

- a) Garder certaines fonctions dans des modules distincts
- b) Permettre une communication restreinte entre certaines régions du programme
- c) Vérifier l'intégrité de certaines variables névralgiques

Voici des exemples de contraintes de conception valables : exigences matérielles, exigences de performance, normes de conception de logiciels, normes d'assurance-qualité des logiciels.

Donc, les exigences devraient être énoncées d'un point de vue purement externe. Lorsqu'on utilise des modèles pour illustrer des exigences, il faut se rappeler que le modèle n'indique que le comportement externe, sans préciser de critères de conception.

## **4.8 Incorporation des exigences du projet**

La SEL devrait traiter du logiciel, non pas du processus de production du logiciel.

Les exigences du projet proviennent d'une entente conclue par le client et le fournisseur sur les questions touchant la production du logiciel; elles ne devraient donc pas faire partie de la SEL. Ces questions concernent habituellement :

- a) le coût
- b) le calendrier de livraison
- c) la préparation de rapport
- d) les méthodes de développement des logiciels
- e) l'assurance-qualité
- f) les critères de validation et de vérification
- g) les procédures d'acceptation

Les exigences du projet sont précisées dans d'autres documents, en général dans le plan de développement de logiciels, le plan de l'assurance-qualité des logiciels ou l'énoncé des travaux.

## 5. LES PARTIES DE LA SEL

La présente section aborde les parties essentielles de la SEL, qui sont disposées à la fig. 1 sous forme de plan, qui peut servir d'exemple pour la rédaction d'une SEL.

### Table des matières

1. Introduction
    - 1.1 Objet
    - 1.2 Portée
    - 1.3 Définitions, acronymes et abréviations
    - 1.4 Références
    - 1.5 Vue d'ensemble
  2. Description générale
    - 2.1 Environnement
    - 2.2 Fonctions
    - 2.3 Caractéristiques des utilisateurs
    - 2.4 Contraintes
    - 2.5 Hypothèses et dépendances
  3. Exigences spécifiques (voir l'explication des exigences spécifiques possibles en 5.3.1-5.3.8. On trouvera aussi à l'annexe A plusieurs organisations différentes de cette section.)
- Annexes  
Index

### Fig. 1 – Plan type d'une SEL

Même si la SEL ne doit pas obligatoirement suivre ce plan ou utiliser les noms donnés ici aux différents articles, une bonne SEL devrait inclure tous les renseignements abordés ici.

### 5.1 Introduction (chapitre 1 de la SEL)

L'introduction devrait fournir une vue d'ensemble complète de la SEL. Elle devrait contenir les sections suivantes :

- a) Objet
- b) Portée
- c) Définitions, acronymes et abréviations
- d) Références
- e) Vue d'ensemble

#### 5.1.1 Objet (section 1.1 de la SEL)

Cet article devrait réaliser les tâches suivantes :

- a) Formuler l'objet de la SEL
- b) Préciser les destinataires de la SEL



### **5.1.2 Portée (section 1.2 de la SEL)**

Cette section devrait :

- a) identifier par leur nom le ou les logiciels à produire (par exemple : système de gestion de base de données, générateur de rapports, etc.)
- b) Expliquer ce que le ou les logiciels feront et, le cas échéant, ce qu'ils ne feront pas
- c) Décrire les applications du logiciel ainsi spécifié, y compris les avantages, objectifs et buts pertinents
- d) Être cohérente avec les énoncés similaires dans les spécifications de niveau supérieur (par exemple, les spécifications d'exigences de système), s'il y a lieu

### **5.1.3 Définitions, acronymes et abréviations (section 1.3 de la SEL)**

Cette section devrait fournir la définition de tous les termes, acronymes et abréviations dont le lecteur pourrait avoir besoin pour interpréter correctement la SEL. Ces renseignements peuvent être fournis dans des annexes à la SEL ou par des renvois à d'autres documents.

### **5.1.4 Références (section 1.4 de la SEL)**

Cette section devrait :

- a) Fournir une liste complète de tous les documents auxquels la SEL fait allusion
- b) Donner le titre, le numéro (s'il y a lieu), la date de publication et l'éditeur de chaque document
- c) Préciser les sources où on peut obtenir ces documents

Ces renseignements peuvent être fournis en annexe ou par renvoi à un autre document.

### **5.1.5 Vue d'ensemble (section 1.5 de la SEL)**

Cette section devrait :

- a) Décrire les autres sections de la SEL
- b) Expliquer comment la SEL est organisée

## **5.2 Description générale (chapitre 2 de la SEL)**

Ce chapitre de la SEL devrait décrire les facteurs généraux qui influent sur le produit et ses exigences. Elle ne donne les exigences spécifiques, mais fournit un contexte pour ces exigences (décrites en détails au chapitre 3) et en facilite la compréhension.

Le chapitre est habituellement divisé en six sections, comme suit :

- a) Environnement
- b) Fonctions
- c) Caractéristiques des utilisateurs
- d) Contraintes
- e) Hypothèses et dépendances
- f) Sous-ensembles d'exigences

### **5.2.1 Environnement (section 2.1 de la SEL)**

Cette section de la SEL devrait situer le produit dans le contexte des autres produits reliés. S'il est indépendant et parfaitement autonome, il faut le mentionner ici. S'il fait partie d'un système plus vaste, comme c'est souvent le cas, il faut énoncer ici les exigences de ce système par rapport aux fonctions du logiciel et décrire les interfaces entre le système et le logiciel.

Il peut être utile d'inclure un schéma fonctionnel montrant les principales composantes du système et leurs relations, de même que les interfaces externes.

Cette section devrait également indiquer à quelles contraintes doit se plier le logiciel, notamment :

- a) Les interfaces avec le système
- b) Les interfaces avec les utilisateurs
- c) Les interfaces avec le matériel
- d) Les interfaces avec les logiciels
- e) Les interfaces de communication
- f) Les contraintes de mémoire
- g) Les activités
- h) Les exigences d'adaptation aux sites

#### 5.2.1.1 Interfaces avec le système

Cet article devrait énumérer les interfaces avec le système, indiquer la fonction du logiciel qui satisfait à l'exigence du système et décrire l'interface qui correspond au système.

#### 5.2.1.2 Interfaces avec les utilisateurs

Cet article devrait préciser ce qui suit :

- a) *Les caractéristiques logiques de chaque interface logiciel-utilisateurs.* Cela comprend les caractéristiques de configuration (p. ex. : formats d'écran requis, disposition des pages ou des fenêtres, contenus des rapports ou des menus, disponibilité de clés de fonction programmables) nécessaires à la satisfaction des exigences du logiciel.
- b) *Tous les aspects de l'optimisation de l'interface avec la personne devant utiliser le système.* Cela peut se composer d'une simple liste de ce qui doit ou ne doit pas figurer à l'écran. Par exemple : on pourrait préciser si les messages d'erreur doivent être longs ou courts. Comme toute exigence, celles-ci doivent être vérifiables; dire par exemple, « un commis de niveau 4 doit pouvoir exécuter la fonction X en Z minutes après avoir reçu une formation d'une heure », plutôt que « un commis doit pouvoir exécuter la fonction X ». (Ce type d'exigence peut être précisée dans les Attributs d'un logiciel, dans un chapitre intitulé « Facilité d'utilisation ».)

#### 5.2.1.3 Interfaces avec le matériel

Ce paragraphe devrait préciser les caractéristiques logiques de chaque interface entre le logiciel et les composantes matériels du système. Cela comprend les caractéristiques de configuration (nombre de ports, jeux d'instruction, etc.). Cela comprend également des sujets comme les appareils qui doivent être soutenus, comment ils doivent être soutenus, ainsi que les protocoles. Par exemple, relativement au soutien des terminaux, on pourrait exiger un soutien plein page plutôt que ligne par ligne.

#### 5.2.1.4 Interfaces avec les logiciels

Ce paragraphe devrait préciser l'utilisation exigée d'autres logiciels (par exemple, un système de gestion des données, un système d'exploitation, un logiciel mathématique) et les interfaces avec d'autres applications (par exemple, les liens entre un système de comptes clients et un système de grand livre. Pour chaque logiciel exigé, il faut fournir :

- a) Le nom
- b) Le mnémonique
- c) Le numéro de spécification
- d) Le numéro de version
- e) La source

Pour chaque interface, il faut :

- a) Expliquer le but du logiciel d'interface par rapport au logiciel demandé
- b) Décrire l'interface sur le plan du contenu et du format des messages. Il n'est pas nécessaire de décrire en détail une interface bien documentée, mais il faut renvoyer au document qui en fait la description.

#### 5.2.1.5 Interfaces de communication

Ce paragraphe devrait préciser les diverses interfaces avec les réseaux de communication, comme les protocoles de réseau local, etc.

#### 5.2.1.6 Contraintes de mémoire

Ce paragraphe devrait préciser toute caractéristique des mémoires primaire et secondaire et les restrictions qui s'y appliquent.

#### 5.2.1.7 Les activités

Ce paragraphe devrait préciser les types d'activités, normales et spéciales

- a) Les différents types d'activités dans l'entreprise; par exemple, les activités initiées par l'utilisateur
- b) Les périodes d'activité interactive et les périodes d'activité sans surveillance
- c) Les fonctions de soutien du traitement des données
- d) Les activités de sauvegarde et de récupération

NOTE – Ces éléments sont parfois précisés au paragraphe des Interfaces avec les utilisateurs.

#### 5.2.1.8 Exigences d'adaptation aux sites

Ce paragraphe pourrait :

- a) Définir les exigences relatives à toute séquence de données ou d'initialisation spécifique à un site, une mission ou un mode d'exploitation donné; par exemple : valeurs d'une grille, limites de sécurité, etc.
- b) Préciser les fonctions liées au site ou à la mission qui devraient être modifiées pour adapter le logiciel à une installation particulière.

### 5.2.2 Fonctions (section 2.2 de la SEL)

Cette section de la SEL donne un résumé des fonctions principales que le logiciel doit exécuter. Par exemple, la SEL relative à un programme de comptabilité peut ici aborder la maintenance des comptes des clients, les relevés de compte et la préparation des factures, sans mentionner les très nombreux détails qu'exige chacune de ses fonctions.

Parfois, le résumé des fonctions à inclure ici peut être tiré directement de la section de la spécification de niveau supérieur (le cas échéant) qui attribue des fonctions particulières au logiciel. À noter que, à des fins de clarté :

- a) Les fonctions doivent être organisées de façon à ce que la liste puisse être comprise du client ou de quiconque lit le document pour la première fois.
- b) Les différentes fonctions et leurs relations peuvent être indiquées par des moyens textuels ou graphiques. Un tel schéma n'est pas destiné à montrer la conception du produit, mais simplement à indiquer les rapports logiques entre diverses variables.

### **5.2.3 Caractéristiques des utilisateurs (section 2.3 de la SEL)**

Cette section de la SEL décrit les caractéristiques générales des utilisateurs du produit, notamment : niveau d'instruction, expérience, connaissances techniques. Elle n'inclut pas d'exigences spécifiques, mais fournit les raisons qui justifient certaines des exigences énoncées au chapitre 3 de la SEL.

### **5.2.4 Contraintes (section 2.4 de la SEL)**

Cette section de la SEL décrit de manière générale tout autre élément qui risque de limiter les options offertes au concepteur, notamment :

- a) Politiques réglementaires
- b) Limites imposées par le matériel (p. ex. : exigences relatives à la synchronisation du signal)
- c) Interfaces avec les autres applications
- d) Exploitation en parallèle
- e) Fonctions de vérification
- f) Fonctions de contrôle
- g) Exigences relatives aux langages évolués
- h) Protocoles d'échange de signaux (par ex., XON-XOFF, ACK-NACK)
- i) Exigences de fiabilité
- j) Niveau d'importance de l'application
- k) Considérations relatives à la sécurité et à la sûreté

### **5.2.5 Hypothèses et dépendances (section 2.5 de la SEL)**

Cette section de la SEL énumère tous les facteurs qui influent sur les exigences énoncées dans la SEL. Elle ne vise pas les contraintes de conception, mais les modifications éventuelles à ces dernières, qui pourraient se répercuter sur les exigences. Par exemple, on pourrait poser comme hypothèse que le système d'exploitation sera disponible pour le matériel que l'on choisit pour faire fonctionner le logiciel. S'il n'était pas disponible, il faudrait modifier la SEL en conséquence.

### **5.2.6 Répartition des exigences (section 2.6 de la SEL)**

Cette section de la SEL indique les exigences qui peuvent être traitées dans des versions subséquentes du logiciel.

## **5.3 Exigences spécifiques (chapitre 3 de la SEL)**

Ce chapitre de la SEL contient toutes les exigences quant au logiciel, à un niveau suffisamment détaillé pour que les concepteurs puissent concevoir un système capable de satisfaire ces exigences, et que les responsables des essais puissent vérifier si le système les satisfait. Chacune des exigences devrait être perceptible de l'extérieur par les utilisateurs, les opérateurs ou les autres systèmes externes. Elles devraient au minimum décrire tous les intrants au système (les stimulus), tous les extrants (les réponses) et toutes les fonctions exécutées par le système en réaction à un intrant ou pour soutenir un extrant. Comme il s'agit souvent de la plus grosse partie de la SEL, il y a lieu de suivre les principes suivants :

- a) L'énoncé des exigences spécifiques devrait répondre à toutes les caractéristiques décrites à la section 4.3 de la présente pratique.
- b) Les exigences spécifiques devraient être accompagnées de renvois aux documents antérieurs pertinents.
- c) Chaque exigence devrait être unique et identifiable comme telle.
- d) Il faut organiser les exigences avec soin pour qu'elles soient aussi lisibles que possible.

Avant d'aborder les différentes façons d'organiser les exigences, il est utile de comprendre leurs différentes composantes, qui sont décrites aux articles suivants.

### 5.3.1 Exigences des interfaces externes

Il s'agit d'une description détaillée de tous les intrants et les extrants du logiciel. Elle devrait compléter plutôt que répéter la description des interfaces mentionnée en 5.2.

Elle devrait inclure aussi bien le contenu et la forme :

- a) Nom de l'élément
- b) But
- c) Provenance des intrants ou destination des extrants
- d) Échelle, degré de précision et/ou degré de tolérance acceptable
- e) Unités de mesure
- f) Synchronisation
- g) Rapports avec les autres intrants/extrants
- h) Format et organisation des écrans
- i) Format et organisation des fenêtres
- j) Format des données
- k) Format des commandes
- l) Messages de fin

### 5.3.2 Exigences fonctionnelles

Les exigences fonctionnelles définissent les actions principales que doit exécuter le logiciel, pour la réception et le traitement des intrants, ainsi que le traitement et la génération des extrants. On exprime généralement ces exigences sous la forme suivante : « Le système doit... »

Parmi ces exigences, on peut préciser notamment :

- a) Vérification de la validité des intrants
- b) Séquence exacte des activités
- c) Réponses aux situations anormales, y compris :
  - 1) Dépassement
  - 2) Installations de télécommunications
  - 3) Traitement des erreurs et récupération
- d) Effet des paramètres
- e) Rapports entre extrants et intrants, y compris
  - 1) Séquences intrants/extrants
  - 2) Formules de conversion d'intrant à extrant

Il peut être utile de diviser ces exigences en sous-fonctions ou en sous-processus. Cela n'implique pas nécessairement que le logiciel sera conçu selon ces mêmes divisions.

### 5.3.3 Exigences de performance

Cette section précise les exigences numériques – statiques et dynamiques – qui doivent être satisfaites par le logiciel ou par l'interaction entre l'humain et le logiciel. Les exigences statiques peuvent comprendre :

- a) Le nombre de terminaux qu'il doit supporter
- b) Le nombre d'utilisateurs qu'il doit supporter simultanément
- c) Le volume et le type de données qu'il doit traiter

Les exigences statiques sont parfois précisées dans une autre section, intitulée « Capacité ».

Les exigences numériques dynamiques peuvent comprendre, par exemple, le nombre de transactions et de tâches et le volume de données à traiter au cours d'une certaine période, dans des conditions de travail normales et lors des périodes de pointe.

Toutes ces exigences doivent être énoncées de manière à être mesurables.

Par exemple,

*95% des transactions doivent être traitées en moins de 1 s*

plutôt que

*L'opérateur ne doit pas être obligé d'attendre la fin d'une transaction.*

NOTE – Les limites numériques qui s'appliquent à une fonction spécifique sont habituellement précisées dans le paragraphe relatif à cette fonction qui concerne le traitement.

### **5.3.4 Exigences logiques relatives aux bases de données**

Cette partie décrit les exigences logiques relatives à toute information incorporée à une base de données. Elles peuvent inclure :

- a) Les types d'information utilisées par les diverses fonctions
- b) La fréquence d'utilisation
- c) Les capacités d'accès
- d) Les entités et leurs relations
- e) Les contraintes d'intégrité
- f) Les exigences relatives à la rétention des données

### **5.3.5 Contraintes de conception**

Cette partie précise les contraintes de conception qui peuvent être imposées par d'autres normes, les limites du matériel, etc.

#### **5.3.5.1 Conformité aux normes**

Cette partie précise les exigences qui sont imposées par les normes et réglementations existantes, notamment :

- a) Format des rapports
- b) Nom des données
- c) Procédures de comptabilité
- d) Traçage de vérification

Par exemple, on pourrait spécifier ici l'exigence de pouvoir retracer les activités de traitement. Cette possibilité de traçage est parfois nécessaire pour que l'application respecte des normes réglementaires ou financières minimales. Une exigence relative au traçage de vérification pourrait préciser, par exemple, que toute modification à une base de données feuille de paie doit être enregistrée dans un fichier de traçage, avec les données avant et après le traitement.

### **5.3.6 Attributs**

Certains attributs de logiciel peuvent servir d'exigences. Il est important de spécifier les attributs exigés, pour que leur matérialisation puisse être vérifiée objectivement. En voici une liste partielle :

### 5.3.6.1 Disponibilité

On précise ici les facteurs susceptibles de garantir le niveau de disponibilité spécifié pour le système dans son ensemble, comme le point de contrôle, la récupération et le redémarrage.

### 5.3.6.2 Sécurité

On précise ici les facteurs susceptibles de protéger le logiciel d'interventions accidentelles ou malveillantes (accès, utilisation, modification, destruction ou divulgation). Les exigences spécifiques dans ce domaine pourraient inclure :

- a) L'utilisation de certaines techniques cryptographiques
- b) La conservation certains journaux de bord ou certains ensembles de données historiques
- c) L'assignation de certaines fonctions à des modules distincts
- d) La restriction des communications entre certaines parties du programme
- e) La vérification de l'intégrité des données de certaines variables clés

### 5.3.6.3 Maintenabilité

On précise ici les attributs du logiciel qui sont liés à la facilité de maintenance. On peut spécifier des exigences concernant la modularité, les interfaces, la complexité, etc. Ne pas inclure ici d'exigences considérées simplement comme de bonnes méthodes de conception.

### 5.3.6.4 Transférabilité

On précise ici les attributs du logiciel qui sont liés à sa transférabilité à d'autres ordinateurs hôtes et/ou systèmes d'exploitation, notamment :

- a) Le pourcentage de composants dont le code est lié à l'ordinateur hôte
- b) Le pourcentage du codage lié à l'ordinateur hôte
- c) L'utilisation d'un langage dont la transférabilité est éprouvée
- d) L'utilisation d'un compilateur ou d'un sous-ensemble de langage en particulier
- e) L'utilisation d'un système d'exploitation en particulier

## 5.3.7 Organisation des exigences spécifiques

Pour tous les systèmes hormis les plus triviaux, le texte des exigences détaillées est plutôt long. C'est pourquoi on recommande de veiller à les organiser pour qu'elles soient aussi faciles à comprendre que possible. Il n'y a pas d'organisation idéale qui s'applique à tous les systèmes. Les divers types de systèmes se prêtent à des structures distinctes, tels qu'on les retrouvera au chapitre 3 de la SEL. Certaines de ces structures sont décrites dans les paragraphes suivants.

### 5.3.7.1 Mode d'exploitation du système

Certains systèmes se comportent très différemment selon le mode d'exploitation. Par exemple, un système de contrôle peut avoir des ensembles de fonctions différentes selon qu'il est en mode formation, en mode normal ou en mode d'urgence. Pour organiser le chapitre par mode, utiliser les plans A.1 ou A.2 de l'annexe A.

### 5.3.7.2 Classe d'utilisateur

Certains systèmes offrent différents ensembles de fonctions aux diverses classes d'utilisateurs. Par exemple, un système de commande d'ascenseurs offre des capacités différentes aux passagers, aux préposés à la maintenance et aux pompiers. Pour organiser le chapitre par classe d'utilisateur, utiliser le plan A.3 de l'annexe A.

### 5.3.7.3 Objets

Les objets sont des entités réelles qui ont une contrepartie dans le système. Par exemple, dans un système de surveillance de patients, les « objets » comprennent les patients, les capteurs, les infirmières, les salles, les médecins, les médicaments, etc. Sont associés à chaque objet une série d'attributs (de l'objet) et de fonctions (exécutées par l'objet). Ces fonctions se disent services, méthodes ou processus. Pour organiser le chapitre par objet, utiliser le plan A.4 de l'annexe A. À noter que des ensembles d'objets peuvent partager des attributs et des services. On les regroupe alors par classe.

### 5.3.7.4 Caractéristiques

Une caractéristique est un service rendu par le système, voulu de l'extérieur et qui peut exiger une séquence d'intrants pour obtenir l'effet voulu. Par exemple, dans un système téléphonique, les caractéristiques comprennent les appels locaux, le renvoi d'appel et la téléconférence. Chaque caractéristique est généralement décrite selon une séquence de stimulus et de réponses. Pour organiser le chapitre par caractéristiques, utiliser le plan A.5 de l'annexe A.

### 5.3.7.5 Stimulus

Il est utile d'organiser certains systèmes en décrivant leurs fonctions en termes de stimulus. Par exemple, on peut structurer les fonctions du système d'atterrissage d'un avion en sections sur la perte de puissance, le cisaillement éolien, les modifications subites au roulis, la vitesse verticale excessive, etc. Pour organiser le chapitre par stimulus, utiliser le plan A.6 de l'annexe A.

### 5.3.7.6 Réponses

Il est utile d'organiser certains systèmes en décrivant toutes leurs fonctions comme support à la production d'une réponse. Par exemple, les fonctions d'un système du personnel peuvent être organisées en sections qui correspondent aux fonctions liées à la production de chèques de paie, à celles liées à la production d'une liste d'employés actifs, etc. Utiliser le plan A.6 de l'annexe A, en remplaçant toutes les occurrences du mot *stimulus* par le mot *réponse*.

### 5.3.7.7 Hiérarchie de fonctions

Lorsqu'aucun des schémas d'organisation décrits ci-dessus ne sont utiles, l'ensemble des fonctions peuvent être organisées en une hiérarchie structurée selon les intrants communs, les extrants communs ou l'accès interne commun aux données. On peut utiliser des diagrammes de données et des dictionnaires de données pour indiquer les rapports entre les fonctions et les données. Pour organiser le chapitre par hiérarchie de fonctions, utiliser le plan A.7 de l'annexe A.

### 5.3.8 Autres commentaires

Il est possible, lorsqu'on envisage la rédaction d'un SEL, que plus d'une technique d'organisation lui convienne. Dans ce cas, on peut organiser les exigences spécifiques en une hiérarchie multiple adaptée aux besoins particuliers du système envisagé. Par exemple, voir l'organisation qui combine la classe d'utilisateur et les caractéristiques, au plan A.8 de l'annexe A. Toute exigence supplémentaire peut faire partie d'une section à part, à la fin de la SEL.

Il existe plusieurs notations, méthodes et outils de soutien automatisés qui peuvent faciliter la documentation des exigences. Dans la plupart des cas, leur utilité dépend de l'organisation choisie. Par



exemple, si on structure par modes, on peut recourir aux machines à états finis ou aux diagrammes d'état. Si on structure par objets, l'analyse orientée objets peut être utile. Si on structure par caractéristiques, les séquences stimulus-réponse peuvent être utiles. Si on structure par hiérarchie de fonctions, on peut faire appel aux diagrammes de données et aux dictionnaires de données.

Dans tous les plans A.1 à A.8, les sections intitulées *Exigences fonctionnelles* peuvent être décrits en langue naturelle, en pseudocode, dans un langage de définition de system ou en quatre articles intitulés *Introduction, Intrants, Traitement* et *Extrants*.

## **5.4 Renseignements d'appui**

L'information de soutien à la lecture rend la SEL plus facile à utiliser. Elle comprend :

- a) la table des matières
- b) l'index
- c) les annexes

### **5.4.1 La table de matière et l'index**

La table des matières et l'index sont très importants. Ils devraient suivre les règles habituelles de rédaction.

### **5.4.3 Annexes**

Les annexes ne sont pas toujours considérées comme faisant partie de la spécification des exigences, et ne sont pas toujours nécessaires. Elles peuvent comprendre :

- a) Des spécimens de formats d'entrée/sortie, la description des études d'analyse de coûts, les résultats d'enquêtes auprès des utilisateurs.
- b) Des renseignements d'appui et de l'information documentaire qui peuvent aider les lecteurs à comprendre la SEL.
- c) Une description des problèmes que le logiciel doit régler.
- d) Des instructions particulières d'emballage pour le codage et les supports, pour qu'ils respectent entre autres les exigences relatives à la sécurité, à l'exportation et au chargement initial.

Lorsqu'on inclut des annexes, la SEL devrait stipuler clairement si elles font partie des exigences.

## Annexe A (à titre d'information)

### A.1 Modèle du chapitre 3 de la SEL organisé par mode (version 1)

- 3 Exigences spécifiques
  - 3.1 Exigences des interfaces externes
    - 3.1.1 Interfaces avec les utilisateurs
    - 3.1.2 Interfaces avec le matériel
    - 3.1.3 Interfaces avec les logiciels
    - 3.1.4 Interfaces de communication
  - 3.2 Exigences fonctionnelles
    - 3.2.1 Mode 1
      - 3.2.2.1 Exigence fonctionnelle 1.1
      - .
      - .
      - 3.2.1.*n* Exigence fonctionnelle 1.*n*
    - 3.2.2 Mode 2
      - .
      - .
      - .
    - 3.2.*m* Mode *m*
      - 3.2.*m*.1 Exigence fonctionnelle *m*.1
      - .
      - .
      - .
      - 3.2.*m*.*n* Exigence fonctionnelle *m*.*n*
  - 3.3 Exigences de performance
  - 3.4 Contraintes de conception
  - 3.5 Attributs
  - 3.6 Autres exigences

### A.2 Modèle du chapitre 3 de la SEL organisé par mode (version 2)

- 3 Exigences spécifiques
  - 3.1 Exigences fonctionnelles
    - 3.1.1 Mode 1
      - 3.1.1.1 Exigences des interfaces externes
        - 3.1.1.1.1 Interfaces avec les utilisateurs
        - 3.1.1.1.2 Interfaces avec le matériel
        - 3.1.1.1.3 Interfaces avec les logiciels
        - 3.1.1.1.4 Interfaces de communication
      - 3.1.1.2 Exigences fonctionnelles
        - 3.1.1.2.1 Exigence fonctionnelle 1
        - .
        - .
        - .
        - 3.1.1.1.*n* Exigence fonctionnelle 1.*n*
      - 3.1.1.3 Exigences de performance

- 3.1.2 Mode 2
- .
- .
- .
- 3.1.*m* Mode *m*
- 3.2 Exigences de performance
- 3.3 Contraintes de conception
- 3.3 Attributs
- 3.4 Autres exigences

### **A.3 Modèle du chapitre 3 de la SEL organisé par classe d'utilisateur**

- 3 Exigences spécifiques
  - 3.1 Exigences des interfaces externes
    - 3.1.1 Interfaces avec les utilisateurs
    - 3.1.2 Interfaces avec le matériel
    - 3.1.3 Interfaces avec les logiciels
    - 3.1.4 Interfaces de communication
  - 3.2 Exigences fonctionnelles
    - 3.2.1 Classe d'utilisateur 1
      - 3.2.1.1 Exigence fonctionnelle 1.1
      - .
      - .
      - .
      - 3.2.1.*n* Exigence fonctionnelle 1.*n*
    - 3.2.2 Classe d'utilisateur 2
    - .
    - .
    - .
    - 3.2.*m* Classe d'utilisateur *m*
      - 3.2.*m*.1 Exigence fonctionnelle *m*.1
      - .
      - .
      - .
      - 3.2.*m*.*n* Exigence fonctionnelle *m*.*n*
  - 3.3 Exigences de performance
  - 3.4 Contraintes de conception
  - 3.5 Attributs
  - 3.6 Autres exigences

### **A.4 Modèle du chapitre 3 de la SEL organisé par objet**

- 3 Exigences spécifiques
  - 3.1 Exigences des interfaces externes
    - 3.1.1 Interfaces avec les utilisateurs
    - 3.1.2 Interfaces avec le matériel
    - 3.1.3 Interfaces avec les logiciels
    - 3.1.4 Interfaces de communication
  - 3.2 Classes/objets
    - 3.2.1 Classe/objet 1

- 3.2.1.1 Attributs (directs ou hérités)
  - 3.2.1.1.1 Attribut 1
  - .
  - .
  - 3.2.1.1.*n* Attribut *n*
  - .
  - .
- 3.2.1.2 Fonctions (services, méthodes – directes ou héritées)
  - 3.2.1.2.1 Exigence fonctionnelle 1.1
  - .
  - .
  - .
  - 3.2.1.2.*m* Exigence fonctionnelle 1.*m*
- 3.2.1.3 Messages (communications reçues ou envoyées)
- 3.2.2 Classe/objet 2
- .
- .
- .
- 3.2.*p* Classe/objet *p*
- 3.3 Exigences de performance
- 3.4 Contraintes de conception
- 3.5 Attributs
- 3.6 Autres exigences

## A.5 Modèle du chapitre 3 de la SEL organisé par caractéristiques

- 3 Exigences spécifiques
  - 3.1 Exigences des interfaces externes
    - 3.1.1 Interfaces avec les utilisateurs
    - 3.1.2 Interfaces avec le matériel
    - 3.1.3 Interfaces avec les logiciels
    - 3.1.4 Interfaces de communication
  - 3.2 Caractéristiques du système
    - 3.2.1 Caractéristique 1 du système
      - 3.2.1.1 Introduction/but de la caractéristique
      - 3.2.1.2 Séquence stimulus/réponse
      - 3.2.1.3 Exigences fonctionnelles associées
        - 3.2.1.3.1 Exigence fonctionnelle 1
        - .
        - .
        - .
        - 3.2.1.3.*n* Exigence fonctionnelle *n*
    - 3.2.2 Caractéristique 2 du système
    - .
    - .
    - .
    - 3.2.*m* Caractéristique *m* du système
    - .
    - .
    - .

- 3.3 Exigences de performance
- 3.4 Contraintes de conception
- 3.5 Attributs
- 3.6 Autres exigences

## A.6 Modèle du chapitre 3 de la SEL organisé par stimulus

- 3 Exigences spécifiques
  - 3.1 Exigences des interfaces externes
    - 3.1.1 Interfaces avec les utilisateurs
    - 3.1.2 Interfaces avec le matériel
    - 3.1.3 Interfaces avec les logiciels
    - 3.1.4 Interfaces de communication
  - 3.2 Exigences fonctionnelles
    - 3.2.1 Stimulus 1
      - 3.2.1.1 Exigence fonctionnelle 1.1
      - .
      - .
      - 3.2.1.*n* Exigence fonctionnelle 1.*n*
    - 3.2.2 Stimulus 2
    - .
    - .
    - .
    - 3.2.*m* Stimulus *m*
      - 3.2.*m*.1 Exigence fonctionnelle *m*.1
      - .
      - .
      - .
      - 3.2.*m*.*n* Exigence fonctionnelle *m*.*n*
  - 3.3 Exigences de performance
  - 3.4 Contraintes de conception
  - 3.5 Attributs
  - 3.6 Autres exigences

## A.7 Modèle du chapitre 3 de la SEL organisé par hiérarchie de fonctions

- 3 Exigences spécifiques
  - 3.1 Exigences des interfaces externes
    - 3.1.1 Interfaces avec les utilisateurs
    - 3.1.2 Interfaces avec le matériel
    - 3.1.3 Interfaces avec les logiciels
    - 3.1.4 Interfaces de communication
  - 3.2 Exigences fonctionnelles
    - 3.2.1 Flux de l'information
      - 3.2.1.1 Diagramme de flux des données 1
        - 3.2.1.1.1 Entités
        - 3.2.1.1.2 Processus pertinents
        - 3.2.1.1.3 Topologie
      - 3.2.1.2 Diagramme de flux des données 2
        - 3.2.1.2.1 Entités

- 3.2.1.2.2 Processus pertinents
    - 3.2.1.2.3 Topologie
  - .
  - .
  - .
  - 3.2.1.*n* Diagramme de flux des données *n*
    - 3.2.1.*n*.1 Entités
    - 3.2.1.*n*.2 Processus pertinents
    - 3.2.1.*n*.3 Topologie
- 3.2.2 Description des processus
  - 3.2.2.1 Processus 1
    - 3.2.2.1.1 Entités d'entrée
    - 3.2.2.1.2 Algorithme ou formule du processus
    - 3.2.2.1.3 Entités touchées
  - 3.2.2.2 Processus 2
    - 3.2.2.2.1 Entités d'entrée
    - 3.2.2.2.2 Algorithme ou formule du processus
    - 3.2.2.2.3 Entités touchées
  - .
  - .
  - .
  - 3.2.2.*m* Processus *m*
    - 3.2.2.*m*.1 Entités d'entrée
    - 3.2.2.*m*.2 Algorithme ou formule du processus
    - 3.2.2.*m*.3 Entités touchées
- 3.2.3 Spécifications relatives à la structure des données
  - 3.2.3.1 Structure 1
    - 3.2.3.1.1 Type d'enregistrement
    - 3.2.3.1.2 Champs constitutifs
  - 3.2.3.2 Structure 2
    - 3.2.3.2.1 Type d'enregistrement
    - 3.2.3.2.2 Champs constitutifs
  - .
  - .
  - .
  - 3.2.3.*p* Structure *p*
    - 3.2.3.*p*.1 Type d'enregistrement
    - 3.2.3.*p*.2 Champs constitutifs
- 3.2.4 Dictionnaire de données
  - 3.2.4.1 Élément 1
    - 3.2.4.1.1 Nom
    - 3.2.4.1.2 Représentation
    - 3.2.4.1.3 Unités/format
    - 3.2.4.1.4 Précision/exactitude
    - 3.2.4.1.5 Étendue
  - 3.2.4.2 Élément 2
    - 3.2.4.2.1 Nom
    - 3.2.4.2.2 Représentation
    - 3.2.4.2.3 Unités/format
    - 3.2.4.2.4 Précision/exactitude
    - 3.2.4.2.5 Étendue

- .
  - .
    - .
      - 3.2.4.q Élément *q*
        - 3.2.4.q.1 Nom
        - 3.2.4.q.2 Représentation
        - 3.2.4.q.3 Unités/format
        - 3.2.4.q.4 Précision/exactitude
        - 3.2.4.q.5 Étendue
- 3.3 Exigences de performance
- 3.4 Contraintes de conception
- 3.5 Attributs
- 3.6 Autres exigences

### A.8 Modèle du chapitre 3 de la SEL montrant des organisations multiples

- 3 Exigences spécifiques
  - 3.1 Exigences des interfaces externes
    - 3.1.1 Interfaces avec les utilisateurs
    - 3.1.2 Interfaces avec le matériel
    - 3.1.3 Interfaces avec les logiciels
    - 3.1.4 Interfaces de communication
  - 3.2 Exigences fonctionnelles
    - 3.2.1 Classe d'utilisateur 1
      - 3.2.1.1 Caractéristique 1
        - 3.2.1.1.1 Introduction/but de la caractéristique
        - 3.2.1.1.2 Séquence stimulus/réponse
        - 3.2.1.1.3 Exigences fonctionnelles associées
      - 3.2.1.2 Caractéristique 2
        - 3.2.1.2.1 Introduction/but de la caractéristique
        - 3.2.1.2.2 Séquence stimulus/réponse
        - 3.2.1.2.3 Exigences fonctionnelles associées
      - .
        - .
          - .
            - 3.2.1.m Caractéristique *m*
              - 3.2.1.m.1 Introduction/but de la caractéristique
              - 3.2.1.m.2 Séquence stimulus/réponse
              - 3.2.1.m.3 Exigences fonctionnelles associées
    - 3.2.2 Classe d'utilisateur 2
      - .
        - .
          - .
            - 3.2.n Classe d'utilisateur *n*
  - 3.3 Exigences de performance
  - 3.4 Contraintes de conception
  - 3.5 Attributs
  - 3.6 Autres exigences