Certified Professional for Requirements Engineering

Niveau Fondamentaux

Syllabus

Stan Bühne

Martin Glinz

Hans van Loenhoud

Stefan Staal

3.3.0 | Avril 1, 2026

Conditions d'utilisation :

1. Tout individu ou organisme de formation peut utiliser ce syllabus comme base pour une formation à la condition que les droits d'auteur (copyright) soient reconnus et cités dans les documents relatifs à cette formation. Toute personne utilisant ce syllabus à des fins de publicité doit obtenir l'accord écrit de l'IREB e.V..
2. Tout individu ou groupe d'individus peut utiliser ce syllabus comme base pour des articles, livres ou autres publications dérivées à la condition que les auteurs du syllabus et l'IREB e.V soient cités et reconnus comme source et possesseurs des droits du syllabus.

© IREB e.V.

Tous droits réservés. Aucune partie de cette publication ne peut être reproduite, stockée dans un système de recherche documentaire ou transmise sous quelque forme ou par quelque moyen que ce soit, électronique, mécanique, par photocopie, enregistrement ou autre, sans l'autorisation écrite préalable des auteurs ou de l'IREB e.V.

Remerciements

Ce programme a été initialement créé en 2007 par Karol Frühauf, Emmerich Fuchs, Martin Glinz, Rainer Grau, Colin Hood, Frank Houdek, Peter Hruschka, Barbara Paech, Klaus Pohl et Chris Rupp. Ils ont eu le soutien de : Ian Alexander, Joseph Bruder, Samuel Fricker, Günter Halmans, Peter Jaeschke, Sven Krause, Steffen Lentz, Urte Pautz, Suzanne Robertson, Dirk Schüpferling, Johannes Staub, Thorsten Weyer et Joy Beatty.

La version 3 est une révision majeure, créée par Stan Bühne, Martin Glinz, Hans van Loenhoud et Stefan Staal. Ils étaient soutenus par Karol Frühauf, Rainer Grau, Kim Lauenroth, Chris Rupp et Camille Salinesi.

Au cours de cette révision, Xavier Franch, Karol Frühauf, Rainer Grau, Frank Houdek et Thorsten Weyer ont fait part de leurs commentaires. Wim Decoutere et Hans-Jörg Steffe ont fourni des informations complémentaires.

Les revues ont été réalisées par Christoph Ebert, Barbara Paech et Chris Rupp.

Approuvé pour diffusion le 22 juillet 2020 par le Conseil de l'IREB sur recommandation de Xavier Franch et Frank Houdek.
La traduction française a été réalisée par Eric Riou du Cosquer et Olivier Denoo.

Nous remercions tous les participants pour leur contribution.

Le Copyright © 2007-2026 pour ce syllabus, concerne les auteurs mentionnés ci-dessus. Les droits ont été transférés au comité de l'IREB (International Requirements Engineering Board) e.V. (IREB), Karlsruhe, Allemagne.

Préambule

Au cours de l'été 2017, nous avons mené une enquête afin d'examiner la pertinence de la certification actuelle de niveau Fondation du Certified Professional for Requirements Engineering (CPRE) (version 2.2). L'objectif de l'enquête était d'obtenir un retour d'information sur la pertinence pratique de la certification sur le marché du point de vue des organismes de formation ainsi que des professionnels certifiés IREB travaillant comme ingénieurs [MFeA2019] des exigences. L'enquête a montré que le programme existant de Niveau Fondation du CPRE v2.2 continue généralement à répondre aux besoins les plus importants du marché et à transmettre aux candidats les connaissances pertinentes en matière d'IE. Néanmoins, nous avons reçu des commentaires selon lesquels plusieurs techniques ne sont plus utilisées dans la pratique, tandis que d'autres font défaut dans la recherche d'un développement plus itératif et adaptatif. Ce retour d'information était conforme à la perception qu'a l'IREB des changements dans le domaine de l'ingénierie des exigences (IE). Nous avons donc décidé de procéder à une révision majeure du programme du Niveau Fondation du CPRE, en supprimant les contenus obsolètes et en ajoutant de nouveaux éléments. La version 3 du syllabus reflète l'état actuel de l'IE, couvrant à la fois les approches Agile et planifiées de la spécification et de la gestion des exigences.

Les candidats à la certification CPRE selon ce syllabus doivent avoir des connaissances de base en développement de systèmes avec des approches en Agile et planifiées.

Objectif de ce document

Ce syllabus définit les objectifs d'enseignement et un résumé du contenu pédagogique pour le niveau Fondamentaux de la certification Professionnel Certifié en Ingénierie des Exigences établie par l'International Requirements Engineering Board (IREB). Les organismes de formation peuvent utiliser ce syllabus pour créer leurs supports de cours. Les étudiants peuvent l'utiliser pour se préparer aux examens.

Contenu du Syllabus

Le syllabus du niveau « Fondamentaux » couvre les besoins de toutes les personnes concernées par l'ingénierie des exigences. Il s'agit de personnes occupant des fonctions telles qu'ingénieur des exigences, analyste métier, analyste de système, Product Owner ou ou gestionnaire de produit, développeur, responsable de projet ou IT, ou expert de domaine.

Ce syllabus et le manuel correspondant utilisent l'abréviation "IE" pour Ingénierie des Exigences.

Périmètre

Le niveau fondamentaux du CPRE communique des principes de base qui sont également valables pour tout type de système (par exemple, les applications mobiles, les systèmes d'information ou les systèmes cyberphysiques). En outre, le niveau de fondation du CPRE ne suppose aucun processus de développement spécifique et n'est pas adapté à un domaine d'application particulier. Les prestataires de formation peuvent proposer une formation axée sur des types de systèmes, de processus ou de domaines d'application spécifiques, pour autant que les objectifs d'enseignement de ce programme soient entièrement couverts.

Formulations sensibles au genre

Nous nous sommes délibérément abstenus d'utiliser des formulations sexospécifiques dans ce document.

Il va sans dire qu'à l'IREB, nous soutenons les formulations sensibles au genre.
Cependant, nous voyons aussi la nécessité de formuler des matières complexes d'une manière facilement compréhensible.

Les textes exigeant une forme masculine et féminine seraient, de fait, moins lisibles et donc plus difficiles à comprendre, alors que l'objectif de ce document est de présenter et de communiquer un contenu clair et précis.
Parce que nous voulons aider les lecteurs à rester concentrés sur le contenu, nous avons fait le choix délibéré de n'utiliser que la forme masculine dans ce document.

Ce choix ne doit, en aucun cas, être interprété comme une expression de manque de respect.

Niveau de détail

Le niveau de détail de ce syllabus permet un enseignement et des examens cohérents au niveau international. Dans ce but, le syllabus apporte les éléments suivants :

- Objectifs pédagogiques généraux
- Contenu avec une description des objectifs d'enseignement
- références à d'autres publications (le cas échéant)

Objectifs d'enseignement / Niveaux d'apprentissage

Tous les modules et objectifs d'enseignement de ce syllabus se voient attribuer un niveau d'apprentissage. Les niveaux sont classés comme suit:

- **L1 : Connaître** (décrire, énumérer, caractériser, reconnaître, nommer, se souvenir, ...) – se souvenir ou récupérer des informations précédemment apprises.
- **L2 : Comprendre** (expliquer, interpréter, compléter, résumer, justifier, classer, comparer, ...) – saisir/construire un sens à partir de matériaux ou de situations donnés.
- **L3 : Appliquer** (préciser, écrire, concevoir, développer, mettre en œuvre, ...) – appliquer les connaissances et les compétences dans des situations données.

Les niveaux supérieurs comprennent les niveaux inférieurs. Notez que tous les termes du glossaire qui sont désignés comme termes de base doivent être connus (L1), même s'ils ne sont pas explicitement mentionnés dans les objectifs d'enseignement. Le glossaire est téléchargeable sur la page d'accueil de l'IREB à https://cpre.ireb.org/downloads-and-resources/downloads#cpre-glossary

Structure du Syllabus

Le syllabus est composé de 7 chapitres. principaux Chaque chapitre couvre une unité d'enseignement (UE). Les titres des chapitres principaux contiennent le niveau d'apprentissage de leurs chapitres, qui est le niveau le plus élevé de leurs sous-chapitres. La durée indique le temps qu'une formation doit investir pour ce chapitre. Les entreprises de formation sont libres d'allouer plus de temps mais doivent maintenir les proportions entre les UEs. Les termes importants utilisés dans un chapitre sont énumérés au début du chapitre.

Exemple

UE 4 Pratiques pour l'élaboration des exigences (L3)

Durée : 4 heures 30 minutes

Termes : source d'exigences, limite du système, contexte du système, élucidation des exigences, validation des exigences, partie prenante, modèle de Kano, conflit, résolution de conflit

Cet exemple montre que le chapitre 4 contient des objectifs pédagogiques de niveau L3 et que quatre heures et demie sont prévues pour l'enseignement du contenu de ce chapitre.

Chaque chapitre contient des sous-chapitres. Le titre d'un sous-chapitre indique le niveau d'apprentissage de son contenu.

Les objectifs d'enseignement (OE) sont énumérés avant le texte correspondant. La numérotation indique le sous-chapitre auquel ils appartiennent. Par exemple, l'objectif d'enseignement OE 4.2.1 est décrit dans le sous-chapitre 4.2.

Ordre des sujets dans le syllabus

L'ordre des chapitres de ce syllabus constitue un ordre logique des sujets. Cependant, les sujets ne doivent pas obligatoirement être enseignés exactement dans cet ordre. Les prestataires de formation sont libres d'enseigner le contenu dans l'ordre qu'ils jugent approprié dans le cadre de leur formation et qui correspond à leurs concepts didactiques (y compris l'imbrication de sujets provenant de différentes UE).

L'examen

Ce syllabus est la base de l'examen pour la certification IREB – niveau Fondamentaux.

Une question de cet examen peut faire référence aux contenus de plusieurs chapitres. Tous les chapitres (UE 1 à UE 7) sont susceptibles d'être traités lors de l'examen.

L'examen est sous forme d'un questionnaire à choix multiples.

Les examens peuvent avoir lieu immédiatement après une formation, mais aussi indépendamment des formations (par ex. dans un centre d'examen). Une liste des organismes de certification agréés par l'IREB est disponible sur le site web https://www.ireb.org.

Historique des versions

Version	Date	Commentaire
2.1	Novembre 2010	Version initiale
3.0.0	1er avril, 2021	Mise à jour majeure pour refléter l'état actuel de l'IE, en couvrant à la fois les approches dirigées par un plan et les approches agiles pour spécifier et gérer les exigences.
3.1.0	1er septembre, 2022	Correction d'erreurs et de références dans le document pour améliorer sa lisibilité. UE 1: Modification de l'objectif d'enseignement OE 1.3.2 en OE 1.1.2. Mise à jour de OE 1.2.2 et 1.3.1 UE 3 Mise à jour de l'objectif d'enseignement OE 3.1.3 UE 3.1.2: Reprise du paragraphe sur les niveaux d'abstraction UE 3.4: Tous les types de modèles ne devant pas être appliqués au Niveau Fondation ont été déplacés dans une nouvelle section 3.4.6 UE 3.6: Mise à jour du titre "Documents et Structures de Documentation des Exigences" UE 4 UE 4.1: La description des parties prenantes et des rôles des parties prenantes a été précisée. UE 4.2: L'introduction et la justification pour les techniques de conception et de génération d'idées ont été mises à jour et précisées. UE 4.3: Mise à jour du titre "Résoudre les Conflits concernant les Exigences" UE 6 OE 6.3.1 et OE 6.5.2 ont été légèrement modifiés
3.1.1	1er janvier 2024	Mise à jour de la conception de l'entreprise

Version	Date	Commentaire
3.2.0	26 février 2024	Corrections de mise en page pour la nouvelle conception d'entreprise
3.3.0	1er Avril 2026	Modifications relatives au contenu EO 3.4.3 et EU 3.4.6 : Le modèle d'interaction a été supprimé des objectifs d'apprentissage. EU 3.8 : clarification indiquant que les critères de qualité mentionnés sont des critères recommandés et ne constituent pas une liste exhaustive. Modifications rédactionnelles Mises à jour rédactionnelles des références et de la section 4.1, y compris l'ajout d'une référence en amont au principe 3. Mise à jour de la section Références et vérification de tous les liens hypertextes.

Table des matières

Table des matières 8

1 Introduction et aperçu de l'ingénierie des exigences (L2) 11

1.1 Ingénierie des exigences : Quoi ? (L1) 11

1.2 Ingénierie des exigences : Pourquoi ? (L2) 12

1.3 Ingénierie des exigences : Où ? (L2) 12

1.4 Ingénierie des exigences : Comment ? (L1) 13

1.5 Le rôle et les tâches d'un ingénieur des exigences (L1) 13

1.6 Que faut-il savoir sur l'ingénierie des exigences ? (L1) 13

2 Principes fondamentaux de l'ingénierie des exigences (L2) 14

2.1 Aperçu des principes (L1) 14

2.2 Les principes expliqués (L2) 14

3 Produits d'activités et pratiques de documentation (L3) . 20

3.1 Produits d'activités dans l'ingénierie des exigences (L2) 21

3.1.1 Caractéristiques des produits d'activités (L1) 21

3.1.2 Niveaux d'abstraction (L2) 22

3.1.3 Niveau de détail (L2) 22

3.1.4 Aspects à prendre en compte dans les produits d'activités (L1) 23

3.1.5 Lignes directrices pour la documentation générale (L1) 23

3.1.6 Planifier les produits d'activités à utiliser (L1) 24

3.2 Produits d'activités basés sur le langage naturel (L2) 24

3.3 Produits d'activités basés sur des templates (L3) 25

3.4 Produits d'activités basés sur des modèles (L3) 26

3.4.1 Le rôle des modèles dans l'ingénierie des exigences (L2) 26

3.4.2 Modélisation du contexte (L2) 27

3.4.3 Modélisation de la structure et des données (L3) 28

3.4.4 Modélisation des Fonctions et des Flux (L3) 28

3.4.5 Modélisation de l'Etat et du Comportement (L2) 29
3.4.6 Autres Types de Modèles dans l'Ingénierie des Exigences (L1) 29
3.5 Glossaires (L2) 30
3.6 Documents et Structures de Documentation des Exigences (L2) 30
3.7 Prototypes en ingénierie des exigences (L1) 31
3.8 Critères de qualité pour les produits d'activités et les exigences (L1) 32
4 Pratiques pour l'élaboration des exigences (L3) 33
4.1 Sources des exigences (L3) 33
4.2 Élucidation des exigences (L2) 35
4.3 Résoudre les Conflits concernant les Exigences (L2) 36
4.4 Validation des exigences (L2) 37
5 Processus et Structure de Travail (L3) 39
5.1 Facteurs d'influence (L2) 39
5.2 Facettes du processus d'ingénierie des exigences (L2) 40
5.3 Configuration d'un processus d'ingénierie des exigences (L3) 42
6 Pratiques de gestion des exigences (L2) 44
6.1 Qu'est-ce que la gestion des exigences ? (L1) 44
6.2 Gestion du cycle de vie (L2) 44
6.3 Contrôle des versions (L2) 45
6.4 Configurations et baselines (L1) 45
6.5 Attributs et vues (L2) 45
6.6 Traçabilité (L1) 46
6.7 Gestion du changement (L1) 46
6.8 Priorisation (L1) 47

7 Support des outils (L2) 48

7.1 Outils dans l'ingénierie des exigences (L1) 48

7.2 Mise en place des Outils (L2) 49

Références 50

1 Introduction et aperçu de l'ingénierie des exigences (L2)

Objectif : savoir ce qu'est l'IE et comprendre la valeur de l'IE

Durée : 1 heure

Termes : Exigence, spécification des exigences, ingénierie des exigences (IE), partie prenante, système, ingénieur des exigences

Objectifs pédagogiques

OE 1.1.1 Se souvenir de la terminologie fondamentale (L1)
OE 1.1.2 Comprendre les différents types d'exigences (L2)
OE 1.2.1 Expliquer la valeur de l'IE (L2)
OE 1.2.2 Énumérer les symptômes d'une IE inadéquate (L1)
OE 1.3.1 Savoir où l'IE peut être appliquée et où les exigences entrent en jeu (L1)
OE 1.4.1 Connaître les principales tâches de l'IE et savoir qu'un processus d'IE doit être adapté pour les réaliser (L1)
OE 1.5.1 Caractériser le rôle et les tâches d'un ingénieur des exigences (L1)
OE 1.6.1 Se souvenir de ce qu'un ingénieur des exigences doit apprendre (L1)

1.1 Ingénierie des exigences : Quoi ? (L1)

Les personnes et les organisations ont des désirs et des besoins de construire de nouvelles choses ou de faire évoluer des choses existantes. Nous appelons ces besoins des *exigences*.

Les choses à construire ou à faire évoluer peuvent être :

- Des *Produits* fournis aux clients
- Des *Services* mis à la disposition des clients
- Tout autre *livrable*, tel que des systèmes, des procédures ou des outils qui aident les personnes et les organisations à atteindre un objectif spécifique
- Des *Compositions* ou *composants* de produits, services ou autres livrables

Toutes ces choses peuvent être considérées comme des *systèmes*. Dans ce syllabus, nous utilisons le terme "*système*" pour désigner toutes sortes de choses pour lesquelles *les parties prenantes* ont des exigences. Les *parties prenantes* sont des personnes ou des organisations qui influencent les exigences d'un système ou qui sont touchées par ce système.

L'objectif de l'IE est de spécifier et de gérer les exigences relatives aux systèmes de telle sorte que les systèmes mis en œuvre et déployés répondent aux souhaits et aux besoins des parties prenantes.

En matière d'IE, on distingue trois types d'exigences [Glin2025]:

- Les *exigences fonctionnelles* concernent un résultat ou un comportement qui doit être fourni par une fonction d'un système. Cela comprend les exigences relatives aux données ou à l'interaction d'un système avec son environnement.

- Les *exigences* de qualité concernent les préoccupations de qualité qui ne sont pas couvertes par les exigences fonctionnelles, telles que la performance, la disponibilité, la sécurité ou la fiabilité.
- Les *contraintes* sont des exigences qui limitent l'espace de la solution au-delà de ce qui est nécessaire pour répondre aux exigences fonctionnelles et aux exigences de qualité données.

1.2 Ingénierie des exigences : Pourquoi ? (L2)

Une IE adéquate ajoute de la *valeur* dans le processus de développement et d'évolution d'un système :

- Réduire le risque de développer le mauvais système
- Une meilleure compréhension du problème
- Base pour l'estimation de l'effort et du coût de développement
- Prérequis pour tester le système

Les symptômes typiques d'une IE inadéquate sont des exigences manquantes, peu claires ou incorrectes. Cela est dû en particulier à :

- Se précipiter dans la construction du système
- Des problèmes de communication entre les parties concernées
- L'hypothèse selon laquelle les exigences sont évidentes
- Une formation et des compétences insuffisantes en matière d'IE

1.3 Ingénierie des exigences : Où ? (L2)

L'IE peut être appliquée aux exigences de tout type de système. Cependant, le cas d'application dominant pour l'IE aujourd'hui est représenté par les systèmes dans lesquels les logiciels jouent un rôle majeur. Ces systèmes sont généralement constitués de composants logiciels, d'éléments physiques et d'éléments organisationnels.

Les exigences peuvent se présenter sous forme de :

- Les *Exigences système* – ce qu'un système doit faire
- Les *Exigences des parties prenantes* – ce que les parties prenantes attendent de leur point de vue
- Les *Exigences des utilisateurs* – ce que les utilisateurs veulent de leur point de vue
- Les *Exigences du domaine* – propriétés requises du domaine
- Les *Exigences métier* – buts, objectifs et besoins métier d'une organisation

1.4 Ingénierie des exigences : Comment ? (L1)

Les principales tâches en matière d'IE sont l'élucidation (4.2), la documentation (3), la validation (4.4) et la gestion (6) des exigences. Le support d'outils (7) peut aider à accomplir ces tâches. L'analyse des exigences et la résolution des conflits d'exigences (4.3) sont considérées comme faisant partie de l'élucidation. Afin de mener à bien les activités d'IE, un processus d'IE approprié doit être adapté à partir d'un large éventail de possibilités (5).

1.5 Le rôle et les tâches d'un ingénieur des exigences (L1)

L'ingénieur des exigences n'est généralement pas un titre de poste, mais un *rôle* joué par les personnes en charge de :

- Elucider, documenter, valider et/ou gérer les exigences dans le cadre de leurs fonctions.
- Avoir une connaissance approfondie de l'IE.
- Combler le fossé entre le problème et les solutions potentielles.

En pratique, les analystes métier, les spécialistes des applications, les Product Owner, les ingénieurs systèmes et même les développeurs jouent le rôle d'un ingénieur des exigences.

1.6 Que faut-il savoir sur l'ingénierie des exigences ? (L1)

Ce programme couvre l'ensemble des compétences fondamentales qu'un Ingénieur des exigences doit apprendre. Il englobe les principes fondamentaux de l'IE (2), la manière de documenter les exigences sous diverses formes (3), la manière d'élaborer des exigences avec diverses pratiques (4), la manière de définir et de travailler avec des processus d'IE appropriés (5), la manière de gérer les exigences existantes (6) et la manière d'utiliser le soutien des outils (7).

2 Principes fondamentaux de l'ingénierie des exigences (L2)

Objectif : connaître et comprendre les principes de l'IE

Durée : 1 heure 30 minutes

Termes : Contexte, exigence, ingénierie des exigences (IE), partie prenante, compréhension partagée, validation

Objectifs pédagogiques

OE 2.1.1 Énumérer les principes de l'IE (L1)
OE 2.2.1 Se souvenir des termes associés aux principes (L1)
OE 2.2.2 Expliquer les principes et les raisons de leur importance (L2)

2.1 Aperçu des principes (L1)

L'IE est régie par un ensemble de principes fondamentaux qui s'appliquent à toutes les tâches, activités et pratiques de l'IE. Les neuf principes suivants constituent la base des pratiques présentées dans les chapitres suivants de ce syllabus.

1. Orientation vers les valeurs: Les exigences sont un moyen pour atteindre une fin, pas une fin en soi
2. Parties prenantes: L'IE consiste à satisfaire les désirs et les besoins des parties prenantes
3. Compréhension commune: Le développement réussi de systèmes est impossible sans une base commune
4. Contexte: Les systèmes ne peuvent être compris isolément
5. Problème – Exigence – Solution: Un triple entrelacement inévitable
6. Validation: Les exigences non validées sont inutiles
7. Évolution: L'évolution des exigences n'est pas un accident, mais le cas normal
8. Innovation: Il ne suffit pas d'en faire plus
9. Un travail systématique et discipliné: On ne peut pas s'en passer dans l'IE

2.2 Les principes expliqués (L2)

Principe 1 - Orientation vers la valeur : Les exigences sont un moyen d'atteindre une fin, et non une fin en soi

La valeur d'une exigence est égale à son bénéfice moins le coût de l'élucidation, de la documentation, de la validation et de la gestion de l'exigence. L'avantage d'une exigence est son degré de contribution pour :

- Construire des systèmes qui satisfont les désirs et les besoins de leurs parties prenantes.
- Réduire le risque de défaillance et de retouches coûteuses lors du développement du système.

Principe 2 - Les parties prenantes : L'IE vise à satisfaire les désirs et les besoins des parties prenantes

Comme l'IE consiste à comprendre les désirs et les besoins des parties prenantes, une des tâches essentielles de l'IE est de gérer correctement les parties prenantes. Chaque partie prenante a un rôle à jouer dans le contexte du système à construire – par exemple, l'utilisateur, le client, l'opérateur ou le régulateur. Une partie prenante peut également avoir plusieurs rôles en même temps. Pour les rôles de parties prenantes portés par trop de personnes ou par des personnes inconnues, des descriptions fictives d'archétypes connues sous le nom de *personas* peuvent être définies comme substitut. Il ne suffit pas de prendre en compte uniquement les exigences des utilisateurs finaux ou des clients. Cela signifierait que nous risquerions de passer à côté d'exigences critiques d'autres parties prenantes. Les utilisateurs qui fournissent un retour d'information sur un système en cours d'utilisation doivent également être considérés comme des parties prenantes.

Les parties prenantes peuvent avoir des besoins et des points de vue différents, ce qui peut entraîner des exigences contradictoires. C'est une tâche de l'IE d'identifier et de résoudre ces conflits.

Il est essentiel d'impliquer les bonnes personnes dans les rôles des parties prenantes concernées pour assurer le succès de l'IE. Les pratiques d'identification, de priorisation et de travail avec les parties prenantes sont décrites dans 4.

Principe 3 - Compréhension partagée : Le développement réussi de systèmes est impossible sans une base commune

L'IE crée, encourage et garantit une compréhension commune entre les parties concernées : parties prenantes, ingénieurs des exigences et développeurs. Il existe deux formes de compréhension commune :

- Une *compréhension commune explicite* (obtenue par des exigences documentées et acceptées)
- Une *compréhension commune implicite* (basée sur une connaissance partagée des besoins, des visions, du contexte, etc.)

La connaissance du domaine, une collaboration antérieure réussie, une culture et des valeurs communes et une confiance mutuelle sont des facteurs favorables à une compréhension commune, tandis que la distance géographique, l'externalisation ou les grandes équipes à forte rotation sont des obstacles.

Les pratiques éprouvées pour parvenir à une compréhension commune comprennent la création de glossaires (3.5), la création de prototypes (3.7) ou l'utilisation d'un système existant comme point de référence. Les pratiques d'évaluation de la compréhension commune comprennent des exemples de résultats attendus, l'exploration de prototypes ou l'estimation du coût de mise en œuvre d'une exigence. La pratique la plus importante pour réduire l'impact des malentendus consiste à utiliser un processus avec de courtes boucles de rétroaction (5).

Principe 4 - Contexte : Les systèmes ne peuvent être compris isolément

Les systèmes sont intégrés dans un *contexte*. Sans comprendre ce contexte, il est impossible de spécifier correctement un système. Dans le domaine de l'IE, le contexte d'un système est la partie de l'environnement d'un système qui est pertinente pour comprendre le système et ses exigences. La limite du *système* est la limite entre un système et son contexte environnant. Au départ, la limite du système n'est souvent pas claire, et elle peut même changer avec le temps.

Clarifier la limite du système et définir les interfaces externes entre le système et les éléments de contexte avec lesquels il interagit sont de véritables tâches d'IE. En même temps, il faut déterminer le *périmètre* du système, c'est-à-dire l'ensemble des choses qui peuvent être façonnées et conçues lors du développement du système. Nous devons également tenir compte de ce que l'on appelle la *limite du contexte* qui sépare la partie de l'environnement d'un système concernée par l'IE du reste du monde.

Lors de la spécification d'un système, il ne suffit pas de prendre en compte uniquement les exigences à l'intérieur de la limite du système. L'IE doit également prendre en considération :

- Les changements dans le contexte qui peuvent avoir un impact sur les exigences système.
- Les exigences du monde réel pertinentes pour le système (et comment les mettre en correspondance avec les exigences du système).
- Les hypothèses de contexte qui doivent être retenues pour faire fonctionner le système et répondre aux exigences du monde réel.

Principe 5 - Problème - Exigence - Solution : Un triple entrelacement inévitable

Un *problème* survient lorsque les parties prenantes ne sont pas satisfaites de la situation telle qu'elle est. Les *exigences* saisissent ce dont les parties prenantes ont besoin pour se débarrasser du problème ou le simplifier. Un système socio-technique qui satisfait à ces exigences constitue une *solution*.

Les problèmes, les exigences et les solutions ne se présentent pas nécessairement dans cet ordre. Les idées de solutions peuvent créer des besoins utilisateurs qui doivent être élaborés en exigences et mis en œuvre dans une solution réelle. C'est généralement le cas lorsqu'on innove.

- Les problèmes, les exigences et les solutions sont étroitement liés : ils ne peuvent être abordés isolément.
- Néanmoins, les ingénieurs des exigences visent à séparer autant que possible les problèmes, les exigences et les solutions les uns des autres lors de la réflexion, de la communication et de la documentation. Cette séparation des préoccupations rend les tâches dr l'IE plus faciles à gérer.

Principe 6 -Validation : Les exigences non validées sont inutiles

Enfin, nous devons valider que le système déployé répond aux désirs et aux besoins des parties prenantes. Afin de contrôler le risque d'insatisfaction des parties prenantes dès le début, la validation des exigences doit commencer dès l'IE. Nous devons vérifier si :

- Un accord sur les exigences a été obtenu entre les parties prenantes,
- Les souhaits et les besoins des parties prenantes sont couverts de manière adéquate par les exigences,
- Les hypothèses relatives au contexte (voir Principe 4ci-dessus) sont raisonnables.

Les pratiques de validation des exigences sont examinées dans le document 4.4.

Principe 7 - Évolution : Modifier des exigences n'est pas un accident, mais le cas normal

Les systèmes et leurs exigences sont sujets à *évolution*. Cela signifie qu'ils changent constamment. Les demandes de changement d'une exigence ou d'un ensemble d'exigences pour un système peuvent être causées, par exemple, par :

- Des modifications de processus métier
- Des concurrents qui lancent de nouveaux produits ou services

- Des clients qui changent leurs priorités ou leurs opinions
- Des changements de technologie
- Des changements de lois ou de règlements
- Des retours d'informations d'utilisateurs du système demandant de nouvelles fonctionnalités ou des modifications

En outre, les exigences peuvent changer en raison des retours des parties prenantes lors de la validation des exigences, de la détection de défauts dans les exigences précédemment élucidées ou de l'évolution des besoins des parties prenantes.

En conséquence, les ingénieurs des exigences doivent poursuivre deux objectifs apparemment contradictoires :

- Permettre des changements d'exigences
- Maintenir la stabilité des exigences

Des détails sur la manière d'y parvenir sont discuttés dans le document 6.7.

Principe 8 -Innovation : Il ne suffit pas d'en faire plus

En donnant aux parties prenantes exactement ce qu'elles veulent, on laisse passer l'occasion de construire des systèmes qui satisfont les besoins des parties prenantes mieux que ce qu'elles attendent. Une bonne IE s'efforce non seulement de satisfaire les parties prenantes, mais aussi de les rendre heureuses, excitées ou de les faire se sentir en sécurité. C'est à cela que correspond l'innovation en fin de compte.

L'IE façonne des systèmes innovants :

- À petite échelle, en s'efforçant d'offrir de nouvelles fonctionnalités et une facilité d'utilisation.
- A grande échelle, en s'efforçant de trouver de nouvelles idées disruptives.

Dans 4.2, plusieurs techniques permettant de favoriser l'innovation dans le domaine de l'IE sont examinées.

Principe 9 - Travail systématique et discipliné : On ne peut pas s'en passer en IE

Il est nécessaire de s'appuyer sur des processus et des pratiques appropriés pour élucider, documenter, valider et gérer les exigences de manière systématique, quel que soit le processus de développement utilisé. Même lorsqu'un système est développé de manière ad hoc, une approche systématique et disciplinée de l'IE améliore la qualité du système qui en résulte.

Il n'existe pas de processus ou de pratique unique en matière d'IE qui fonctionne bien dans chaque situation donnée ou du moins dans la plupart des situations. Un travail systématique et discipliné signifie que les ingénieurs des exigences :

- Adaptent leurs processus et leurs pratiques au problème, au contexte et à l'environnement donnés.
- N'utilisent pas toujours le même processus et le même ensemble de pratiques.
- Ne réutilisent pas sans réflexion les processus et les pratiques issus de précédents projets d'IE ayant réussi.

Pour chaque effort d'IE, il faut choisir les processus, les pratiques et les produits d'activités qui correspondent le mieux à la situation spécifique. Les détails sont précisés dans 3, 4, 5 et 6.

3 Produits d'activités et pratiques de documentation (L3)

Objectif : Comprendre le rôle fondamental des produits d'activités dans l'IE et être capable de créer des produits d'activités

Durée: 7 heures

Termes : produit d'activités, produits d'activités basés sur le langage naturel, produits d'activités basés sur des templates, produits d'activités basés sur des modèles, glossaire, critères de qualité, spécification des exigences

Objectifs pédagogiques

OE 3.1.1 Connaître les caractéristiques des produits d'activité de l'IE et énumérer les types de produits d'activités fréquemment utilisés (L1)
OE 3.1.2 Savoir à quoi chaque produit d'activités peut être utilisé et connaître la durée de vie des produits d'activités (L1)
OE 3.1.3 Expliquer les différents niveaux d'abstraction pour les exigences, y compris la manière de choisir les niveaux d'abstraction et les niveaux de détail appropriés (L2)
OE 3.1.4 Connaître les aspects à prendre en compte dans les produits d'activités et les relations entre ces aspects (L1)
OE 3.1.5 Nommer les directives générales de documentation (L1)
OE 3.1.6 Décrivez pourquoi il est utile de planifier les produits d'activités à utiliser (L1)
OE 3.2.1 Connaître les produits d'activités basés sur le langage naturel et leurs avantages et inconvénients (L1)
OE 3.2.2 Expliquer les règles de rédaction de bonnes exigences basées sur le langage naturel (L2)
OE 3.3.1 Connaître les catégories de produits d'activités basés sur des templates et leurs avantages et inconvénients (L1)
OE 3.3.2 Spécifier une exigence individuelle et une User Story en utilisant un gabarit de phrase(L3)
OE 3.3.3 Spécifier un cas d'utilisation en utilisant un modèle de formulaire (L3)
OE 3.4.1 Comprendre le rôle, l'objectif et l'utilisation des modèles dans l'IE (L2)
OE 3.4.2 Comprendre les avantages et les limites de la modélisation en ER (L2)
OE 3.4.3 Connaître les termes modèle, langage de modélisation, modèle d'activité, diagramme d'activité, modèle de classe, diagramme de classe, modèle de contexte, diagramme de contexte, modèle de domaine, modèle d'objectif, modèle de processus, diagramme de séquence, diagramme d'état, machine à état, diagramme de machine à état, cas d'utilisation, diagramme de cas d'utilisation (L1)
OE 3.4.4 Comprendre comment sélectionner un type de modèle approprié pour spécifier les exigences dans une situation donnée (L2)
OE 3.4.5 Comprendre et interpréter des modèles simples, écrits en UML le cas échéant, des types suivants : modèles de contexte, cas d'utilisation et diagrammes de cas d'utilisation, modèles de domaine, modèles de classe, modèles d'activités, modèles de processus et diagrammes d'état (L2)
OE 3.4.6 Spécifier un modèle simple des données d'un système ou des objets d'un domaine en utilisant un diagramme de classe UML (L3)
OE 3.4.7 Spécifier une fonction de système simple ou un processus métier par un diagramme d'activité UML (L3)
OE 3.5.1 Expliquer le but des glossaires et comment en créer un (L2)

OE 3.6.1 Connaître les documents de spécification des exigences fréquemment utilisés (L1)
OE 3.6.2 Expliquer quelles structures de documents servent quel objectif et les critères de structuration (L2)
OE 3.7.1 Connaître les différents types de prototypes et leur utilisation (L1)
OE 3.8.1 Connaître les critères de qualité pour les exigences uniques (L1)
OE 3.8.2 Connaître les critères de qualité pour les produits d'activités (L1)

3.1 Produits d'activités dans l'ingénierie des exigences (L2)

Un produit d'activités est un résultat intermédiaire ou final enregistré généré dans un processus de travail. Il existe toute une série de produits d'activités dans le domaine de l'IE, allant, par exemple, d'esquisses graphiques éphémères à des documents de spécifications d'exigences contractuelles de plusieurs centaines de pages livrés formellement, en passant par des collections évolutives de User Stories.

3.1.1 Caractéristiques des produits d'activités (L1)

Les produits d'activités se caractérisent par leur but, leur représentation, leur taille et leur durée de vie. Les produits d'activités suivants sont fréquemment utilisés dans la pratique pour les objectifs donnés. Notez qu'un produit d'activités peut contenir d'autres produits d'activités.

- Les produits d'activités pour une exigence unique comprennent des exigences individuelles et des User Stories.
- Les produits d'activités pour un ensemble cohérent d'exigences comprennent des cas d'utilisation, des modèles graphiques d'un certain type (3.4), des descriptions de tâches, des descriptions d'interfaces externes et des Epics.
- Les produits d'activités qui constituent des documents complets ou des structures de documentation comprennent les spécifications des exigences système, les backlogs de produit et de sprint, et les story maps.
- Parmi les autres produits d'activités on trouve des glossaires, des notes textuelles, des croquis graphiques et des prototypes.

Les produits d'activités peuvent être *représentés* sous différentes formes :

- Basé sur le langage naturel (3.2)
- Basé sur un template (3.3)
- Basé sur un modèle (3.4)
- Autres représentations, telles que des dessins ou des prototypes (3.7)

La plupart des produits d'activités sont *stockés* électroniquement sous forme de fichiers, dans des bases de données ou dans des outils d'IE. Des produits d'activités informels et temporaires peuvent également être stockés sur d'autres supports – par exemple, du papier ou des post-it sur un tableau Kanban.

Lorsque l'on examine la durée de vie des produits d'activités, on distingue trois catégories :

- *Produits d'activités temporaires :* créés pour soutenir la communication et créer une compréhension commune.
- *Produits d'activités évolutifs :* se développent typiquement en plusieurs itérations au fil du temps ; ont besoin de certaines métadonnées (EU 6.5), le contrôle du changement peut s'appliquer.
- *Produits d'activités durables :* ont été *établis* ou *publiés* ; nécessitent un ensemble complet de métadonnées (EU 6.5) ; le processus de changement doit être suivi (EU6.3 , EU6.4)

Un produit d'activités temporaire peut être converti en un produit d'activités évolutif (en le conservant et en y ajoutant des métadonnées). De même, un produit d'activités temporaire ou évolutif peut devenir durable en étant associé à une baseline ou livré.

3.1.2 Niveaux d'abstraction (L2)

Les exigences existent généralement à de nombreux *niveaux d'abstraction différents* – par exemple, depuis des exigences de haut niveau pour un nouveau processus métier, jusqu'à des exigences à un niveau très détaillé, comme la réaction d'un composant logiciel spécifique à un événement exceptionnel.

Le choix du niveau d'abstraction approprié dépend du sujet à spécifier et de l'objectif de la spécification. Il est toutefois important de ne pas mélanger des exigences qui se situent à des niveaux d'abstraction différents. Dans les produits d'activités de petite taille et de moyenne taille, les exigences doivent être plus ou moins au même niveau d'abstraction.

Dans les grands produits d'activités tels que la spécification des exigences système, les exigences des différents niveaux d'abstraction doivent rester séparées en structurant le document de spécification en conséquence (3.6). Une exigence à un haut niveau d'abstraction peut être raffinée en plusieurs exigences détaillées à des niveaux inférieurs, plus concrets.

Lorsque des exigences de haut niveau, métier ou issues des parties prenantes, sont exprimées dans des produits d'activités durables, tels que les spécifications des exigences métier, les spécifications des exigences des parties prenantes ou les documents de vision, elles précèdent la spécification des exigences système. Dans d'autres projets, les exigences métier, les exigences des parties prenantes et les exigences système peuvent évoluer conjointement.

3.1.3 Niveau de détail (L2)

Le niveau de *détail* auquel les exigences doivent être spécifiées dépend de plusieurs facteurs, notamment

- Le contexte du problème et du développement
- Le degré de compréhension commune du problème
- Le degré de liberté laissé aux concepteurs et aux développeurs

- Disponibilité d'un retour d'information rapide des parties prenantes pendant la conception et l'implémentation
- Coût vs. valeur d'une spécification détaillée
- Les normes imposées et les contraintes réglementaires

Plus le niveau de détail des exigences spécifiées est élevé, plus le risque d'obtenir finalement quelque chose d'inattendu ou de non spécifié est faible. Cependant, le coût de la spécification augmente à mesure que le niveau de détail augmente.

3.1.4 Aspects à prendre en compte dans les produits d'activités (L1)

Lors de la spécification des exigences dans les produits d'activités, différents aspects doivent être pris en compte.

1. Les exigences sont classées selon leur type (1.1) en :
   a) Exigences fonctionnelles
   b) Exigences qualité
   c) Contraintes
2. Les exigences fonctionnelles se concentrent sur différents aspects de la fonctionnalité d'un système :
   a) Structure et données
   b) Fonction et flux
   c) État et comportement
3. Enfin, les exigences ne peuvent être comprises que dans leur contexte (Principe 4 en 2) :
   a) *Contexte du système*, y compris les acteurs externes
   b) *Limite* du système et interfaces externes

Il existe de nombreuses interrelations et dépendances entre les aspects mentionnés. Par exemple, une requête émise par un utilisateur (contexte) peut déclencher une transition d'état (état et comportement) qui déclenche une action suivie d'une autre action (fonction et flux) qui nécessite des données (structure et données) pour fournir un résultat à l'utilisateur (contexte) dans un intervalle de temps donné (qualité).

Certains produits d'activités se concentrent sur un aspect spécifique et font abstraction des autres aspects. C'est notamment le cas des modèles d'exigences (3.4). D'autres produits d'activités, tels qu'une spécification des exigences du système, couvrent tous ces aspects. Lorsque différents aspects sont documentés dans des produits d'activités distincts ou dans des chapitres distincts du même produit d'activités, ces produits ou chapitres doivent être cohérents entre eux.

3.1.5 Lignes directrices pour la documentation générale (L1)

Indépendamment des techniques utilisées, les lignes directrices suivantes s'appliquent à la création de produits d'activités :

- Sélectionnez un type de produit d'activités correspondant à l'*objectif visé*.
- *Évitez* la *redondance* en référençant le contenu au lieu de répéter le même contenu.
- *Veiller à ce qu'il n'y ait pas* d'*incohérences* entre les produits d'activités, en particulier lorsqu'ils couvrent des aspects différents.
- *Utilisez les termes de manière cohérente*, comme définis dans le glossaire.
- *Structurer* les produits d'activités de manière appropriée.

3.1.6 Planifier les produits d'activités à utiliser (L1)

Chaque configuration de projet et chaque domaine étant différents, l'ensemble des produits d'activités qui en résultent doit être défini pour chaque situation. Par conséquent, les questions suivantes doivent faire l'objet d'un accord :

- Dans quels produits d'activités les exigences doivent-elles être enregistrées et à quelles fins ?
- Quels sont les niveaux d'abstraction à prendre en compte ?
- Jusqu'à quel niveau de détail les exigences doivent-elles être documentées pour chaque niveau d'abstraction ?
- Comment les exigences seront-elles représentées dans ces produits d'activités ?

Les produits d'activités à utiliser doivent être définis à un stade précoce du projet. Cela présente plusieurs avantages, comme :

- Aider à planifier les efforts et les ressources.
- Veiller à ce que les notations appropriées soient utilisées.
- Veiller à ce que tous les résultats soient enregistrés dans les bons produits d'activités.
- Veiller à ce qu'aucun remaniement majeur de l'information ou "mise au point finale" ne soit nécessaire.
- Contribuer à éviter les redondances, ce qui se traduit par moins de travail et une maintenabilité plus facile.

3.2 Produits d'activités basés sur le langage naturel (L2)

Depuis le début de l'IE systématique, les exigences exprimées en langage naturel ont été un moyen essentiel de spécifier les exigences dans la pratique.

Les produits d'activités basés sur le langage naturel présentent des avantages majeurs :

- Le langage naturel sans contrainte est extrêmement expressif et flexible.
- Presque toutes les exigences imaginables, sous quelque aspect que ce soit, peuvent être exprimées en langage naturel.
- Le langage naturel est utilisé dans la vie quotidienne et enseigné à l'école, aucune formation spécifique n'est donc nécessaire pour lire et comprendre les textes en langage naturel.

La contrepartie de ces avantages est que des textes rédigés en langage naturel peuvent souvent être interprétés de différentes manières, ce qui constitue un problème lors de la

spécification des exigences. En outre, la détection des ambiguïtés, des omissions et des incohérences dans ces textes est difficile et coûteuse.

La rédaction de bonnes exigences en langage naturel peut être soutenue par :

- Rédiger des phrases courtes et bien structurées.
- Définir et utiliser de manière cohérente une terminologie uniforme (3.5).
- Éviter les termes et phrases vagues ou ambigus.
- Connaître les pièges de la rédaction technique énumérés ci-dessous.

Lors de la rédaction de documents techniques en langage naturel, il existe des pièges bien connus qu'il convient d'éviter ou d'utiliser avec précaution [GoRu2003].

Les choses à éviter :

- Descriptions incomplètes
- Noms non spécifiques
- Conditions incomplètes
- Comparaisons incomplètes

Des choses à utiliser avec précaution :

- Voix passive
- Quantificateurs universels (tels que "tout" ou "jamais")
- Nominalisations (c'est-à-dire les noms dérivés d'un verbe, par exemple "authentification")

3.3 Produits d'activités basés sur des templates (L3)

Les produits d'activités basés sur des modèles sont utilisés pour pallier certaines des lacunes des produits d'activités basés sur le langage naturel en fournissant des structures prédéfinies pour les exigences.

- Les *gabarits de phrases* fournissent une structure syntaxique prédéfinie pour une phrase qui exprime une exigence, en particulier une exigence individuelle ou une User Story.
- Les *modèles de formulaires* fournissent un ensemble de champs prédéfinis dans un formulaire à remplir, par exemple pour rédiger un cas d'utilisation ou une exigence de qualité mesurable.
- Les *modèles de documents* fournissent une structure prédéfinie pour un document d'exigences.

Divers modèles ont été décrits dans la littérature. [ISO29148, [MWHN2009], et [Rupp2014] fournissent des gabarits de phrases pour les exigences individuelles. [Cohn2004] définit un gabarit de phrase largement utilisé pour les User Stories et [Cock2001]décrit des modèles de formulaires pour les cas d'utilisation. [Laue2002] a proposé un modèle de description de tâches. [ISO29148 et **Error! Reference source not found.**fournissent des modèles de documents pour l'ensemble des spécifications. En outre, un client peut prescrire l'utilisation de modèles qui lui sont spécifiques dans un projet.

Avantages des produits d'activités basés sur des modèles :

- Fournir une structure claire et réutilisable
- Aider à saisir les informations les plus pertinentes
- Uniformiser les exigences et les spécifications des exigences
- Améliorer la qualité globale des exigences et des spécifications des exigences

Inconvénients et pièges des produits d'activités basés sur des modèles :

- Les gens se concentrent souvent sur le fait de remplir complètement le modèle plutôt que sur le contenu.
- Les aspects qui ne sont pas inclus dans le modèle sont plus susceptibles d'être omis.

3.4 Produits d'activités basés sur des modèles (L3)

Les exigences représentées en langage naturel ont des limites[Davi1993], notamment en ce qui concerne l'obtention d'une vue d'ensemble sur un ensemble d'exigences et la compréhension des relations entre les exigences. La modélisation des exigences traite ces limitations.

3.4.1 Le rôle des modèles dans l'ingénierie des exigences (L2)

Un *modèle* est une représentation abstraite d'une partie de la réalité existante ou d'une partie de la réalité à créer. La notion de réalité comprend tout ensemble d'éléments, de phénomènes ou de concepts concevables, y compris d'autres modèles. Par rapport à un modèle, la partie modélisée de la réalité est appelée l'*original*. On peut citer comme exemple de modèles ne relevant pas du domaine du génie logiciel, les modèles d'information sur les bâtiments (BIM) [ISO19650], qui modélisent les éléments nécessaires à la planification, à la construction et à la gestion des bâtiments et autres éléments de construction.

En IE, les modèles aident à comprendre les relations et les interconnexions entre les exigences et donnent un aperçu d'un ensemble d'exigences. On y parvient principalement en se concentrant sur certains aspects – par exemple, le comportement – tout en faisant abstraction de tous les autres aspects. L'utilisation d'une notation graphique pour un modèle permet également d'obtenir une vue d'ensemble. Cependant, les modèles peuvent également être représentés de manière non graphique, par exemple avec des tableaux.

Les modèles d'exigences présentent des avantages par rapport aux exigences représentées en langage naturel :

- Les relations et les interconnexions entre les exigences sont plus faciles à comprendre avec des modèles graphiques qu'en les spécifiant en langage naturel.
- Se concentrer sur un seul aspect réduit la charge cognitive pour comprendre les exigences modélisées.
- Les langages de modélisation des exigences ont une syntaxe restreinte qui réduit les ambiguïtés et les omissions possibles.

Les modèles ont également des limites :

- Il est difficile de maintenir la cohérence entre des modèles qui se concentrent sur différents aspects.
- Les informations provenant de différents modèles doivent être intégrées pour une compréhension causale.
- Les modèles se concentrent principalement sur les exigences fonctionnelles ; la plupart des exigences qualité et des contraintes ne peuvent être exprimées dans les modèles avec un effort raisonnable.
- La syntaxe restreinte d'un langage de modélisation graphique implique que toutes les informations pertinentes ne peuvent pas être exprimées dans un modèle.

C'est pourquoi les modèles d'exigences et les exigences en langage naturel sont souvent combinés.

En IE, les modèles peuvent être utilisés pour :

- *Spécifier* les exigences (principalement fonctionnelles) en partie ou même complètement, comme moyen de remplacer les exigences représentées textuellement.
- *Décomposer* une réalité complexe en aspects bien définis et complémentaires, chaque aspect étant représenté par un modèle spécifique.
- *Paraphraser* des exigences représentées textuellement afin de les rendre plus faciles à comprendre, notamment en ce qui concerne les relations entre elles.
- *Valider* des exigences représentées textuellement dans le but de découvrir des omissions, ambiguïtés et incohérences.

Les *langages de modélisation* sont utilisés pour exprimer des modèles. Plusieurs langages de modélisation, par exemple UML [OMG2017]ou BPMN[OMG2013], ont été normalisés. Lorsque les exigences sont spécifiées dans un langage de modélisation non normalisé, une légende est requise pour expliquer la syntaxe et la sémantique du langage de modélisation utilisé.

Il existe de nombreux types de modèles qui peuvent être utilisés en IE. Un ingénieur des exigences doit comprendre quel type de modèle est le mieux adapté pour spécifier les exigences dans une situation donnée.

Dans une première phase, l'ingénieur des exigences commence souvent par modéliser le contexte (3.4.2) ou les objectifs du système prévu.

3.4.2 Modélisation du contexte (L2)

Les modèles qui se concentrent sur l'aspect contextuel spécifient l'intégration structurelle d'un système dans son environnement et l'interaction entre un système et les acteurs dans le contexte du système.

Les *modèles de contexte* spécifient un système et les acteurs qui, dans le contexte du système, interagissent avec le système. Un modèle de contexte esquisse également les

interfaces entre un système et son contexte (par exemple, en termes d'informations échangées).

Les *diagrammes de contexte* sont utilisés comme langage de modélisation graphique pour exprimer les modèles de contexte. Il n'y a pas de notation standardisée pour les diagrammes de contexte. Des diagrammes de contexte issus d'une analyse structurée [DeMa1978] ou des diagrammes sur-mesure composés de boîtes et de lignes [Glin2019 peuvent être utilisés pour représenter des modèles de contexte.

Dans le langage de modélisation UML [OMG2017], les *diagrammes de cas d'utilisation* fournissent un moyen de modéliser un système et son contexte avec les cas d'utilisation du système et les acteurs présents dans le contexte du système qui interagissent avec le système par le biais de ces cas d'utilisation.

Les *cas d'utilisation* modélisent l'interaction dynamique entre un acteur dans le contexte du système et un système du point de vue de l'acteur. Les cas d'utilisation sont le plus souvent rédigés à l'aide de modèles de formulaires en langage naturel (3.3) ou en utilisant des diagrammes d'activité UML (3.4.4).

3.4.3 Modélisation de la structure et des données (L3)

Les modèles qui se concentrent sur les aspects liés à la structure et aux données spécifient des exigences relatives aux propriétés structurelles statiques d'un système ou d'un domaine.

Les modèles statiques de domaines spécifient les objets (métier) et leurs relations dans un domaine d'intérêt. Ils peuvent être exprimés à l'aide de diagrammes [OMG2017]de classes UML .

Les *modèles de classes* spécifient principalement les classes d'un système et leurs attributs et relations. Les classes représentent des entités tangibles et intangibles du monde réel que le système doit connaître pour remplir ses tâches. Les diagrammes de classes UML sont généralement utilisés comme langage de modélisation pour les modèles de classes.

3.4.4 Modélisation des Fonctions et des Flux (L3)

Les modèles qui se concentrent sur les aspects de fonction et de flux spécifient les exigences relatives à la séquence d'actions requises pour produire les résultats requis à partir d'intrants donnés ou les actions requises pour exécuter un processus (métier), y compris le flux de contrôle et de données entre les actions et qui est responsable de quelle action.

Les *modèles d'activité* sont utilisés pour spécifier les fonctions du système. Dans le langage de modélisation UML [OMG2017], les *diagrammes* d'activité sont utilisés pour exprimer les modèles d'activité. Ils fournissent des éléments pour la modélisation des actions et le flux de contrôle entre les actions. Les diagrammes d'activité peuvent également exprimer qui est responsable de quelle action. Des éléments de modélisation avancés (non couverts par le niveau de base du CPRE) fournissent les moyens de modéliser le flux de données.

Les *modèles de processus* sont utilisés pour décrire des processus métier ou des processus techniques. Ils peuvent être exprimés à l'aide de diagrammes d'activité UML ou de langages de modélisation de processus spécifiques tels que BPMN [OMG2013]. Au Niveau Fondation du CPRE, nous n'utilisons que des diagrammes d'activité UML pour la modélisation des processus.

3.4.5 Modélisation de l'Etat et du Comportement (L2)

Les modèles qui se concentrent sur l'état et le comportement spécifient des exigences relatives au comportement d'un système ou d'un composant du domaine .en termes de réactions, dépendantes de l'état, à des événements ou à la dynamique de l'interaction des composants.

Les *machines d'état* modélisent les événements qui déclenchent la transition d'un état à un autre et les actions qui doivent être effectuées lorsqu'une transition d'état a lieu. Les *diagrammes d'état* [Hare1988] sont des machines d'état avec des états qui sont décomposés hiérarchiquement et/ou orthogonalement. Les machines d'état, y compris les diagrammes d'état, peuvent être exprimées dans le langage de modélisation UML [OMG2017] avec des *diagrammes de machines d'état* (également appelés *diagrammes d'états*).

3.4.6 Autres Types de Modèles dans l'Ingénierie des Exigences (L1)

Au Niveau Fondation du CPRE, la compréhension et l'application des modèles sont limitées à certains types de modèles importants. D'autres types de modèles sont utilisés dans l'Ingénierie des Exigences. Au Niveau Fondation du CPRE, il suffit de les connaître et de savoir à quoi ils servent.

Les modèles d'objectifs représentent un ensemble d'objectifs, de sous-objectifs et les relations entre eux. Les modèles d'objectifs peuvent également inclure les tâches et les ressources nécessaires pour atteindre un objectif, les acteurs qui veulent atteindre un objectif et les obstacles qui empêchent la réalisation d'un objectif.

En SysML [OMG2019], *les diagrammes de définition de blocs* peuvent être adaptés pour représenter des diagrammes de contexte en utilisant des blocs stéréotypés pour le système et les acteurs. Les diagrammes de définition de blocs peuvent également être utilisés pour modéliser la structure d'un système en termes d'entités conceptuelles du système et de relations entre elles.

Les modèles de Stories de domaine peuvent être utilisés pour modéliser la fonction et le flux, en spécifiant des Stories visuelles sur la façon dont les acteurs interagissent avec des dispositifs, des artéfacts et d'autres éléments dans un domaine, en utilisant généralement des symboles spécifiques au domaine [HoSch2020]. Ils permettent de comprendre le domaine d'application dans lequel un système fonctionnera.

Les *diagrammes de séquence* UML sont un moyen courant pour spécifier l'interaction entre des objets.

3.5 Glossaires (L2)

Dans toute démarche d'IE impliquant plusieurs personnes, il existe un risque d'absence de compréhension commune de la terminologie, c'est-à-dire que certaines personnes interprètent les mêmes termes de manière différente. Pour atténuer ce problème, la compréhension commune des termes pertinents est consignée dans un glossaire.

Un *glossaire* est une collection centralisée de définitions pour : les termes spécifiques au contexte, les termes quotidiens ayant une signification particulière dans le contexte donné, les abréviations et les acronymes. Les synonymes (différents termes désignant la même chose) doivent être indiqués comme tels. Les homonymes (utilisation du même terme pour des choses différentes) doivent être évités ou marqués comme tels.

Les règles suivantes s'appliquent aux glossaires :

- Gérer le glossaire de manière centralisée.
- Maintenir le glossaire pendant toute la durée du développement du système.
- Désigner une personne ou un petit groupe qui est responsable du glossaire.
- Utiliser un style et une structure uniformes pour le glossaire.
- Impliquer les parties prenantes et rechercher un accord sur la terminologie.
- Rendre le glossaire accessible à toutes les personnes concernées.
- Rendre l'utilisation du glossaire obligatoire.
- Vérifier les produits d'activités par rapport à une utilisation correcte du glossaire.

3.6 Documents et Structures de Documentation des Exigences (L2)

Les documents de spécification des exigences (3.1.1) comprennent plusieurs produits d'activités de l'IE. Il est donc important d'organiser ces documents avec une structure bien définie, afin de créer un ensemble d'exigences cohérent et maintenable. Outre les exigences, un document d'exigences peut également contenir des informations et des explications supplémentaires – par exemple, un glossaire, les conditions d'acceptation, des informations sur le projet ou les caractéristiques de la mise en œuvre technique.

Les exigences peuvent également être organisées dans des structures de documentation autres que les documents classiques.

Les documents fréquemment utilisés sont :

- Spécification des exigences des parties prenantes
- Spécification des exigences des utilisateurs (un sous-ensemble d'une spécification des exigences des parties prenantes, couvrant uniquement les exigences des utilisateurs)
- Spécification des exigences système
- Spécification des exigences métier
- Document de vision

Des structures de documentation alternatives fréquemment utilisées sont :

- Product backlog
- Sprint backlog
- Story map

Le choix d'une structure de documentation et l'organisation interne de la structure choisie dépendent :

- Du processus de développement choisi (5)
- Du type et du domaine de développement
- Du contrat (un client peut prescrire l'utilisation d'une structure de documentation donnée)
- De la taille du document

Les modèles de documents peuvent aider à structurer une spécification des exigences Des modèles sont disponibles dans la littérature [Vole2020]**Error! Reference source not found.**et dans les normes[ISO29148. Les modèles peuvent également être réutilisés à partir de projets antérieurs similaires ou peuvent être imposés par un client. Une organisation peut également décider de créer un modèle en tant que norme interne.

3.7 Prototypes en ingénierie des exigences (L1)

Dans le domaine de l'IE, les *prototypes* sont un moyen de spécifier des exigences par l'exemple et de les valider. Les prototypes peuvent notamment être utilisés si les parties prenantes concernées ne souhaitent pas rédiger et revoir des produits d'activités basés sur le langage naturel, sur des templates ou sur des modèles.

Les [LiSZ1994]*prototypes exploratoires* sont utilisés pour apporter une compréhension commune, clarifier des exigences ou valider des exigences à différents niveaux de fidélité. Ils sont jetés après usage.

- Les *wireframes* sont des prototypes basse-fidélité construits avec des matériaux simples ou des outils d'esquisses qui servent principalement à discuter et à valider des idées de conception et les concepts d'interface utilisateur.
- Les *maquettes* sont des prototypes moyenne-fidélité. Lors de la spécification de systèmes numériques ils utilisent des écrans et des flux de clics réels, mais sans réelle fonctionnalité. Elles servent principalement à spécifier et à valider les interfaces utilisateurs.
- Les *prototypes natifs* sont des prototypes haute-fidélité qui implémentent des parties critiques d'un système dans une mesure telle que les parties prenantes peuvent utiliser le prototype pour voir si la partie prototypée du système fonctionnera et se comportera comme prévu.

Selon le degré de fidélité, les prototypes exploratoires peuvent être un produit d'activités coûteux, de sorte qu'il y a toujours un compromis entre le coût et la valeur acquise.

Les [LiSZ1994]*prototypes évolutifs* sont des systèmes pilotes qui forment le noyau d'un système à développer. Le système final évolue en étendant et en améliorant progressivement le système pilote en plusieurs itérations.

3.8 Critères de qualité pour les produits d'activités et les exigences (L1)

Une exigence doit répondre à des critères de qualité spécifiques pour être considérée comme une bonne exigence. Dans un IE moderne avec des approches axées sur la valeur (Principe 1 dans 2), le degré de réalisation d'un critère de qualité doit correspondre à la valeur créée par cette exigence. Cela signifie que les exigences n'ont pas à adhérer complètement à tous les critères de qualité – mais plus la valeur d'une exigence est élevée, plus les critères de qualité sont pertinents, pour réduire le risque d'échec.

Adéquation et compréhensibilité sont les critères de qualité les plus importants pour les exigences uniques. Sans elles, une exigence est inutile, voire nuisible, indépendamment de la satisfaction de tous les autres critères.

Critères de qualité recommandés pour les *exigences uniques* :

- Adéquate (décrit les besoins réels et convenus des parties prenantes)
- Nécessaire
- Non ambiguë
- Complète (autonome)
- Compréhensible.
- Vérifiable

Comme décrit dans 3.1.1, les exigences sont généralement enregistrées dans divers produits d'activités qui couvrent des exigences uniques ou multiples. Les critères de qualité ci-dessus doivent être utilisés pour créer des exigences uniques de grande qualité dans un produit d'activités. Pour les produits d'activités qui couvrent plus d'une exigence, les critères de qualité suivants doivent être pris en compte en plus.

Critères de qualité recommandés pour les produits d'activités couvrant *plusieurs exigences* :

- Cohérent
- Non redondant
- Complet (aucune exigence connue et pertinente n'a été omise)
- Modifiable
- Traçable
- Conforme

4 Pratiques pour l'élaboration des exigences (L3)

Objectif : Comprendre l'utilisation des pratiques pour identifier les sources d'exigences, pour élucider des exigences, pour identifier et résoudre les conflits, et pour valider les exigences

Durée : 4 heures 30 minutes

Termes : source d'exigences, limite du système, contexte du système, élucidation des exigences, validation des exigences, partie prenante, modèle de Kano, conflit, résolution de conflit

Objectifs pédagogiques

OE 4.1.1 Déterminer les limites du système pour se concentrer sur les exigences pertinentes (L3)
OE 4.1.2 Se souvenir des sources pertinentes pour le système à créer (L1)
OE 4.1.3 Identifier les parties prenantes et rédiger une liste des parties prenantes (L3)
OE 4.1.4 Comprendre les bénéfices de la gestion des parties prenantes (L2)
OE 4.2.1 Comprendre comment le modèle Kano peut contribuer à élucider les bonnes exigences (L2)
OE 4.2.2 Comprendre la différence entre les techniques de collecte, et les techniques de conception et de génération d'idées (L2)
OE 4.2.3 Comprendre comment choisir une technique d'élucidation appropriée pour une situation donnée (L2)
OE 4.3.1 Se souvenir des différents types de conflits (L1)
OE 4.3.2 Comprendre quelles activités sont nécessaires pour résoudre les conflits (L2)
OE 4.3.3 Comprendre comment appliquer les techniques appropriées de résolution des conflits (L2)
OE 4.4.1 Comprendre pourquoi les documents d'exigences doivent être validés (L2)
OE 4.4.2 Se souvenir des quatre aspects importants pour la validation des exigences (L1)
OE 4.4.3 Comprendre comment appliquer les techniques appropriées pour la validation des exigences (L2)

4.1 Sources des exigences (L3)

La qualité et l'exhaustivité des exigences dépendent en grande partie des sources d'exigences impliquées. L'absence d'une source pertinente entraînera une compréhension incomplète des exigences ou des exigences incomplètes. L'identification des sources d'exigences est un processus itératif et récursif qui nécessite une reconsidération constante.

Le principe 3 du chapitre 2 souligne la nécessité d'une compréhension commune (explicite et implicite) entre et parmi toutes les parties concernées. Comprendre le contexte du système à développer dans un certain domaine d'application est un prérequis pour pouvoir identifier les sources d'exigences pertinentes. La zone située entre la limite du système et la limite du contexte est appelée contexte (du système) (Principe 4 dans 2). Le contexte (du système) est nécessaire pour comprendre la nature des exigences à développer et donc pour identifier les sources originales des exigences.

Les sources d'exigences sont classées en trois types :

- Parties prenantes
- Documents
- Systèmes

Les parties prenantes d'un système (voir [Glin2025] pour une définition ; voir aussi Principe 2dans 2) sont la principale source d'exigences. Les rôles typiques des parties prenantes comprennent [BiSp2003:

- Utilisateurs (également appelés utilisateurs finaux)
- Sponsors
- Managers
- Développeurs
- Autorités
- Clients

En outre, les personnes ou les organisations qui sont impactées par un système doivent être considérées comme des parties prenantes (indirectes).

L'identification systématique des parties prenantes devrait avoir lieu au début d'une opportunité de développement et les résultats devraient être gérés tout au long du développement comme une activité continue. Cela inclut l'identification des rôles des parties prenantes et des personnes qui occupent ces rôles.

Pour tous les systèmes avec une interface utilisateur, les *utilisateurs finaux* du système constituent un groupe de parties prenantes qui présente un intérêt particulier pour l'ingénieur des exigences. Les utilisateurs finaux doivent être associés dans des groupes (par exemple, en fonction de rôles, de tâches ou de responsabilités similaires).

Lorsque les utilisateurs finaux peuvent être identifiés individuellement, des représentants de chaque groupe doivent être choisis. Sinon, des personas peuvent être définis pour représenter les groupes d'utilisateurs finaux concernés [Coop2004].

Les sources potentielles d'identification des parties prenantes pertinentes sont les suivantes:

- Checklists de groupes et rôles typiques des parties prenantes
- Structures organisationnelles
- Documentation des processus métier
- Rapports de marché
- Premières parties prenantes pour identifier des parties prenantes supplémentaires

Les parties prenantes doivent être documentées dans une liste actualisée des parties prenantes avec (au moins) les informations suivantes :

- Nom
- Fonction (rôle)
- Coordonnées et informations personnelles
- Disponibilité temporelle et géographique tout au long du projet
- Pertinence
- Domaine et étendue de l'expertise

- Objectifs et intérêts par rapport au projet

Des problèmes avec les parties prenantes peuvent survenir si les droits et les obligations d'une partie prenante ne sont pas clairs ou si les besoins de la partie prenante ne sont pas suffisamment pris en compte. La gestion [Bour2009] des relations avec les parties prenantes est un moyen efficace de contrer les problèmes avec les parties prenantes.

Dans la plupart des contextes de système, davantage de sources sont disponibles. Elles doivent également être prises en compte pour la réussite d'un nouveau système, car la plupart des parties prenantes ne parlent pas de ce qui est évident : leurs exigences "subconscientes" (4.2).

Les sources supplémentaires d'exigences comprennent :

- Systèmes existants et anciens
- Documents de processus
- Documents juridiques ou réglementaires
- Réglementations spécifiques à l'entreprise
- Informations (marketing) sur les futurs utilisateurs potentiels

Une autre source d'exigences peut être trouvée en examinant des situations similaires dans des domaines complètement différents.

4.2 Élucidation des exigences (L2)

Dans le cadre de l'élucidation, il incombe à l'ingénieur des exigences de comprendre les souhaits et les besoins des parties prenantes tout en s'assurant que les exigences de toutes les sources pertinentes ont été recueillies, en appliquant les techniques appropriées pour les élucider. Un point important dans l'élucidation est de transformer les demandes, souhaits et attentes implicites en exigences explicites.

Pour élucider les exigences, il est essentiel de connaître la nature et l'importance de chaque exigence. Celles-ci peuvent changer d'un projet à l'autre et aussi au fil du temps. Le modèle de Kano [KaeA1984] classe les exigences en trois catégories pertinentes :

- Excitation (synonymes : facteurs de ravissement, exigences inconscientes)
- Performance (synonymes : facteurs de performance, exigences conscientes)
- Basique (synonymes : facteurs de base, facteurs d'insatisfaction, exigences subconscientes)

Il existe de nombreuses techniques différentes pour élucider ces catégories d'exigences. Nous faisons une distinction entre :

- Techniques de collecte
- Techniques de conception et de génération d'idées

Les *techniques de collecte* sont des techniques établies pour l'élucidation des exigences [BaCC2015] qui permettent d'identifier des facteurs de performance et des facteurs basiques en enquêtant auprès de différentes sources.

On peut distinguer quatre grandes catégories :

- Techniques d'interview
- Techniques de collaboration
- Techniques d'observation
- Techniques basées sur des artefacts

Les *techniques de conception et de génération d'idées* sont destinées à stimuler la créativité pendant l'élucidation des exigences. Elles visent à créer des idées pour résoudre un problème et à explorer des idées de conception [Kuma2013]. Cela peut conduire à des exigences nouvelles ou innovantes qui sont souvent des facteurs d'enthousiasme. Des exemples populaires de ces techniques sont le brainstorming [Osbo1979], les analogies, le prototypage (par exemple, les maquettes), les scénarios et les story-boards.

Un concept plus large lié à la conception et à la génération d'idées est le Design Thinking. Il existe différentes approches, telles que le *d.school* [Sdsc2012] et le *Designing for Growth* [LiOg2011], qui offrent un large éventail de techniques pouvant être utilisées pour l'élucidation des exigences.

Les techniques d'élucidation doivent pouvoir détecter toutes sortes d'exigences – exigences fonctionnelles, exigences qualité et aussi des contraintes. Dans la pratique, les exigences de qualité et les contraintes reçoivent souvent moins d'attention.

Pour élucider des *exigences de qualité*, un modèle de qualité tel que la norme [ISO25010]ISO 25010 doit être utilisé comme checklist. Ce modèle peut également être utile pour quantifier les exigences.

Les *contraintes* peuvent être trouvées en considérant les restrictions possibles de l'espace de solution potentiel – par exemple, les questions techniques, juridiques, organisationnelles, culturelles ou environnementales.

Choisir les bonnes techniques d'élucidation est une compétence clé essentielle qui dépend de nombreux facteurs différents, tels que

- Type de système
- Modèle de cycle de vie du développement logiciel
- Personnes concernées
- Structure organisationnelle

Les meilleurs résultats sont généralement obtenus par une combinaison de différentes techniques d'élucidation. [CaDJ2014 représentent une approche systématique pour la sélection des techniques.

4.3 Résoudre les Conflits concernant les Exigences (L2)

Les techniques d'élucidation ne garantissent pas à elles seules que l'ensemble des exigences qui en résultent est cohérent, complet, conforme, etc. (3.8). Pour l'ensemble final, cependant, toutes les parties prenantes doivent comprendre et accepter toutes les exigences qui les concernent. Si certaines parties prenantes ne sont pas d'accord, cette situation doit être reconnue comme un conflit qui doit être résolu en conséquence. Les

techniques de résolution de conflits appropriées doivent être choisies en fonction du type de conflit et des informations contextuelles. Cela nécessite une compréhension approfondie de la nature du conflit d'exigences et de l'attitude des parties prenantes concernées.

Les tâches pour identifier et résoudre les conflits sont :

- Identification du conflit
- Analyse du conflit
- Résolution des conflits
- Documentation de la résolution des conflits (décisions prises)

Il est utile de distinguer les différents types de conflits [Moor2014]. Les types de conflits suivants requièrent souvent l'attention de l'ingénieur des exigences :

- Conflit de sujets
- Conflit de données
- Conflit d'intérêt
- Conflit de valeur
- Conflit relationnel
- Conflit structurel

Pour résoudre les conflits avec succès, des techniques communes peuvent être appliquées :

- Accord
- Compromis
- Vote
- Le chef a toujours raison
- Définition de variantes

En outre, il existe plusieurs techniques auxiliaires, par exemple :

- Considérer tous les faits (Consider-All-Facts) ;
- Points forts/Points faibles (Plus-Minus-Interesting) ;
- Matrice de décision.

4.4 Validation des exigences (L2)

La validation des exigences est une étape importante pour la réussite du système (Principe 6 dans 2). Le fait de garantir la qualité des exigences en amont réduira le gaspillage d'efforts en aval. La validation des exigences consiste à vérifier la qualité des produits d'activités en cours et des exigences individuelles qu'ils contiennent (voir 3.8 pour plus de détails).

Les aspects importants à prendre en compte dans la validation des exigences sont les suivants

- Implication des bonnes parties prenantes
- Séparer l'identification et la correction des défauts
- Valider selon des points de vue différents
- Répéter la validation

Il existe plusieurs techniques de validation (par exemple[GiGr1993, [OleA2018]). Ces techniques de validation sont souvent classées en :

- Les *techniques de revue*, notamment :
  - Relectures techniques
  - Inspections
- T*echniques exploratoires*, par exemple :
  - Prototypage
  - Tests alpha et bêta
  - Test A/B [KoTh2017]
  - Construire un produit minimum viable (MVP)
- *Développement d'échantillons*

Ces techniques diffèrent par leur formalité et leur effort. La technique à choisir dépend de facteurs tels que le modèle de cycle de vie de développement logiciel, la maturité du processus de développement, la complexité et le niveau de risque du système, les exigences légales ou réglementaires éventuelles et/ou la nécessité d'un audit.

5 Processus et Structure de Travail (L3)

Objectif : Expliquer les concepts d'un processus d'IE et appliquer les configurations de processus appropriées

Durée : 1 heure 15 minutes

Termes : processus, processus IE

Objectifs pédagogiques

OE 5.1.1 Connaître les facteurs importants qui influencent un processus d'IE (L1)
OE 5.1.2 Comprendre comment et pourquoi ces facteurs influencent (L2)
OE 5.2.1 Comprendre les facettes à prendre en compte pour la configuration d'un processus d'IE (L2)
OE 5.3.1 Connaître les configurations typiques des processus d'IE (L1)
OE 5.3.2 Comprendre les étapes de configuration d'un processus d'IE (L2)
OE 5.3.3 Pour les réglages simples du système et du développement, sélectionner et appliquer une configuration du processus d'IE appropriée (L3)

Un processus est nécessaire pour façonner et structurer le travail d'IE à effectuer dans un contexte donné. Comme il n'existe pas de processus d'IE passe-partout (0), un processus d'IE sur mesure doit être configuré pour s'adapter au contexte du développement et du système.

Le processus d'IE façonne le flux d'informations et le modèle de communication entre les différents participants (par exemple, les clients, les utilisateurs, les ingénieurs des exigences, les développeurs, les testeurs) et définit également les produits d'activités à utiliser ou à produire. Ainsi, le processus d'IE fournit le cadre permettant d'élucider, de documenter, de valider et de gérer les exigences.

5.1 Facteurs d'influence (L2)

De nombreux facteurs influencent la configuration d'un processus d'IE. Les principaux facteurs sont les suivants :

- Adaptation du processus global : le processus d'IE doit s'adapter au processus global de développement du système.
- Contexte du développement
- Capacité et disponibilité des parties prenantes
- Compréhension commune
- Complexité et criticité du système à développer
- Contraintes
- Temps et budget disponibles
- Volatilité des exigences
- Expérience des ingénieurs des exigences

Une analyse des facteurs d'influence fournit des informations sur la manière de configurer le processus d'IE. Les facteurs d'influence limitent également l'espace des configurations possibles du processus. Par exemple, lorsque les parties prenantes ne sont disponibles qu'au

début du projet, on ne peut pas choisir un processus qui s'appuie sur des réactions continues des parties prenantes.

5.2 Facettes du processus d'ingénierie des exigences (L2)

Trois aspects décisifs doivent être pris en compte lors de la configuration d'un processus [Glin2019d'IE .

Facette temporelle : linéaire versus itérative

Dans un processus linéaire, les exigences sont spécifiées à l'avance dans une seule phase du processus. Dans un processus itératif, les exigences sont spécifiées de façon incrémentale, en commençant par les objectifs généraux et certaines exigences initiales, puis en ajoutant ou en modifiant des exigences à chaque itération.

Critères de choix d'un processus d'IE *linéaire* :

- Le processus de développement du système est planifié et essentiellement linéaire.
- Les parties prenantes connaissent leurs exigences et peuvent les spécifier dès le départ.
- Une spécification des exigences complète est nécessaire comme base contractuelle pour l'externalisation de la conception et de l'implémentation du système.
- Les autorités réglementaires exigent une spécification complète et formellement publiée des exigences à un stade précoce du développement.

Critères de choix d'un processus d'IE *itératif* :

- Le processus de développement du système est itératif et agile.
- De nombreuses exigences ne sont pas connues à l'avance, mais elles apparaîtront et évolueront au cours du développement du système.
- Les parties prenantes sont disponibles de sorte que de courtes boucles de rétroaction peuvent être établies comme moyen d'atténuer le risque de développer le mauvais système.
- La durée du développement permet de dépasser une ou deux itérations.
- La capacité à modifier facilement les exigences est importante.

Facette de l'objectif : prescriptif versus exploratoire

Dans un processus d'IE prescriptif, la spécification des exigences constitue un contrat : toutes les exigences sont obligatoires et doivent être mises en œuvre. Dans un processus d'IE exploratoire, seuls les objectifs sont connus a priori, tandis que les exigences concrètes doivent être explorées.

Critères de choix d'un processus d'IE *prescriptif* :

- Le client a besoin d'un contrat fixe pour le développement du système.
- La fonctionnalité et la portée ont la priorité sur le coût et les délais.
- Le développement du système spécifié peut faire l'objet d'un appel d'offres ou être sous-traité.

Critères de choix d'un processus d'IE *exploratoire* :

- Les parties prenantes n'ont au départ qu'une vague idée de leurs exigences.
- Les parties prenantes sont fortement impliquées et fournissent un retour d'information continu.
- Les délais et le coût ont la priorité sur la fonctionnalité et la portée.
- Il n'est pas clair a priori quelles exigences seront effectivement implémentées et dans quel ordre elles le seront.

Facette cible : spécifique au client versus orientée marché

Dans un processus d'IE spécifique au client, le système est commandé par un client et développé par un fournisseur. Dans un processus d'IE orienté marché, le système est développé comme un produit ou un service pour un marché, ciblant des segments d'utilisateurs spécifiques.

Critères de choix d'un processus d'IE *spécifique au client* :

- Le système sera principalement utilisé par l'organisation qui l'a commandé et qui paie pour son développement.
- Les parties prenantes importantes sont principalement associées à l'organisation du client.
- Des personnes individuelles peuvent être identifiées pour les rôles de parties prenantes.
- Le client veut une spécification des exigences qui puisse servir de contrat.

Critères de choix d'un processus d'IE *orienté marché* :

- L'organisation à l'initiative du développement a l'intention de vendre le système dans un segment de marché quelconque en tant que produit ou service.
- Les utilisateurs potentiels ne sont pas identifiables individuellement.
- Les ingénieurs des exigences doivent concevoir les exigences de manière à ce qu'elles correspondent aux besoins prévus des utilisateurs ciblés.
- Les Product Owners, les responsables du marketing, les concepteurs numériques et les architectes systèmes sont les principales parties prenantes.

Conseils et mises en garde

- Les critères présentés ci-dessus sont des *heuristiques* plutôt que des règles fixes. Par exemple, l'externalisation du développement du système se fait de préférence avec un processus d'IE prescriptif plutôt qu'exploratoire car le contrat entre le client et le fournisseur est généralement basé sur une spécification des exigences complète. Cependant, il est également possible de négocier un contrat d'externalisation basé sur un processus d'IE exploratoire.
- Les processus linéaires d'IE ne fonctionnent que si un processus sophistiqué de changement des exigences est en place.
- Les processus d'IE linéaires impliquent de longues boucles de rétroaction : Pour réduire le risque de développer le mauvais système, les exigences doivent être validées de manière intensive.

- Lors de la définition d'un processus d'IE, les termes "*linéaire*" et "*prescriptif*" sont souvent choisis ensemble.
- Les processus d'IE exploratoires sont généralement aussi des processus itératifs (et vice versa).
- Dans un processus orienté marché, le retour d'information des utilisateurs est le seul moyen de valider si le produit répondra réellement aux besoins du segment d'utilisateurs ciblé.
- La facette orientée marché ne se combine pas bien avec les facettes linéaires et prescriptives.

5.3 Configuration d'un processus d'ingénierie des exigences (L3)

Dans un contexte concret de développement de système, les responsables de l'IE doivent configurer le processus d'IE à appliquer. Sur la base d'une analyse des facteurs d'influence (5.1), il est possible d'utiliser une combinaison appropriée des facettes du processus décrites dans (5.2) [Glin2019. Trois combinaisons typiques sont décrites ci-dessous.

Processus d'IE participatif : Itératif, Exploratoire et Spécifique au client

Principal cas d'application :	le fournisseur et le client collaborent étroitement ; les parties prenantes sont fortement impliquées à la fois dans les processus d'IE et de développement
Produits d'activités typiques :	Product Backlog avec des User Stories et/ou des descriptions de tâches, prototypes
Flux d'information typique :	interaction continue entre les parties prenantes, les Product Owners, les ingénieurs des exigences et les développeurs ; peut inclure un retour d'information de la part des utilisateurs

Processus contractuel d'IE : Généralement linéaire (parfois itératif), prescriptif et spécifique au client

Principal cas d'application :	La spécification des exigences constitue la base contractuelle pour le développement d'un système par des personnes non impliquées dans la spécification et avec peu d'interaction avec les parties prenantes après la phase des exigences.
Produits d'activités typiques :	spécification classique des exigences du système, consistant en des exigences basées sur le langage naturel et des modèles
Flux d'information typique :	Principalement des parties prenantes vers les ingénieurs des exigences

Processus d'IE orienté produit : Itératif, Exploratoire & Orienté marché

Principal cas d'application :	Une organisation spécifie et développe un logiciel afin de le vendre ou de le distribuer en tant que produit ou service
Produits d'activités typiques :	Product Backlogs, prototypes
Flux d'information typique :	interaction entre le Product Owner, le marketing, les ingénieurs des exigences, les concepteurs numériques, les développeurs et (peut-être) un retour d'information rapide des clients/utilisateurs

Notez qu'il peut y avoir des contextes de système et de développement dans lesquels aucune des configurations susmentionnées ne convient. Par exemple, des contraintes réglementaires peuvent imposer l'utilisation d'un processus conforme à des normes données telles que la norme ISO/IEC/IEEE 29148 [ISO29148.

Lors de la configuration d'un processus d'IE, nous recommandons d'utiliser une procédure en cinq étapes :

1. Analyser les facteurs d'influence (EU 5.1)
2. Évaluer les critères de facette (EU 5.2)
3. Configurer le processus (EU 5.3)
4. Déterminer les produits d'activités (EU 3)
5. Sélectionner les pratiques appropriées

6 Pratiques de gestion des exigences (L2)

Objectif : Comprendre la nécessité et l'intérêt de la gestion des exigences

Durée : 2 heures

Termes : gestion des exigences, gestion du changement, traçabilité, attributs des exigences, cycle de vie des exigences, priorisation

Objectifs pédagogiques

OE 6.1.1 Savoir en quoi consiste la gestion des exigences et pourquoi elle est nécessaire (L1)
OE 6.2.1 Expliquer pourquoi les produits d'activités des exigences ont besoin d'un modèle d'état/cycle de vie (L2)
OE 6.3.1 Expliquer à quoi ressemble un concept de version des exigences dans une situation de projet donnée (L2)
OE 6.4.1 Connaître l'utilisation des configurations des exigences et des baselines (L1)
OE 6.5.1 Connaître l'objectif des attributs des exigences (L1)
OE 6.5.2 Expliquer à quoi ressemble un ensemble approprié d'attributs pour les exigences dans une situation de projet donnée (L2)
OE 6.5.3 Expliquer l'objectif des vues et nommer les différentes vues des exigences (L2)
OE 6.6.1 Nommer des raisons de mettre en place la traçabilité des exigences (L1)
OE 6.6.2 Résumer les différences entre la traçabilité implicite et explicite (L1)
OE 6.6.3 Savoir comment la traçabilité explicite peut être documentée (L1)
OE 6.7.1 Savoir comment gérer les changements dans les approches linéaires (basées sur un plan) et en Agile (L1)
OE 6.8.1 Connaître l'utilité de la priorisation et connaître des critères d'évaluation significatifs (L1)
OE 6.8.2 Nommer les étapes pour prioriser les exigences (L1)
OE 6.8.3 Nommer différentes catégories de techniques de priorisation (L1)

6.1 Qu'est-ce que la gestion des exigences ? (L1)

La gestion des exigences est le processus de gestion des exigences existantes enregistrées dans divers produits d'activités. Cela comprend en particulier l'enregistrement, les changements et la traçabilité des exigences [Glin2025]. La gestion des exigences peut se faire de différentes manières et à différents niveaux selon le processus de développement choisi et le contexte – voir, par exemple, [Leff2011], [Rupp2014], [WiBe2013]. Quelles que soient les circonstances, la tâche de la gestion des exigences consiste à maintenir les exigences de telle sorte que tous les rôles dans un projet puissent fonctionner de manière efficace et efficiente.

6.2 Gestion du cycle de vie (L2)

La gestion du cycle de vie fait référence au processus de suivi de tous les produits d'activités en fonction de leur statut dans leur cycle de vie. Chaque exigence documentée et chaque produit d'activités a son propre cycle de vie : il est créé, puis évalué et affiné avant d'être revu, retravaillé, consolidé, approuvé, etc. Pour permettre d'identifier quel produit d'activité est dans quel état, un modèle de cycle de vie est nécessaire, définissant chaque état du

cycle de vie et chaque transition d'état autorisés. Le statut réel d'un produit d'activités doit toujours être clair, y compris (généralement) l'historique de ses transitions.

6.3 Contrôle des versions (L2)

Le contrôle de version des exigences fait référence au processus de suivi de tous les produits d'activités au cours de leur évolution. Tout changement dans un produit d'activités doit être reflété par une nouvelle version. Le contrôle de version permet de retracer l'histoire d'un produit d'activités jusqu'à son origine et de restaurer un produit d'activités dans toute version antérieure. À cette fin, le contrôle de version nécessite la mise en place de trois mesures:

- Un numéro de version pour identifier de manière unique la version d'un produit d'activités.
- Un historique de ce qui a été changé.
- Un concept de stockage des produits d'activités.

Le contrôle de version doit être envisagé pour tous les produits d'activités [WiBe2013]. Un numéro de version se compose généralement d'au moins deux parties : la version et l'incrément.

6.4 Configurations et baselines (L1)

Une *configuration* d'exigences est un ensemble cohérent de produits d'activités qui contiennent des exigences. Chaque configuration est définie dans un but précis et contient au maximum une version de chaque produit d'activités [Glin2025]. Le but des configurations est, par exemple, d'examiner un ensemble de produits d'activités ou de faciliter une estimation de l'effort de développement.

Une *baseline* est une *configuration* stable et contrôlée de produits d'activités, utilisée pour la planification des versions ou d'autres jalons de livraison dans un projet[Glin2025].

Les configurations ont les propriétés suivantes :

1. Connexion logique
2. Cohérence
3. Caractère unique
4. Inchangeabilité
5. Base de la réinitialisation

6.5 Attributs et vues (L2)

Les *attributs* sont nécessaires pour documenter les métadonnées importantes pour un produit d'activités et sont généralement utilisés pour répondre à un certain nombre de questions importantes au cours du cycle de vie du projet ou du produit.

L'objectif d'assortir les exigences d'attributs est de permettre aux membres de l'équipe et aux autres parties prenantes d'obtenir les informations sur les exigences dont ils ont besoin à tout moment au cours du projet.

La définition d'ensemble d'attributs pertinent dépend des besoins d'information des différentes parties prenantes du projet. Les normes existantes, par exemple[ISO29148, donnent un aperçu des attributs les plus pertinents.

Les vues sont un aperçu de l'ensemble complet des exigences qui ne contient que le contenu qui est actuellement d'intérêt. D'un point de vue technique, une vue est une combinaison de paramètres de filtrage et de tri qui peuvent être mis à disposition ou réutilisés par d'autres participants en enregistrant la combinaison sélectionnée.

Nous distinguons trois types de vues :

- *Vues sélectives*
- *Vues projectives*
- *Vues agrégées*

Dans la plupart des cas, les vues des exigences sont des combinaisons de vues sélectives, projectives et agrégées pour la création de rapports.

6.6 Traçabilité (L1)

La traçabilité [GoFi1994est la capacité de relier une exigence *à son origine* (c'est-à-dire aux parties prenantes, aux documents, aux justifications, etc.) et *vers les produits d'activité ultérieurs (par exemple, les cas de test)*, ainsi qu'*aux autres exigences* dont elle dépend.

La traçabilité est un prérequis à la gestion des exigences et est souvent explicitement exigée par les normes, les lois et les règlementations. La mise en œuvre de la traçabilité signifie essentiellement le maintien de dépendances entre différents produits d'activités (3.1) à différents niveaux d'abstraction (3.1.2), niveaux de détail (3.1.3) et vers tous les prédécesseurs et successeurs pertinents pour des raisons d'analyse, de conformité et d'information.

La traçabilité peut être documentée *implicitement,* en structurant et en normalisant les produits d'activités*,* ou *explicitement,* en reliant de différentes façons les produits d'activités les uns aux autres sur la base de leurs identifiants uniques [HuJD2011]. Les formes de représentation les plus courantes sont les hyperliens, les références, les matrices, les tableaux ou les graphiques.

6.7 Gestion du changement (L1)

Les exigences ne sont pas statiques. Les changements d'exigences sont dus à de nombreuses raisons différentes et doivent être traités correctement, (Principe 7 en 2) par exemple en créant une *demande de modification* formelle ou en ajoutant un nouvel élément au Product Backlog.

La prise de décision, la planification et le contrôle de la mise en œuvre d'un changement dépendent de l'approche de développement et du moment où le changement se produit.

Dans une approche *linéaire*, la décision sur un changement est souvent prise par un comité de contrôle des changements (dans les projets) ou un comité consultatif des changements (en fonctionnement). Dans une approche plus *itérative*, le Product Owner inclut le changement dans le Product Backlog et priorise le nouvel élément en conséquence.

6.8 Priorisation (L1)

Toutes les exigences ne sont pas d'égale importance[Davi2005]. L'évaluation et la priorisation servent à déterminer les exigences les plus pertinentes pour la prochaine release ou le prochain incrément de produit.

L'*évaluation* des exigences est la base de leur priorisation, souvent déterminée par l'utilisation de multiples critères d'évaluation tels que la valeur métier, l'urgence, l'effort, les dépendances et autres.

La *priorité* d'une exigence décrit l'importance d'une exigence unique par rapport à d'autres exigences selon certains critères [Glin2025]. La *priorisation* elle-même est effectuée sur la base d'un seul critère ou de plusieurs critères ; cela dépend principalement de la technique de priorisation choisie.

Étapes pour la priorisation :

- Définir les principaux objectifs et les principales contraintes pour la priorisation
- Définir les critères d'évaluation souhaités
- Définir les parties prenantes qui doivent être impliquées
- Définir les exigences qui doivent être priorisées
- Choisir la technique de priorisation
- Établir les priorités

Les *techniques de priorisation* peuvent être classées en :

- Les techniques de priorisation *Ad-hoc*
- Les techniques de priorisation *Analytiques*

7 Support des outils (L2)

Objectif : donner un aperçu du rôle des outils d'IE et des aspects de la mise en œuvre

Durée : 30 minutes

Termes : outil, outil d'IE

Objectifs pédagogiques

OE 7.1.1 Connaître les différents types d'outils d'IE (L1)
OE 7.2.1 Expliquer ce qu'il faut prendre en compte lors de l'introduction d'outils d'IE (L2)

7.1 Outils dans l'ingénierie des exigences (L1)

Le processus d'IE peut être étayé par des outils qui soutiennent des tâches et des activités spécifiques. Comme le processus d'IE est plutôt individuel (5), les outils d'IE existants se concentrent souvent seulement sur certains aspects de l'IE et soutiennent rarement toutes les activités. Avant de choisir un outil, les ingénieurs des exigences doivent décider quelles tâches et activités du processus d'IE doivent être soutenues et comment. Nous faisons la différence entre les outils qui soutiennent :

- La gestion des exigences
  - La définition et l'enregistrement des attributs des exigences
  - La priorisation des exigences
  - La gestion des versions et des configurations
  - Le suivi et la traçabilité des exigences
  - La gestion des changements d'exigences
- Gestion du processus d'IE :
  - Mesurer et rendre compte du processus d'IE
  - Mesurer et rendre compte de la qualité des produits
  - Gérer le flux de travail de l'IE
- Documentation des connaissances sur les exigences :
  - Partager les exigences
  - Créer une compréhension commune des exigences
- Modélisation des exigences
- Collaboration dans l'IE
- Test/simulation des exigences

Les outils se présentent souvent sous la forme d'un mélange des caractéristiques mentionnées ci-dessus. Afin de garantir que toutes les tâches liées à l'IE soient couvertes de manière appropriée, différents outils peuvent être combinés.

Parfois, d'autres types d'outils, par exemple des outils de bureautique ou de suivi des incidents, sont utilisés pour documenter ou gérer les exigences. Ces outils ont des limitations et ne devraient être utilisés que lorsque le processus d'IE est sous contrôle et que les exigences sont alignées et assez stables.

7.2 Mise en place des Outils (L2)

La sélection d'un outil d'IE n'est pas différente de la sélection de n'importe quel autre outil . L'objectif, le contexte et les exigences doivent être décrits avant pour que la sélection puisse être réussie[Fugg1993.

Un outil approprié ne peut être recherché qu'une fois que les procédures et techniques d'IE appropriées ont été introduites. La mise en place d'un outil requiert des responsabilités et des procédures d'IE claires. Dans le processus de mise en place d'un outil d'IE, les aspects suivants sont pertinents :

- Prendre en compte tous les coûts du cycle de vie au-delà des coûts de licence
- Considérer les ressources nécessaires
- Utiliser des projets pilotes pour éviter les risques
- Evaluer l'outil selon des critères définis
- Apprendre aux employés à utiliser l'outil

Références

[AnPC1994] Annie I. Antón, W. Michael McCracken, Colin Potts: Goal Decomposition and Scenario Analysis in Business Process Reengineering. CAiSE (Conference on Advanced Information Systems Engineering), 1994, 94–104.

[BaCC2015] K. Baxter, C. Courage, K. Caine: Understanding Your Users: A Practical Guide to User Research Methods, 2nd edition. Morgan Kaufmann, Burlington, 2015.

[BiSp2003]K. Bittner, I. Spence : Use Case Modelling. Pearson Education, Boston, 2003.

[Bour2009] L. Bourne: Stakeholder Relationship Management: A Maturity Model for Organisational Implementation. Gower Publishing Ltd, Birlington, 2009.

[CaDJ2014]D. Carrizo, O. Dieste, N. Juristo: Systematizing requirements elicitation technique selection. Information and Software Technology 2014, 56(6) : 644–669.

[Cock2001] A. Cockburn: Writing Effective Use Cases. Addison-Wesley, Boston 2001.

[Cohn2004] M. Cohn: User Stories Applied – For Agile Software Development. Addison-Wesley, Boston, 2004.

[Coop2004] A. Cooper: The Inmates Are Running the Asylum: Why High-Tech Products Drive Us Crazy and How to Restore the Sanity. Que, Indianapolis, 2004.

[Davi2005] A. M. Davis: Just Enough Requirements Management – Where Software Development Meets Marketing. Dorset House Publishing, New York, 2005.

[Davi1993] A. M. Davis: Software Requirements – Objects, Functions, & States, 2nd edition, Prentice Hall, New Jersey, 1993.

[DeMa1978] T. DeMarco: Structured Analysis and System Specification. Yourdon Press, New York, 1978.

[Fugg1993] A. Fuggetta: A classification of CASE technology. IEEE Computer 1993, 26 (12) : 25–38.

[GiGr1993] T. Gilb, D. Graham: Software Inspection. Addison Wesley, Boston, 1993.

[Glin2019] M. Glinz : Ingénierie des exigences I. Course Notes, University of Zurich, 2019. https://www.ifi.uzh.ch/en/rerg/courses/archives/hs19/re-i.html#resources. Dernière visite Janvier 2026.

[Glin2025] M. Glinz: A Glossary of Requirements Engineering Terminology. Version 2.2. https://cpre.ireb.org/downloads-and-resources/downloads#cpre-glossary. Dernière visite Janvier 2026.

[GoFi1994] O. Gotel, A. Finkelstein: An Analysis of the Requirements Traceability Problem. 1st International Conference on Requirements Engineering, Colorado Springs, 1994. 94–101.

[GoRu2003] R. Goetz, C. Rupp: Psychotherapy for System Requirements. 2nd IEEE International Conference on Cognitive Informatics (ICCI'03), London, 2003. 75–80.

[GRL2020] Goal oriented Requirement Language. University of Toronto, Canada https://www.cs.toronto.edu/km/GRL. Dernière visite Janvier 2026.

[Hare1988] D. Harel: On Visual Formalisms. Communications of the ACM 1988, 31 (5): 514–530.

[HoSch2020] S. Hofer, H. Schwentner: Domain Storytelling — A Collaborative Modeling Method. Disponible auprès de Leanpub, http://leanpub.com/domainstorytelling. Dernière visite Janvier 2026.

[HuJD2011] E. Hull, K. Jackson, et J. Dick: Requirements Engineering. Springer, 3e édition, 2011.

[ISO29148] ISO/IEC/IEEE 29148 : Ingénierie des systèmes et des logiciels – Processus du cycle de vie – Ingénierie des exigences, Organisation internationale de normalisation, 2018.

[ISO19650] ISO 19650 : Organisation et numérisation des informations sur les bâtiments et les travaux de génie civil, y compris les informations sur la modélisation des bâtiments (BIM) – Gestion des informations à l'aide de la modélisation des informations sur les bâtiments – Partie 1 et 2, Organisation internationale de normalisation, 2018.

[ISO25010] ISO/IEC/IEEE25010:2011 : Exigences de qualité et évaluation des systèmes et des logiciels (SQuaRE) –– Modèles de qualité des systèmes et des logiciels. Organisation internationale de normalisation, Genève, 2011.

[Jack1995] M. A. Jackson: Software Requirements and Specifications: A Lexicon of Practice, Principles and Prejudices. Addison-Wesley, New York, 1995.

[Jack1995b] M Jackson: The World and the Machine. 17e conférence internationale sur le génie logiciel 1995 (ICSE 1995). 287–292.

[KaeA1984] N. Kano et al.: Attractive quality and must-be quality. Journal of the Japanese Society for Quality Control 1984, 14(2) : 39–48. (en japonais)

[KoTh2017] R. Kohavi, S. Thomke: The Surprising Power of Online Experiments – Getting the most out of A/B and other controlled tests. Harvard Business Review, Sept–Oct 2017: 74–82.

[Kuma2013] V. Kumar : 101 Design Methods – A Structured Approach for Driving Innovation in Your Organization. John Wiley & Sons, Hoboken, 2013.

[Laue2002] S. Lauesen: Software Requirements. Styles et techniques. Addison-Wesley, Harlow, 2002.

[Leff2011] D. Leffingwell : Agile Software Requirements, Lean Requirements Practices for Teams, Programs, and the Enterprise. Addison-Wesley, Boston, 2011.

[LiOg2011] J. Liedtka, T. Ogilvie: Designing for Growth: A Design Thinking Tool Kit For Managers. Columbia University Press, 2011.

[LiSZ1994] H. Lichter, M. Schneider-Hufschmidt, H. Zullighoven: Prototyping in Industrial Software Projects – Bridging the Gap Between Theory and Practice. IEEE Transactions on Software Engineering 1994, 20 (11) : 825–832.

[MFeA2019] D. Méndez Fernández, X. Franch, N. Seyff, M. Felderer, M. Glinz, M. Kalinowski, A. Volgelsang, S. Wagner, S. Bühne, K. Lauenroth: Do We Preach What We Practice? Investigating the Practical Relevance of Requirements Engineering Syllabi – The IREB Case. CIbSE 2019 : 476–487.

[Moor2014] C. W. Moore : The Mediation Process – Practical Strategies for Resolving Conflicts, 4e édition. John Wiley & Sons, Hoboken, 2014.

[MWHN2009] A. Mavin, P. Wilkinson, A. Harwood, et M. Novak: Easy Approach to Requirements Syntax (EARS). 17th IEEE International Requirements Engineering Conference (RE'09), Atlanta, Georgia, 2009. 317–322.

[OleA2018] K. Olsen et al : Certified Tester, Foundation Level Syllabus – Version 2018. International Software Testing Qualifications Board, 2018.

[OMG2013] Object Management Group: Business Process Model and Notation (BPMN), version 2.0.2. OMG document formal/2013-12-09 http://www.omg.org/spec/BPMN. Dernière visite Janvier 2026.

[OMG2017] Object Management Group: OMG Unified Modeling Language (OMG UML), version 2.5.1. OMG document formal/2017-12-05. https://www.omg.org/spec/UML/About-UML/. Dernière visite Janvier 2026.

[OMG2019] Object Management Group: OMG Systems Modeling Language (OMG SysML™), Version 1.6. OMG Document formal/19-11-01. https://www.omg.org/spec/SysML/. Dernière visite Janvier 2026.

[Osbo1979] A. F. Osborn: Applied Imagination, 3rd revised edition. Charles Scribner's Sons, New York, 1979.

[RoRo2013] S. Robertson et J. Robertson : Mastering the Requirements Process, 3e édition. Addison-Wesley, Boston, 2012.

[Rupp2014] C. Rupp: Requirements-Engineering und Management, 6. Auflage. Hanser, München, 2014. (en allemand).

[Sdsc2012] Stanford d.school: An Introduction to Design Thinking.. Institut de design Hasso Plattner, Stanford, 2012. https://web.stanford.edu/~mshanks/MichaelShanks/files/509554.pdf . Dernière visite Janvier 2026.

[vLam2009] Axel van Lamsweerde: Requirements Engineering: From System Goals to UML Models to Software Specifications. Chichester: John Wiley & Sons, 2009.

[Vole2020] Volere: Requirements Resources. https://www.volere.org. Dernière visite Janvier 2026.

[WiBe2013] K. Wiegers et J. Beatty: Software Requirements, 3rd edition. Microsoft Press, Redmond, 2013.