AI 時代の要求管理
ハンドブック

～経営・現場・開発がともに描く、問いから実現への道筋～

Version 1.0.0

独立行政法人
情報処理推進機構
Innovation Platform Agency, Japan

目次

はじめに.....6
本書の使い方.....9
想定読者.....9
主要用語の解説.....11
コラム：国際標準における「Requirements」と本書の「要求・要件」.....0
実プロジェクトでのテーラリング.....0
第 1 部　要求管理の基礎.....1
第 1 章　要望・要求・要件とは何か.....2
1.1 要求定義の出発点.....2
1.2 要望、要求、要件の違い.....2
1.3 木のブランコの例から学ぶ正確な要求定義の重要性.....4
1.4 所与の条件をシンプルにする.....5
1.5 要求定義が生む価値.....6
1.6 要求管理は職人芸から組織能力へ.....7
1.7 AI 時代に一層重要となる要求管理の観点.....8
第 2 章　要求工学の基本プロセス.....8
2.1 要求工学とは何か.....8
2.2 要求獲得.....10
2.3 要求分析.....11
2.4 要求仕様化.....13
2.5 要求評価.....14
2.6 要求管理.....16
第 3 章　要求管理を支える考え方.....18
3.1 要求管理は単独の方法論ではない.....18
3.2 デザイン思考との関係.....19
3.3 ミッションエンジニアリングとの関係.....20
3.4 要求工学との関係.....21
3.5 システムズエンジニアリングとの関係.....21
3.6 MBSE との関係.....22
3.7 アジャイルガバナンスとの関係.....22
3.8 各方法論の関係整理.....23
第 4 章　グローバル経営時代の要求管理.....24
4.1 経営・事業変革において要求管理が重要になる理由.....24
4.2 グローバルな取引を例に見る関係者間の共通理解.....26

4.3 迅速なサプライチェーン構築と要求管理 27
4.4 M&A における要求定義の役割 27
4.5 文化的背景や制度の違いへの対応 28
4.6 言語の違いを吸収する要求定義 29
4.7 グローバル要求定義で追加すべき観点 30
4.8 グローバル経営時代の要求管理の進め方 30
第 2 部 AI 時代に要求管理はどう変わるか 32
第 5 章 AI が変える要求管理の前提 33
5.1 変化速度が上がる時代の要求管理 33
5.2 AI システムは変化し続ける 34
5.3 AI システムにおける不確実性を要件として定義する 35
5.4 プロンプトエンジニアリングも要求工学の一部である 38
5.5 AI の社会的受容性と倫理的影響を要件として扱う 39
5.6 レジリエンスと継続的アシュアランスを要求として定義する 41
第 6 章 AI による要求管理支援 45
6.1 AI は要求管理の作業を支援する 45
6.2 AI によるヒアリング支援 46
6.3 AI によるドキュメンテーション支援 47
6.4 AI による過不足・矛盾の確認 47
6.5 AI による要求のモデル出力 48
6.6 AI による変更影響分析と再検証支援 49
6.7 AI 活用時の注意点 51
6.8 複数の AI エージェントの連携による要求管理 51
第 7 章 AI 時代の要求管理プロセス 52
7.1 AI 時代の要求管理プロセスの全体像 52
7.2 各工程の考え方 53
第 8 章 AI 時代の要求獲得・分析・仕様化 56
8.1 要求獲得の対象の拡大 56
8.2 マスの意見と「光る原石」を区別する 56
8.3 潜在要求を見つけるための問い 57
8.4 データ供給チェーンの透明性とトレーサビリティ 58
8.5 AI 時代の要求分析 62
8.6 AI 時代の要求仕様化 63
第 3 部 モデリングとトレーサビリティ 65
第 9 章 要求を可視化する 66
9.1 なぜ要求を可視化するのか 66

9.2 Human Readable から Machine Understandable へ 67
9.3 代表的なモデル・モデリング手法・構造化手法 68
9.4 建築や製造から学ぶ可視化 68
9.5 要求情報とモデルを接続する 69
9.6 モデリングを導入する現実的な進め方 70
9.7 形式手法による要求の厳密化と検証 71
第 10 章 ビルディングブロックやサービスによるシステムの実現 73
10.1 モデリングから実現手段へ 73
10.2 ビルディングブロックとは何か 74
10.3 機能ブロックの実現方法 75
10.4 ビルディングブロック化の利点 76
10.5 セキュリティ対策におけるビルディングブロック 76
10.6 ビルディングブロックを選定するための要件 77
10.7 ビルディングブロックと標準化・相互運用性 78
10.8 ビルディングブロック化と調達・契約 79
10.9 ビルディングブロック化の注意点 79
第 11 章 トレーサビリティと DevOps 型要求管理 80
11.1 なぜトレーサビリティが重要か 80
11.2 何をつなぐべきか 80
11.3 DevOps 型の要求管理 82
11.4 トレーサビリティマトリクス 82
11.5 変更管理と影響分析 83
11.6 ReqIF と OSLC による要求情報の交換・連携 85
第 4 部 ルール・制度・コンプライアンスと要求管理 89
第 12 章 外部要件としての法令・規制・標準 89
12.1 外部要件とは何か 89
12.2 「法令を遵守すること」では要件にならない 90
12.3 外部要件を社内要件へ変換する 91
第 13 章 規則・ガイドライン・マニュアルを要求定義に活かす 92
13.1 社内規則は「使われる要件」でなければならない 92
13.2 社内ルールを種類ごとに整理する 93
第 14 章 社内ルールの構造化と標準形式 96
14.1 文書を構造化する意義 96
14.2 Markdown による軽量な構造化 97
14.3 XML による構造化 98
14.4 Akoma Ntoso の考え方 99

14.5 ReqIF との接続 100
14.6 社内ルールを AI で活用する 100
第 5 部 AI 時代の要求管理プロセスの組織への定着 102
第 15 章 組織への導入と成熟度 102
15.1 要求管理を組織能力にする 102
15.2 役割分担 103
15.3 導入ステップ 104
15.4 成熟度モデル 105
15.5 調達・委託時の注意点 105
第 16 章 成果物一覧 106
16.1 要求定義書と要求管理台帳の位置づけ 106
16.2 AI 時代の要求管理での代表的な成果物 107
おわりに 109
参考文献 110

本書の内容に関して

- 本書を発行するにあたって、内容に誤りのないよう可能な限りの注意を払いましたが、本書の内容を適用した結果生じたこと、また、適用できなかった結果について、著者、発行人は一切の責任を負いませんので、ご了承ください。
- 本書の一部あるいは全部について、著者、発行人の許諾を得ずに無断で転載、複写複製、電子データ化することは禁じられています。
- 本書は、産業界及び学術界の有識者約 10 名によるレビューを踏まえて作成しています。一方、要求管理に唯一の正解はなく、技術や開発環境、社会的要請の変化に応じて、適切な考え方や実践も変化し得るため、本書は完成版ではなく、継続的に見直し・更新するリビングドキュメントとして位置付けます。そのため、本書の内容に関する意見や改善提案については次に示す連絡先までご連絡ください。今後の改訂に活用すべく、積極的に受け付けております。

独立行政法人 情報処理推進機構（IPA）
デジタル&AI システムズ・デザインセンター アーキテクチャ基盤設計部
システム・ソフトウェアエンジニアリンググループ
dadc-info@ipa.go.jp

商標

- 本書に記載する組織名、製品名などは、それぞれの組織の商標もしくは登録商標です。本書の文中においては、これらの表記において商標登録表示、その他の商標表示を省略しています。

はじめに

AI、データ、ソフトウェアが社会や産業の基盤となる時代において、要求管理（要求・要件定義を含む）の重要性はこれまで以上に高まっている。従来、要求管理（特に、要求定義）、情報システム開発の上流工程において、利用者や業務部門の要望を整理し、システムが実現すべき機能や性能を明確にする作業として捉えられてきた。しかし、AI の利用が広がり、社会やビジネスの変化速度が高まる中で、要求管理は単なるシステム開発の準備作業ではなく、企業や行政が変化に対応し、価値を継続的に生み出すための基盤的な活動になりつつある。

AI 時代のシステムは、あらかじめ決められた機能をそのまま実行するのみではなくなっていく。データを利用し、状況を判断し、利用者と対話し、継続的に更新される。業務プロセス、データ、AI モデル、ソフトウェア、法令、標準、社内規則、マニュアルは相互に関係し、一つの変更が広い範囲に影響を及ぼす。例えば、AI モデルの更新は、データ品質、説明責任、監査、利用者への通知、社内規則、契約、法令遵守に影響する。逆に、法令や標準の変更は、業務フロー、データ項目、AI の利用条件、システム仕様、運用手順の見直しを必要とする。このような環境では、要求を一度作って終わりにすることは望ましくない。

また、AI は要求定義を大きく支援する可能性を持つ。大量の問い合わせ、クレーム、事故報告、議事録、マニュアル、ログ、既存仕様書を分析し、要求候補を抽出し、分類し、要約することができる。対話形式で利用者や業務担当者から不足している情報を聞き出し、業務フローやデータモデルのたたき台を作成することもできる。既存システムやコードから構造を推定し、リバースエンジニアリングを支援することも可能である。

しかし、AI が要求定義を完全に代替するわけではない。AI は過去のデータや一般的な実務に基づいて回答するため、まだ普及していない新しい方法、制度の抜け道ではなく本質的な目的、現場に埋もれた潜在要求、少数だが重要なリスクを見落とすことがある。AI が出力した要求が妥当かどうかを判断するためには、人間側に要求工学、業務分析、データ、AI、法令・標準、リスク管理に関する基本的な理解が必要である。そのため、AI 時代には、要求定義を AI に任せるのではなく、AI を使いこなすために要求定義の力がより重要になると考えられる。

要求管理の出発点は、要望、要求、要件を区別することである。要望は、利用者や関係者の望み、困りごと、期待であり、しばしば漠然としている。要求は、その要望をもとに、実現したいこととして整理したものである。要件は、その要求を、実現しなければならない事項として具体化したものである。AI 時代の要求管理では、この変換過程を属人的な経験や暗黙知に任せるのではなく、関係者が共通理解でき、機械も利用できる形で構造化することが望ましい。

同時に、要求管理は、要望や既存業務をそのまま足し合わせる作業ではない。既存の業務、社内規則、マニュアル、システム制約、データ制約は、所与の条件として与えられることが多い。しかし、これらをそのまま AI やソフトウェアに実装してしまうと、複雑で変更しにくく、説明もしにくい仕組みになってしまう。AI 時代の要求管理では、与えられた条件を目的に照らして見直し、不要な手順、重複した入力、過剰な承認、曖昧な判断基準を整理し、可能な限りシンプルな構造にすることが重要になると考えられる。複雑

なものをそのまま自動化するのではなく、業務、ルール、データ、システムを見直し、単純化したうえで、AI やソフトウェアが扱いやすい形に再構成することが求められる。また、AI 時代であっても、すべての業務や機能に AI 適用が適切とは限らない。AI の利用自体を目的とせず、ルールベースの処理や既存システム等も含めて実現手段を比較し、AI を利用する必要性と適用範囲を見極めることが重要である。
そのためには、要求を単なる文章として記述するのみでは不十分である。目的、要求、制約、業務プロセス、データ、AI モデル、インターフェース、評価基準、テスト、運用、監査、変更管理を相互に接続し、トレーサビリティを確保する必要がある。人が読んで理解できるのみでなく、AI、要求管理ツール、モデリングツール、テストツール、監査ツールが扱えるように、Human Readable から Machine Readable、さらには Machine Understandable へと進めていくことが求められる。その一環として、要求・要件や業務プロセス、データ等をモデルとして構造化・可視化することは、それらを実装へとつなげやすくし、ノーコード・ローコードを含む多様な開発手法を活用するための基盤ともなる。本書では、個別の開発手法そのものではなく、その前提となる要求・要件の構造化・可視化に焦点を当てる。
本書は、このような問題意識に基づき、AI 時代に必要となる要求管理の考え方、手順、成果物、注意点を整理するものである。対象は、情報システム部門や開発者に限らない。経営層、事業部門、業務改革担当、DX 推進部門、AI 開発・利用部門、データ担当、法務・コンプライアンス部門、監査部門、調達部門、行政機関の制度設計担当者など、AI やソフトウェアを用いて価値を実現するすべての関係者を想定している。
また、本書が対象とする AI 活用は、AI を組み込んだシステムを新たに開発する場合に限らない。既存の AI サービス等を用いて、企画、管理、品質保証、問い合わせ対応、文書作成、審査、監査、事務処理などの業務を改善・高度化する場合も含む。このような AI 活用においても、AI を使うことを所与とせず、解決すべき課題や実現したい価値を明らかにした上で、AI 利用の必要性と適用範囲、人と AI の役割分担、期待する効果、評価・確認方法、運用後の見直し条件等を明確にすることが重要である。本書では、これらを要求として明確化し、要件、運用、評価及び改善へと接続して継続的に管理することも、AI 時代の要求管理の対象として捉える。
なお、本書では、ISO/IEC/IEEE 29148 [1]、ISO/IEC/IEEE 15288 [2]、BABOK [3]、INCOSE Guide to Writing Requirements [4]などの要求工学、システムズエンジニアリング、ビジネス分析の知見を踏まえつつ、AI 時代に特に重要となる論点を整理する。具体的には、要求の獲得、分析、仕様化、評価、管理という基本プロセスに加え、AI を活用した要求抽出、対話型の要求作成支援、モデリング、トレーサビリティ、変更管理、AI への要求、データ要求、法令・標準・社内規則との接続、運用後のフィードバックまでを扱う。
さらに、本書では、法令、標準、ガイドライン、社内規則、マニュアルも、要求管理の重要な入力として扱う。AI やソフトウェアが社会や企業活動の中核となる時代には、ルールは外部制約ではなく、システムや業務が満たすべき要求の一部である。法令や標準で定められた義務、禁止事項、評価基準、説明責任、監査証跡は、業務プロセス、データ項目、AI モデル、システム仕様、運用手順に反映されなければならない。この観点から、本書は、制度設計とソフトウェア工学を接続する Legal Engineering for Software-Defined Society（LE4SDS）の考え方も取り入れる。

AI 時代の要求管理で目指すべき姿は、要求・要件を一度のみ文書化することではない。社会や業務の変化に応じて、要求から要件化し、モデル化し、実装し、運用データをもとに検証し、再び要求を更新する、循環的でライブな要求管理を実現することである。これにより、価値をタイムリーに実現し、変化に柔軟に対応し、同時にコンプライアンス、透明性、説明責任を確保することができる。

本書が目指すのは、AI に仕事を任せるための要求定義ではない。人間と AI が共通理解を持ち、社会や組織が実現したい価値を、要求、要件、モデル、データ、ルール、評価、運用、改善へとつなげるための要求管理である。AI 時代において、要求管理は、開発の前工程ではなく、変化に強い社会と組織をつくるための中核的な能力である。

本書の使い方

想定読者

前述のように、本書は経営層、事業部門など、AI やソフトウェアを用いて価値を実現するすべての関係者が読者となることを想定している。基本的には各章を順番に確認することを推奨するが、すべての読者が本書を隅々まで読み解くことは難しいと想像するため、想定読者ごとに「優先的に確認・参照することが望ましい章」を「◎」、「次に確認・参照することが望ましい章」を「〇」でそれぞれ選定し、次の表に示す。必要に応じて参考にされたい。

表 1　想定読者ごとの確認・参照を推奨する章

	経営層	事業部門	業務改革担当	DX 推進部門	IT・情報システム部門	開発者	AI 開発・利用部門	データ部門	セキュリティ部門	法務・コンプライアンス部門	監査部門	調達部門	行政機関の制度設計担当者	利用者・現場代表者
第 1 章　要望・要求・要件とは何か	◎	◎	◎	◎	◎	〇	◎	〇	〇	〇	〇	〇	◎	◎
第 2 章　要求工学の基本プロセス	〇	◎	◎	◎	◎	〇	◎	〇	〇	〇	〇	〇	◎	〇
第 3 章　要求管理を支える考え方	◎	◎	◎	◎	◎	〇	◎	〇	〇	〇	〇	〇	◎	〇
第 4 章　グローバル経営時代の要求管理	◎	〇	〇	〇	〇				〇	◎	〇	◎	〇	
第 5 章　AI が変える要求管理の前提	◎	◎	◎	◎	◎	◎	◎	◎	◎	〇	〇		◎	〇
第 6 章　AI による要求管理支援		〇	◎	◎	◎	◎	◎	〇	〇				〇	〇

	経営層	事業部門	業務改革担当	DX 推進部門	IT・情報システム部門	開発者	AI 開発・利用部門	データ部門	セキュリティ部門	法務・コンプライアンス部門	監査部門	調達部門	行政機関の制度設計担当者	利用者・現場代表者
第 7 章　AI 時代の要求管理プロセス		◎	◎	◎	◎	◎	◎	○	○	○	○	○	◎	○
第 8 章　AI 時代の要求獲得・分析・仕様化	○	◎	◎	◎	◎	◎	◎	◎	○	○	○		◎	◎
第 9 章　要求を可視化する	○	◎	◎	◎	◎	◎	◎	◎	○	○	○	○	○	◎
第 10 章　ビルディングブロックやサービスによるシステムの実現		○	○	◎	◎	◎	○		◎	◎	○	◎		
第 11 章　トレーサビリティと DevOps 型要求管理	○	○	○	◎	◎	◎	◎	○	◎	○	◎	○	○	○
第 12 章　外部要件としての法令・規制・標準	○	○	○	○	○		○	○	◎	◎	◎	◎	◎	
第 13 章　規則・ガイドライン・マニュアルを要求定義に活かす	○	○	○	○	○		◎		○	◎	○		◎	○
第 14 章　社内ルールの構造化と標準形式	○	○	○	○	○	○	◎	◎	○	◎	○		○	○
第 15 章　組織への導入と成熟度	◎	◎	◎	◎	◎	○	◎	○	○	◎	○	◎	○	
第 16 章　成果物一覧		◎	◎	◎	◎	○	◎			○	○		○	

主要用語の解説

本書では、「要求管理」「要求定義」という 2 つの用語を使用する。これらの用語の正確な定義は存在しない。しかし、これまで多くの組織で適切でない「要件」が量産され、また、それらが一度定義された後は見直されないことが多かったことを踏まえ、「要件」に関する固定概念のようなものを払拭するところを第一の期待として「要件定義」ではなく、「要求管理」「要求定義」という用語を次の意味で使用する。

- **要求管理**
  要求管理とは、狭義的には、定義した要求について、状態、優先度、版、変更履歴、実装状況及び検証状況を、ライフサイクルを通じて継続的に管理する活動である。要求を目的、要件、設計、コード、テスト、運用ログ等と関連付け、変更理由や影響範囲を追跡可能にすることを含む。これは、要求工学の学術領域や関連書籍で定義される同様、もしくは、同名のプロセスと合致するものである。
  しかし、本書では要求定義も含めて、要求のライフサイクル全般を継続管理する活動として、広義的に本用語を使用する。本書が「要求管理」ハンドブックであることも、それが理由である。
- **要求定義**
  要求定義とは、関係者の要望や課題の背景にある目的を明らかにし、「何を実現したいか」を要求として整理した上で、実装・評価可能な要件へ具体化する活動である。つまり、要件定義よりも上位の概念であり、要件定義を内包する活動のことである。

これらの概念イメージは次のようになる。

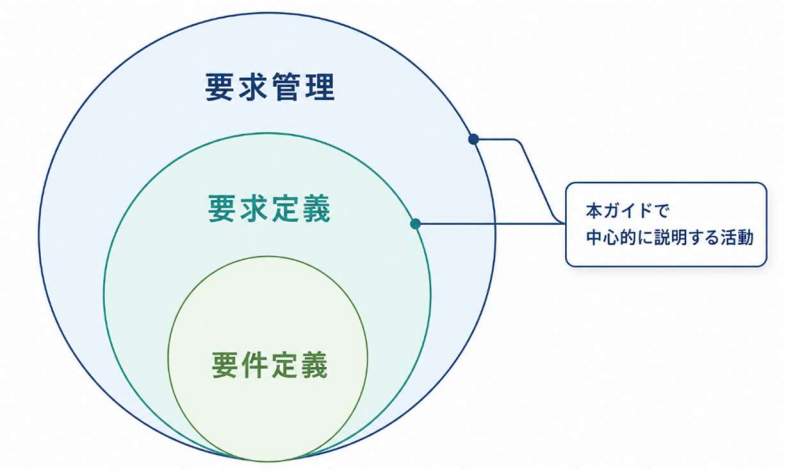


図 1 本書における主要用語の概念イメージ

なお、「要求」「要件」の違いの詳細については、第 1 章で解説しているため、そちらを参照されたい。
AI 時代には、ゼロベースで要件から定義するのではなく要求から考えていくこと、そして、その要求を継続的に管理していくことがより重要であることを、本書を通じて感じていただきたい。

コラム：国際標準における「Requirements」と本書の「要求・要件」

本書では、第 2 章で説明するが、関係者の要望や課題から「何を実現したいか」を整理したものを「要求」、それを実装・評価可能な形へ具体化したものを「要件」と呼び、両者を区別している。一方、国際標準等では、両者に相当する概念を、それぞれ異なる英単語で表し分けているわけではない。
ISO/IEC/IEEE 29148:2018 [1]では、「Requirements」を、ニーズとそれに伴う制約・条件を表現した記述としている。その上で、「Stakeholder Requirements」、「System Requirements」、「Software Requirements」などのように、対象や視点に応じて、「Requirements」に修飾語を付けて区別している。また、これらを定義するプロセスも、深く関連するものの、それぞれ別物として扱っている。
日本産業規格でも同様である。ISO/IEC/IEEE 15288:2023 [2]に対応する JIS X 0170:2025 [5]では、「利害関係者ニーズ及び利害関係者要件（要求事項）定義プロセス」と「システム要件（要求事項）定義プロセス」を区別している。
したがって、本書の「要望」「要求」「要件」という整理は、国際標準、特に ISO/JIS との用語体系との関係を概念的に捉えれば、「Stakeholder Needs」、「Stakeholder Requirements」、「System Requirements」と概ね対応付けて理解することができる[1]。ただし、これは英語と日本語の一対一の翻訳関係を示すものではない。本書では、上位の目的やニーズに近い事項と、システム等が満たすべき事項との違いを読者に分かりやすく示すため、日本語として「要求」と「要件」を使い分けている。
このため、国際的な議論や英文での記述においては、「要求」「要件」を機械的に「Requirements」へ訳すのではなく、対象に応じて「Stakeholder Requirements」、「System Requirements」、「Software Requirements」などと明示することが重要である。同様に、本書で用いる「要求定義」「要件定義」についても、単に「Requirements Definition」と訳すだけでは両者を区別できないため、必要に応じて「Stakeholder Requirements Definition」、「System Requirements Definition」など、対象を明示することが重要である。

---

[1] 本書で紹介した ISO/JIS ではこのような用語体系だが、国際標準すべてが同じという訳ではない。例えば、ISO/IEC/IEEE 29148:2018 でいう「System Requirements」や「Software Requirements」に近しい概念を、CMMI V3.0 では「Solution」やその「Components」に対する「Requirements」と表現している。

コラム：要求工学に関する最新の国際動向

AI 時代の要求管理の在り方を考えるうえで、要求工学分野において世界的にどのような課題や解決策が注目されているかを知ることは有用である。本コラムでは、読者が自らの組織や業務の課題と照らし合わせ、今後何を重視し、どの領域に注力すべきかなどを考えるきっかけとなることを期待して、2026 年に開催された要求工学国際会議（RE 2026）及びソフトウェア工学国際会議（ICSE 2026）の研究発表を中心にその研究動向を次のとおり概観する。

① 実現したい価値や目的を起点とする要求定義

技術的な実現方法を検討する前に、何を実現したいのかを確認し、関係者間で共有された目的を要求・要件へと結び付けることが重視されている。例えば、経営・業務上の課題を、AI の課題設定、データ要求、評価指標などへ具体化する複数の既存手法の評価を行う研究が進められている [6]。また、関係者間の共通理解、役割分担、適切な文書化等の重要性を示唆・提言する研究や、関係者間で組織目的の妥当性を合意し、それを担うアクターと活動まで展開することで、要求工学の上流をつなぐゴールモデルを構築する手法を提案した研究もある [7] [8]。

② AI による要求獲得・分析の高度化

AI による要求獲得の効率化だけでなく、複数の情報源や既存の要求工学の手法を組み合わせ、要求の網羅性や妥当性を高める方向性が見られる。例えば、AI による自動インタビューや要求候補の抽出の研究や、ユースケースやデータモデル等を用いて AI による要求の抜け漏れを補完する研究がある [9] [10]。また、相互に関連するステークホルダー像を AI によって対話ログから追跡可能な形で抽出する要求獲得手法を提案する研究や、プロダクト品質特化型の複数の AI エージェントに要求を交渉させて品質間のトレードオフを明示化・検証する要求分析フレームワークを提案する研究も進められている。
[11] [12]

③ AI による要求・要件の構造化と意味的な妥当性確認

自然言語で記述された要求・要件を、AI を用いて UML 等のモデルへ変換する研究が進展している [13] [14]。こうした構造化は、要求・要件の曖昧さや矛盾を明らかにし、関係者間の認識を合わせることにも役立つ。一方、AI によって生成されたモデルや仕様が形式的に正しくても、本来の意図や実際の業務に照らして妥当であるとは限らない。例えば、医療分野では、AI が生成した構造化仕様について、構文的な正確性と臨床上の妥当性を区別して評価する研究が進められている [15]。このため、AI による仕様化の自動化と、対象領域の知識を持つ人間による内容の妥当性確認を組み合わせることの重要さが増していると見られる。

④ 要求・要件と実装・検証の一貫した接続（トレーサビリティ）

要求・要件や開発者の意図といった自然言語情報を、実装やテストと結びつけて活用する技術・知見が

蓄積されつつある [16] [17]。例えば、AI が要求をどう解釈したかを生成コードから逆抽出し、元の要求記述との食い違いを補正する研究や、変更意図と起点となる編集箇所から同時に修正すべきコード要素の範囲を推論で絞り込む研究が進められている [18] [19]。また、要求・要件を開発前に作成する文書として終わらせず、トレーサビリティを確保しながら、その実現状況を継続的に検証・管理することが重視されている [20]。

⑤ AI システム固有の要求と人間の役割・責任の明確化

AI を組み込んだシステムでは、従来の機能・性能に加え、データの条件や品質、AI モデルの振る舞いの不確実性、想定外の状況や失敗時の対応などを要求・要件として捉える研究が進んでいる。例えば、データ抽出をゴール指向要求工学（KAOS）に組み込む参照モデルと拡張課題を提示した研究や、AI の失敗を起点としてデータ・モデル・システム・要求を繰り返し見直す枠組みを提案する研究がある [21] [20]。また、自律型 AI の動作制約の仕様化と実行時強制する研究や、人間への情報提供、判断支援、制御権限、介入等を要求として整理する研究も進められている [22] [23]。こうした動向からは、AI の性能だけでなく、その利用条件や限界、人間による判断・監督の在り方を明確にすることの重要性が読み取れる。

これらの研究動向は、要求工学が単に要求・要件を文書化するための活動ではなく、実現したい価値を明確にし、関係者間の共通理解を形成しながら、要求・要件と実装・評価・運用を継続的に結び付ける活動であることを改めて示唆している。これは AI 時代になって新たに生まれた考え方ではなく、要求工学が従来から重視してきた基本的な考え方である。AI による要求獲得・仕様化・実装の自動化が進む中で、こうした考え方の重要性が一層高まるとともに、その実践を支える技術や手法も高度化していると捉えることができる。

ただし、AI 活用という点では提案段階や限定的な実証にとどまる研究も多く、各手法の実務での有効性が十分に確立されているわけではない。AI による支援が高度化しても、要求の目的や妥当性を判断し、関係者間で合意する人間の役割は引き続き重要である。

実プロジェクトでのテーラリング

多くのプロジェクトでは、本書で示す活動や成果物のすべてを同じ水準で適用することは費用対効果の面から現実的ではないと考えられる。そのため、プロジェクトの規模、複雑性、リスク、要求の変動性、利害関係者やインターフェースの数、法令・監査への対応等に応じて、要求管理の範囲や深さをテーラリングすることを推奨する。

特に組織への導入初期には、過度な成果物作成や管理作業による形骸化を避けるため、小規模でリスクが低く、試行錯誤しやすいプロジェクトでの試行や、簡易な成果物を用いた管理から始め、効果を確認しながら段階的に拡大することが望ましい。

要求管理に必要な工数について、一律に適用できる最適な比率は存在しない。ただし、参考として、システムズエンジニアリング活動はプロジェクト総コストの 10～14%で行うことが ROI としてよいとされており [24]、その活動の 1 つである要求工学については 2%程度がよいと示す研究もある [25]。

本書では、要求定義を含め、要求のライフサイクル全般を継続的に管理する活動を「要求管理」として扱っている。このため、予算計画上の初期的な目安が必要な場合には、総プロジェクトコストの 2～5%程度を出発点とし、プロジェクトの特性に応じて調整することを推奨する。ただし、この比率は最適値を保証するものではなく、要求起因の手戻り、変更頻度、受入条件の明確性、トレーサビリティ、変更影響分析に要する時間等の実績を確認しながら継続的に見直すことが望ましい。

特に、AI を扱うプロジェクトでは、要求管理の対象や確認事項が増える場合がある。しかし、そのようなプロジェクトすべてで一律に要求管理の工数を増やす必要があるわけではない。AI と人の役割分担、評価方法、利用するデータ、守るべき制約、変更履歴や証跡の管理など、プロジェクトで実際に必要となる活動を踏まえて工数を見積もることが重要である。また、AI を要求の抽出や整合性確認、変更影響分析等に活用することで、要求管理そのものを効率化できる場合もある。このため、投入コストの割合だけでなく、必要な要求管理活動が十分に行われているかを確認する必要がある。

具体的な導入手順や成熟度モデルについては、第 15 章を参照されたい。

第 1 部　要求管理の基礎

AI 時代の要求管理を考えるために、まず、要求管理、特に、要求定義の基礎から見直す。要求定義は、情報システム開発の一工程にとどまるものではない。組織や社会が何を実現したいのかを明確にし、それを業務、データ、ソフトウェア、運用、評価へとつなげる活動である。この基本的な考え方は、AI 時代においても変わらない。

しかし、従来から要求定義は、発注者が作りたいものを説明し、開発者が仕様として整理する作業と考えられることが多かった。実際には、発注者自身も最初から明確な答えを持っているとは限らない。現場には暗黙知があり、経営層には目的があり、利用者には不満や期待があり、法令・標準・社内規則には守るべき制約がある。これらを整理し、関係者が共通理解できる形にすることが要求定義の本質である。

AI 時代には、この重要性がさらに高まる。AI は、文章を作成し、要求候補を抽出し、業務フローを整理し、プログラムコードを生成することができる。しかし、AI に何をさせるのか、どの範囲で使うのか、何を守るべきか、どのように評価するのかが曖昧であれば、AI の出力も曖昧になる。AI 時代に必要なのは、要求定義を省略することではなく、AI も人間も理解できる形で要求を明確にすることである。

このように、要求定義の本質は AI 時代においても変わらないが、AI の活用によって、要求管理の対象や実践方法は拡張される。

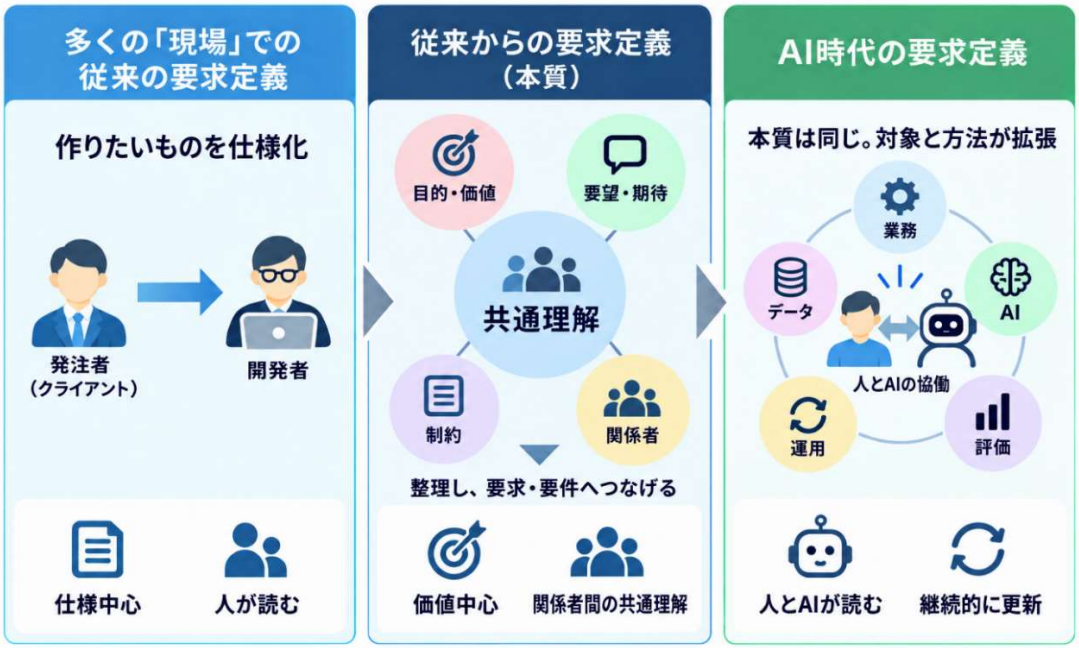


図 2　従来の要求定義と AI 時代の要求定義

第 1 部では、要求定義の基礎として、まず「要望」「要求」「要件」の違いを整理する。そのうえで、要求工学の基本プロセスである、要求獲得、要求分析、要求仕様化、要求評価、要求管理を説明する。さ

らに、AI 時代の要求定義を単独の作業としてではなく、デザイン思考、ミッションエンジニアリング、システムズエンジニアリング、MBSE（Model-Based Systems Engineering）、アジャイルガバナンスと接続する活動として位置づける。

第 1 章　要望・要求・要件とは何か

1.1 要求定義の出発点

要求定義の失敗は、システム開発や AI 導入の失敗に直結する。開発したシステムが使われない、AI の出力が現場の期待と合わない、後から多くの仕様変更が発生する、法令や社内規則に対応できない、運用開始後に想定外のトラブルが起きる。これらの問題の多くは、実装段階に限らず、要求定義段階に原因がある。

要求定義とは、単に「必要な機能を一覧にすること」ではない。何のためにそのシステムや AI を導入するのか、誰にとって価値があるのか、どの業務を変えるのか、どのデータを使うのか、どの制約を守るのか、何をもって成功とするのかを明確にする活動である。

このとき重要なのが、「要望」「要求」「要件」を区別することである。日常会話では、これらの言葉は混同されやすい。しかし、要求定義では、この違いを意識しないと、関係者の期待がずれたまま開発や導入が進んでしまう。

1.2 要望、要求、要件の違い

要望とは、利用者や関係者の望み、期待、困りごとである。要望は、必ずしも具体的ではない。むしろ、最初は漠然としていることが多い。

例えば、ある利用者が「もっと楽に仕事をしたい」「顧客対応を早くしたい」「安全に AI を使いたい」「手続を簡単にしたい」と言ったとする。これらは重要な声であるが、そのままではシステムや AI に実装できない。何をどのように改善すべきか、どの範囲を対象にするのか、どの制約を守るのかが明確ではないためである。

要求[2]とは、要望をもとに、実現したいこととして整理したものである。要望が「困りごと」や「期待」であるのに対し、要求は「何を実現したいか」をより明確にしたものである。

例えば、「顧客対応を早くしたい」という要望は、「問い合わせ内容を分類したい」「過去の回答例を検索したい」「担当部署を自動で振り分けたい」「回答案を作成したい」といった要求に整理できる。

要件とは、要求をさらに具体化し、実現しなければならない事項として定義したものである。要件は、実

---

[2] 国際的には広く Requirements と呼称されており、「要求」「要件」どちらの意味も包含しているが、本書では「要求」と「要件」を区別する。詳細は第 1 部の前に記載したコラムを参照されたい。

装、評価、受入、運用につながるレベルで記述される必要がある。
例えば、「問い合わせ内容を分類したい」という要求は、「問い合わせ受付時に、問い合わせ内容を苦情、質問、申請、障害報告、その他に分類すること」「分類結果は担当部署に通知されること」「分類結果は担当者が修正できること」「分類精度を月次で評価すること」といった要件に具体化される。

表 2 要望・要求・要件の違いと具体例

	意味	例
要望	望み、期待、困りごと	問い合わせ対応を早くしたい
要求	実現したいこと	問い合わせ内容を分類したい
要件	実現しなければならない事項	問い合わせを 5 分類し、担当部署へ通知すること

要求定義では、要望をそのまま受け取って実装しようとしてはいけない。要望の背景にある目的を確認し、要求として整理し、実装・評価可能な要件へ落とし込む必要がある。なお、すべての要求がそのままシステムや AI の要件となるわけではない。要求の実現手段を検討した上で、システムや AI が担う部分について必要な要件を具体化する。

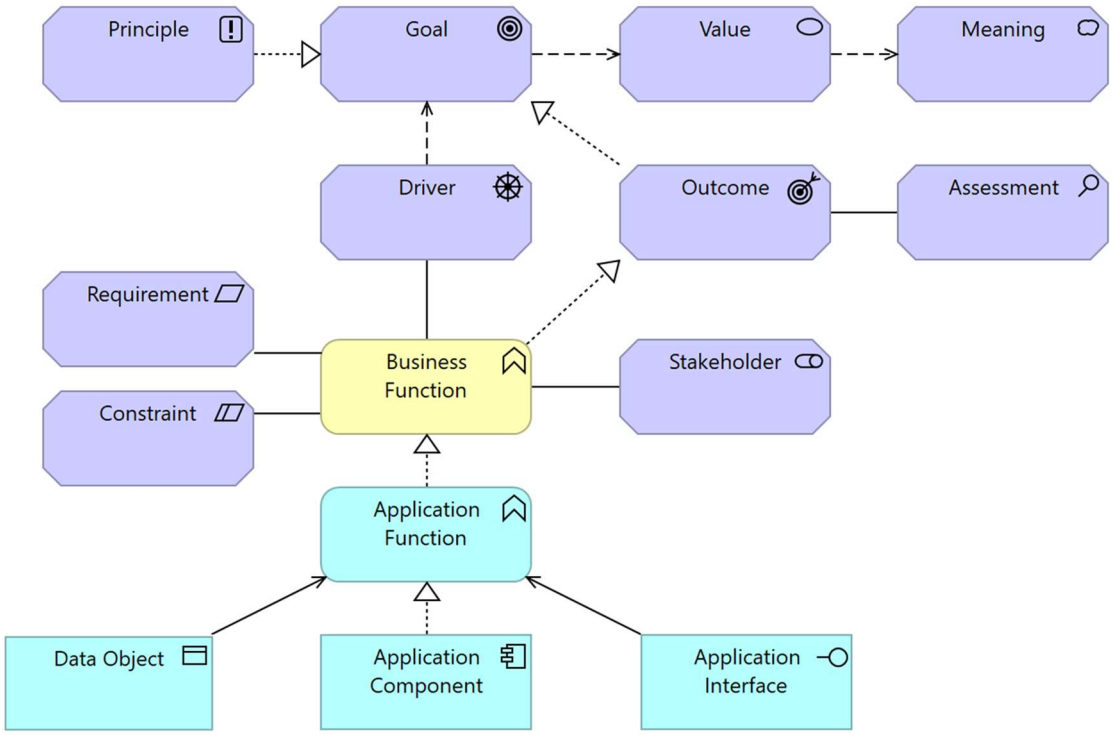


図 3 要件の構成要素例（Archimate[3]による記述）

[3] ArchiMate（アーキメイト）とは、The Open Group が標準化しているエンタープライズアーキテクチャ（EA）を記述するためのモデリング言語である。

何か要望を考えたとき、それは Goal であり、どのような機能があるのか、そこには細かい要求条件や制約があり、多くの関係者が関係するという全体像を考える必要がある。

1.3 木のブランコの例から学ぶ正確な要求定義の重要性

要求定義の説明では、木にぶら下がったブランコの絵がよく使われ、様々な派生形が作られている。ここでは、要望、要求、要件の違いをこの絵を例にして考える。

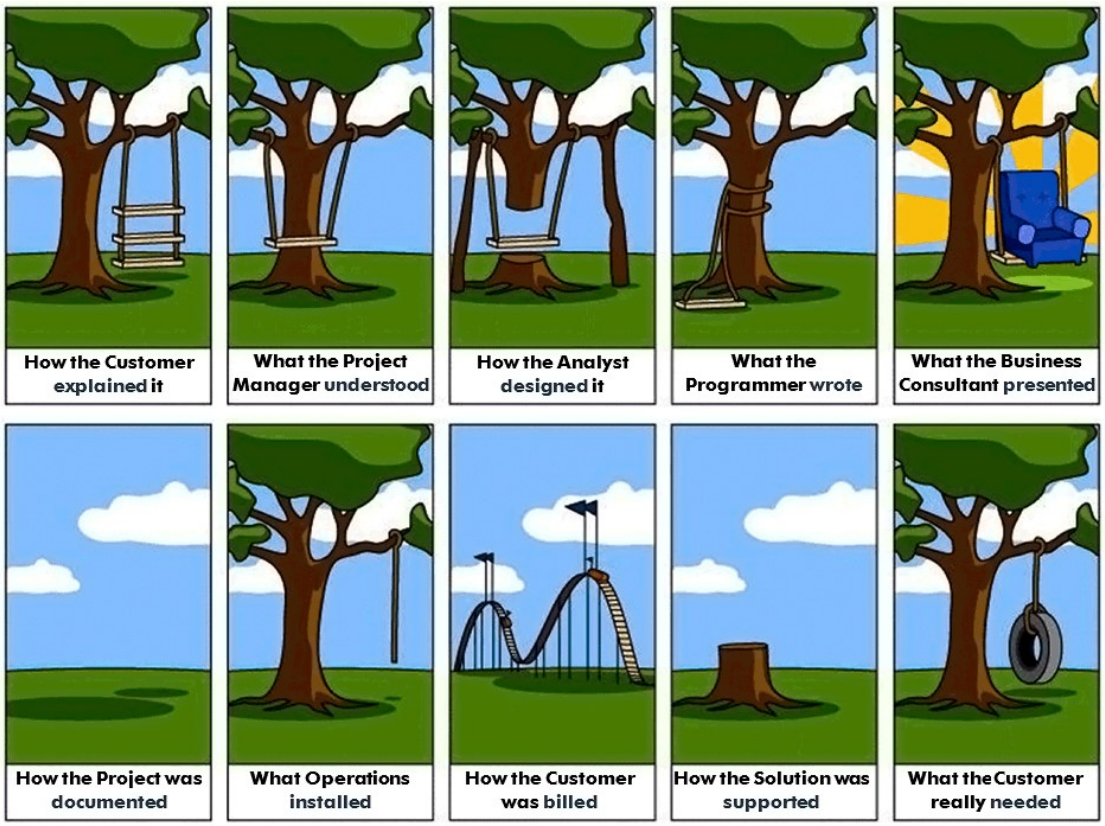


図 4 木のブランコの例にみる「顧客が本当に必要だったもの」
出典：Smart Words “How IT-Projects really work” より [26]

最初の要望は、「木の下で楽しく過ごしたい」ことである。これは、利用者の望みであり、まだ具体的な実現方法は決まっていない。
この要望を整理すると、いくつかの要求が見えてくる。例えば、「枝にロープでぶら下がりたい」「座って遊びたい」「子どもが安全に使えるようにしたい」といったものである。
さらに、制約条件も存在する。例えば、「ロープは 3 メートルしかない」「庭にある部材しか使えない」「大きな工事はできない」「子どもが使うので安全でなければならない」といった条件である。

これらを踏まえると、要件としては、「枝からつるすロープは 1 本とする」「座る部分には庭にあるタイヤを活用する」「ロープの結び目は荷重に耐える方法とする」「地面から座面までの高さは子どもが安全に乗り降りできる高さとする」といった形になる。
この例から分かることは、要望から要件へ直接飛ぶことはできないということである。要望を理解し、要求を整理し、制約条件を確認し、実現方法を検討し、評価可能な要件にする必要がある。

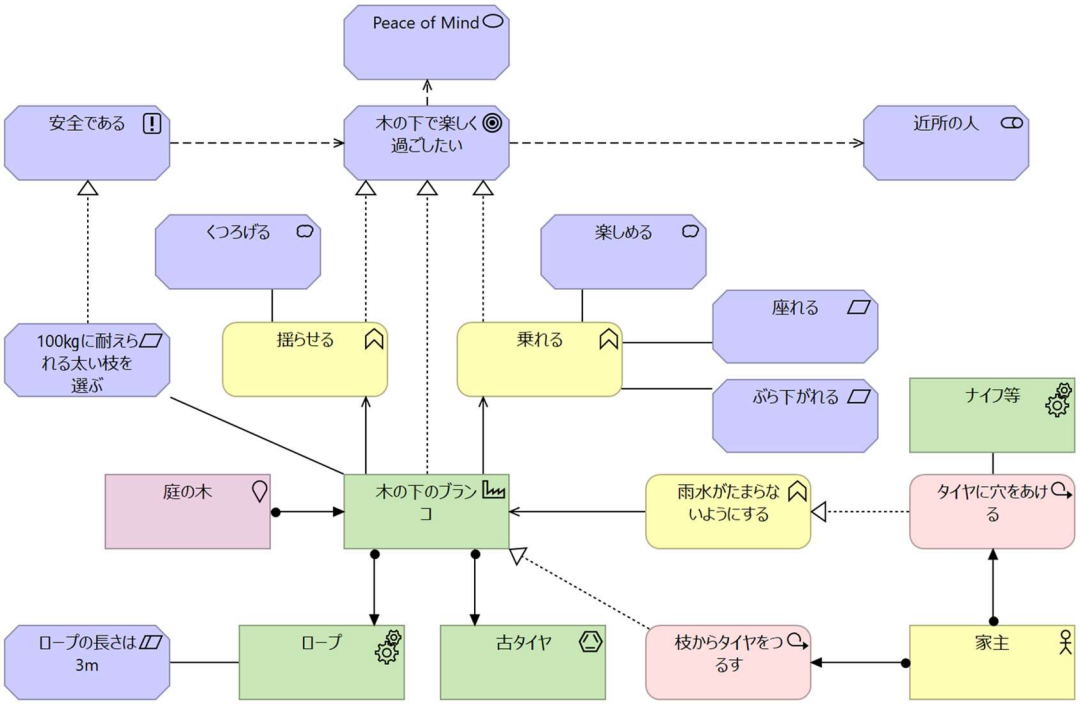


図 5 木のブランコの要件の構成要素例 （Archimate による記述）

AI 時代の要求定義でも同じである。「AI を使いたい」「自動化したい」「効率化したい」という要望のみでは不十分である。何を実現したいのか、どの判断を支援するのか、どのデータを使うのか、人間はどこで関与するのか、誤りが発生した場合にどう対応するのか、どのように効果を測るのかを整理しなければならない。

1.4 所与の条件をシンプルにする

要求定義では、利用者から出てきた要望や、既存業務、既存規則、既存システムをそのまま前提にしてしまうことがある。しかし、重要なのは、与えられた条件を自動的にそのまま要件化することではない。まず、業務の目的に照らして、現在行われている判断そのものが本当に必要なのかを確認する。判断が必要な場合にも、その基準や例外条件を明確にすることで、属人的な判断を減らせないかを検討する。そのうえで、既存の手順、承認、帳票、データ項目、社内規則が本当に必要なのかを目的に照らして確認・整理し、可能な限りシンプルにすることが重要である。AI 時代においては、その重要性がより増す。

例えば、「紙の申請書を PDF で提出できるようにしたい」という要求があった場合、そのまま PDF 提出機能を作ることもできる。しかし、本来の目的が「申請者の負担を減らすこと」「審査を早くすること」であれば、オンラインフォーム、自動入力、既存データの参照、API 連携により、申請書そのものを簡素化できる可能性がある。

また、企業内の承認業務では、「これまで部長承認が必要だったから」という理由のみで、同じ承認プロセスが新システムにも組み込まれることがある。しかし、まずは承認という判断自体が本当に必要なのかを検討すべきである。例えば、低リスクで定型的な案件については、必要な統制を確保しつつ、承認自体を不要にできる場合がある。また、承認が必要な場合にも、金額、リスク、取引先、情報種別に応じて承認レベルを分ければ、低リスク案件は自動処理し、高リスク案件のみを人間が確認する仕組みにできるかもしれない。

AI やソフトウェアは、複雑な業務をそのまま実装することもできる。しかし、判断の必要性や根拠が複雑なまま実装すれば、運用も保守も監査も難しくなる。そのため、要求定義では、所与の条件を疑い、目的に対して不要な判断や手順をなくすとともに、必要な判断の基準や責任を明確にすることが重要である。

シンプル化は、単なる業務削減ではない。要求の本質を明確にし、AI やシステムが扱いやすくし、人間が理解しやすくし、変更に強くするための活動である。そのため、AI 時代には、複雑な業務をそのまま AI に渡すのではなく、要求定義の段階で構造を整理し、可能な限り単純で明確な仕組みにしておくことが重要である。

1.5 要求定義が生む価値

要求定義は、単なる文書作成ではない。良い要求定義は、企業や組織に大きな価値をもたらす。その代表的な価値は次の 8 点である。

1. 関係者の共通理解を促進する

   システムや AI の導入では、経営層、事業部門、現場、IT 部門、法務、セキュリティ、データ担当、外部ベンダーなど、多くの関係者が関わる。それぞれが異なる言葉で話し、異なる期待を持っている。要求定義は、これらの関係者が同じ対象を同じように理解するための共通言語となる。

2. 手戻りを減らし、リードタイムを短縮する

   要求が曖昧なまま開発や AI 導入を進めると、後になって「思っていたものと違う」「この業務には使えない」「法令上問題がある」「データが足りない」といった問題が生じる。要求定義の段階で目的、範囲、制約、評価基準を明確にしておけば、手戻りを減らし、結果としてコスト削減と価値創出を同時に実現できる。

3. システムや AI による実行可能性を高める

   AI やソフトウェアは、曖昧な期待では動かない。何を入力とし、何を出力し、どの条件で処理し、何を例外とし、どのように評価するかが明確であるほど、実装や自動化がしやすくなる。

4. コンポーネント化や再利用を進める

要求が整理されていれば、共通機能、共通データ、共通ルール、共通プロセスを見つけやすくなる。これにより、システムの部品化、サービスの再利用、標準化が進み、開発効率が高まる。

5. 柔軟な改善が可能になる

   要求、要件と実装、データ、テスト、運用ログがつながっていれば、変更が必要になったときに、どこを直せばよいかを把握しやすい。これは、AI やデータを使うシステムのように、運用後も継続的に改善が必要な領域で特に重要である。

6. 事前シミュレーションや検証が可能になる

   要求、要件がモデル化されていれば、業務プロセス、データ、ルール、システムの挙動を事前に確認できる。建物や機械では設計段階でシミュレーションを行うことが一般的である。AI 時代のシステムや社会制度においても、要件をモデル化することにより、事前に影響やリスクを確認することが重要になる。

7. コンプライアンスと説明責任を支える

   法令、標準、契約、社内規則に基づく要求が要件として明確になっていれば、なぜその機能が必要なのか、どの規則に基づくのか、どの証跡を残すべきかを説明しやすくなる。AI の利用においては、判断根拠、ログ、監査証跡、評価結果を残すことがますます重要になる。

8. 危機対応を容易にする

   障害、事故、セキュリティインシデント、AI の誤判定、不正利用が発生した場合、要求、要件、設計、データ、コード、運用ログがつながっていれば、原因究明と対策が迅速になる。逆に、要求定義が十分にされておらず、システム構造も不明確であれば、問題箇所の特定に時間がかかる。

このように、要求定義はコストではなく、価値創出、品質向上、リスク低減、説明責任、継続的改善を支える投資である。

1.6 要求管理は職人芸から組織能力へ

多くの組織では、要求管理が特定の担当者の経験や暗黙知に依存している。優秀な担当者がいればうまくいくが、その人が異動すると品質が落ちる。部署やプロジェクトごとに書き方が異なり、要求、要件の粒度もばらばらになる。変更管理も不十分で、どの要件がいつ、なぜ変わったのか分からなくなる。こうした属人化は、AI 時代以前から要求管理の品質や継続性を損なう要因であった。

AI 時代には、この従来からの課題がさらに深刻になる。AI、データ、クラウド、セキュリティ、法令、標準、社内規則が複雑に関係する中では、要求の変更が及ぼす影響を把握し、関係者間で共有する必要性が一段と高まる。要求管理を個人の職人芸に任せるままでは、AI 時代の早い変化に対応できない。

必要なのは、要求管理を組織能力として整備することである。標準的なプロセス、テンプレート、用語、モデル、レビュー方法、要求管理台帳、変更管理ルール、トレーサビリティ管理を整備し、誰が担当しても一定の品質を確保できるようにする必要がある。

もちろん、標準化は柔軟性を失わせるものではない。むしろ、標準化により、共通理解が生まれ、プロジェ

クト間の比較ができ、AI やツールの活用がしやすくなる。記法やテンプレートは制約ではなく、関係者が協働するための共通言語である。

1.7 AI 時代に一層重要となる要求管理の観点

要求管理では、組織や社会が実現したい価値と、関係者間で合意した要求・要件を結び付け、その実現状況や変更による影響を継続的に把握・管理することが重要である。そのためには、従来から重要とされてきた「可視化」「構造化」「トレーサビリティ」「継続的更新」が、AI 時代にはより一層重要となる。

1. 要求・要件の可視化
   文章のみでは、複雑な業務やシステム、データ、AI の関係を理解しにくい。業務フロー、データモデル、システム構成図、ユースケース、状態遷移、意思決定ルール、リスクモデルなどを用いて、関係者が同じイメージを持てるようにする。
2. 要求・要件の構造化
   目的、要求、制約、例外、データ、評価基準、責任主体、運用条件を整理し、人間（関係者）がその内容や関係性を理解し、合意できるようにする。そのうえで、AI やツールも扱える形に構造化することで、分析や変更管理などに活用できるようにする。Human Readable な文書から、Machine Readable、特に AI 時代には、さらに Machine Understandable な形にすることが重要である。
3. トレーサビリティの確保
   目的から要求、要件、設計、実装、検証などを関連付ける。法令や標準、社内規則に由来する要求も、どのシステム機能やデータ項目に反映されているかを追跡できるようにする。
4. 要求・要件の継続的な更新
   事業環境や利用者ニーズ、技術等の変化を踏まえて要求・要件の妥当性を継続的に確認し、必要に応じて変更の影響を評価しながら適切に更新する。

これらは AI 時代に新たに生まれた考え方ではない。しかし、AI の活用が進むことで、管理すべき対象やその関係が複雑化し、変化への迅速な対応も求められるため、その重要性は従来以上に高まっている。こうした AI 時代における要求管理の対象や方法の変化については、第 2 部で詳しく説明する。

第 2 章　要求工学の基本プロセス

2.1 要求工学とは何か

要求は工学的に管理されることが望ましい。工学的に管理された状態とは、一意に表現でき、再現性があり、シミュレーションができる状態である。つまり、要求工学とは、関係者のニーズを明らかにし、それを要求として整理し、仕様化し、評価し、変更を工学的に管理するための方法論である。情報システムやソフトウェア開発の分野で発展してきたが、AI 時代には、より広い対象に適用される。

要求工学は、単に要求定義書（あるいは、要件定義書）を書くための作業ではない。何を実現すべきか、なぜ実現すべきか、誰にとって価値があるのか、どの制約を守るべきか、どのように確認するのかを明確にする活動である。

なお、具体的な要求を獲得する前には、そもそも何を解決し、何を実現すべきかという上位の目的を明確にすることが重要である。経営戦略や事業・ミッション上の問題・機会、達成すべき目標、成功の尺度等を整理した上で、その実現に必要な要求を具体化していく[4]。これらの分析は一度確定して終わるものではなく、要求獲得や要求分析を通じて得られた知見を踏まえて見直すことも必要である。

一般に、要求工学は次の五つの活動に整理できる[5]。

表 3 要求工学の五つの活動

	英語	概要
要求獲得	Requirement Elicitation	関係者や文書、現場、データから要求候補を集める
要求分析	Requirement Analysis	要求を分類し、矛盾、重複、優先度、実現可能性を整理する
要求仕様化	Requirement Specification	要求を文書、モデル、受入条件として明確に記述する
要求評価	Requirement Validation	要求が正しく、必要で、実現・検証可能か確認する
要求管理	Requirement Management	要求の変更、版、状態、トレーサビリティを管理する

これら五つの活動を通じて重要となるのが、利害関係者間のコミュニケーション、共通理解、合意形成である。要求・要件は、文書として作成するだけでは十分ではない。関係者がその背景や目的、内容、優先順位、制約を正しく理解し、その実現に向けて必要な役割を果たすことについて認識を共有する必要がある。そのため、要求獲得、要求分析、要求仕様化、要求評価、要求管理の各活動では、関係者への情報共有と認識合わせを継続的に行い、必要な合意とコミットメントを得ながら進めることが重要である。

また、この五つの活動は、順番に一度のみ実施するものではない。実際には、要求獲得を進める中で分析が必要になり、分析の結果として再度ヒアリングが必要になり、仕様化した要求を評価した結果として要求の修正が必要になる。AI 時代には、運用後のデータやログ、利用者のフィードバックをもとに、要求管理から再び要求獲得や要求分析に戻ることも多くなる。

2.2 節以降で、この五つの活動について、AI の利用有無にかかわらず共通する要求工学の基本を説明

---

[4] ISO/IEC/IEEE 15288:2023 [2]では、このような活動を「ビジネス又はミッション分析（Business or Mission Analysis）」として位置付けている。

[5] 明確な定義は存在しないが、各参考文献の内容を加味し、IPA 独自に整理している。この 5 分類は REBOK（Requirements Engineering Body of Knowledge） [27]の整理が一番近しい。

したうえで、AI 時代に追加・強化される観点を補足する。AI 時代の要求管理の詳細については、第 2 部で解説する。

2.2 要求獲得

要求獲得とは、経営戦略や事業・ミッション上の目的・課題を踏まえ、関係者や環境から要求候補を集める活動である。本書で整理する要求工学の基本活動の入口に当たり、ここで見落としがあると、その後の分析や仕様化がどれほど丁寧でも、重要な要求が欠けたままになる。

要求獲得に当たっては、単に関係者の要望を集めるのではなく、その背景にある課題や目的を理解することが重要である。特に、ビジネス又はミッション分析を通じて明確にした目的や実現したい価値を関係者間で共有することが、価値のある要求を獲得するための前提となる。

要求は一つの情報源や一つの手法だけでは十分に把握できないことがある。例えば、経営層へのインタビューは上位目的や事業上の意図を把握する上で有効である一方、現場の暗黙知や例外的な業務を把握するには現場観察等が適している。このため、要求獲得では、対象や目的に応じて複数の手法を組み合わせ、異なる観点から要求候補を収集することが重要である。BABOK [3]なども参考にした上で、要求獲得の代表的な方法を次に示す。

表 4 要求獲得の代表的な方法

	内容	向いている場面
インタビュー	関係者に個別に聞く	経営層、現場責任者、専門家の意見を深掘りする
ワークショップ	複数の関係者で議論する	部門間の認識合わせ、業務改革
現場観察	実際の業務を見る	暗黙知、例外処理、現場の工夫を把握する
アンケート	多数の意見を集める	利用者数が多い場合、傾向を把握する
既存文書調査	経営戦略、事業計画、規程、マニュアル、仕様書等を確認する	上位目的、事業課題、既存業務、法令・標準、制約の確認
業務フロー確認	業務の流れを整理する	手続、承認、責任分担の把握
プロトタイプ提示	試作品を見せて妥当性を確認する	利用者に完成イメージを持たせる
ユーザーストーリーマッピング	利用者体験を時系列で整理する	サービス設計、アジャイル開発
ブレーンストーミング	多くのアイデアを生み出し、整理し、優先順位を付ける	創造的な検討、多様な視点の収集、業務改善案の洗い出し、リスクや例外ケースの発見

なお、要求獲得で確認すべき基本的な問いは、5W1H で整理できる。この基本を丁寧に確認すること

が、要求の抜け漏れを防ぐ第一歩である。

表 5 要求獲得で確認すべき 5W1H

	問い
When	いつ、どのタイミングで必要か
Where	どこで、どの状況で必要か
Who	誰が関係するか、誰が責任を持つか
What	何を実現したいか、何が問題か
Why	なぜ必要か、どの価値を実現するか
How	どのように実現するか、どの制約があるか

AI 時代には、要求獲得の対象がさらに広がる。問い合わせ履歴、クレーム、事故報告、ログ、SNS、FAQ、会議録、チャット、保守記録、監査指摘、パブリックコメントなども、要求候補の重要な情報源になる。AI を使えば、これらの大量のテキストやデータを分析し、頻出する課題、例外処理、未対応のニーズ、リスクの兆候を抽出できる。

ただし、AI で抽出された要求候補は、そのまま採用してはいけない。AI は、過去に多く見られる意見や一般的な方法を優先する傾向にあるため、まだ普及していない新しい解決策、少数だが重要なリスク、専門家でなければ気づかない代替手段を見落とす可能性がある。要求獲得では、AI による分析と人間による専門的判断を組み合わせることが重要である。

こうした AI 時代の要求獲得については、第 8 章で詳しく解説する。

2.3 要求分析

要求分析とは、要求獲得で集めた要求候補を整理し、実現すべき要求として妥当かどうかを分析する活動である。要求候補は、多くの場合、そのままでは使えない。重複しているもの、矛盾しているもの、抽象度が違うもの、目的と手段が混ざっているもの、法令や技術制約に合わないものが含まれるためである。

要求分析では、まず要求を分類する。要求は、実現したい目的、要求を持つ主体、要求の対象、求められる品質や制約など、様々な観点から分類できる。代表的な分類の例を次に示す。

表 6 要求の分類の例

	内容
ビジネス要求	経営・事業として達成したいこと
ステークホルダー要求	利用者、顧客、現場、管理者等の要求
システム要求	システム全体が満たすべき要求
ソフトウェア要求	ソフトウェアが実装すべき機能や制約

データ要求	データの項目、品質、出典、保存、利用条件
ルール要求	法令、標準、契約、社内規則等に基づく要求
評価要求	テスト、受入基準、監査、適合性評価
運用要求	保守、監視、障害対応、教育、問い合わせ対応

この分類は一例であり、相互に排他的なものではない。例えば、データや運用に関する要求は、ステークホルダー要求やシステム要求の一部として整理される場合もある。また、システム要求をさらに機能要求と非機能要求に分けたり、非機能要求のうちの 1 つのセキュリティ要求を大分類の 1 項目に昇格させたりと、プロジェクトの特性に応じて分類の仕方を検討することが望ましい。

次に、要求間の関係を確認する。ある要求が別の要求の前提になっている場合もあれば、互いに矛盾する場合もある。例えば、「利用者登録を簡単にしたい」という要求と、「本人確認を厳格に行いたい」という要求は、どちらも重要だが、実現方法によっては衝突する。要求分析では、このようなトレードオフを明らかにし、優先度や実現方法を検討する。

また、要求の背景にある目的を確認することも重要である。要求は、しばしば手段として表現される。例えば、「紙の申請書を PDF で提出できるようにしたい」という要求があった場合、本当の目的は「申請者の負担を減らしたい」「審査を早くしたい」「遠隔でも手続できるようにしたい」かもしれない。目的を確認すれば、PDF 提出ではなく、オンラインフォーム、自動入力、データ連携、API 連携の方が適切な解決策になる場合がある。さらに言えば、要求の実現手段は、必ずしもシステムや AI に限らない。人による判断、業務プロセスの変更、組織や権限の見直し、データの整備、ルールや制度の変更等によって実現する場合もある。要求分析では、それぞれの要求について適切な実現手段を検討し、そのうちシステムや AI に担わせる範囲を明確にすることが重要である。

要求分析では、少なくとも次の観点を確認する。

- その要求はどの目的に対応しているか
- その要求は本当に必要か
- 所与の条件を簡素化できないか
- ルールベース、既存システム、人、AI による対応等の代替手段を比較したか
- システムや AI に担わせる範囲はどこか
- 既存業務や既存ルールをそのまま実装していないか
- AI などの特定の技術や実現手段が目的化していないか
- 要求同士に重複はないか
- 要求同士に矛盾はないか
- 優先度は明確か
- 実現可能か
- 検証可能か
- 技術的制約はあるか
- 法令・標準・社内規則と整合しているか

- データは取得・利用できるか
- 運用後に監視・改善できるか

要求分析は、要求定義の品質を左右する重要な工程である。要求を十分に分析せずに仕様化すると、後工程で大きな手戻りが発生する。

また、要求分析において、コストや期間も重要な判断要素となる。要求を実現する手段は一つとは限らないため、特定の技術を前提とせず、複数の実現手段を比較することが重要である。特に AI を利用する場合は、開発・導入時の費用のみではなく、データ整備、評価、監視、外部サービス利用、モデル等の更新、人間による確認等の運用後の費用も含めて検討し、期待する効果に対して妥当なライフサイクルコストとなるかを確認する必要がある。その上で、一覧化された要求に優先順位をつけて実現していく必要がある。同時に社会や現場の変化や効果を想定し、要求見直しのロードマップを作成することが望ましい。

AI を利用する場合には、従来の要求分析に加え、AI の利用が目的の達成に適しているかを検討し、その適用範囲を明確にする必要がある。また、AI と人間の役割分担、AI の出力に求められる品質、利用するデータの品質や制約、運用後の監視・評価なども、要求として分析する必要がある。

これらは従来の要求分析に置き換わるものではなく、AI の特性を踏まえて分析対象を拡張するものである。具体的な観点については、8.5 節で詳しく解説する。

2.4 要求仕様化

要求仕様化とは、分析された要求を、関係者が理解し、実装や評価に使える形で記述する活動である。仕様化では、文章に加え、図、表、モデル、データ定義、受入条件を組み合わせる。特に複雑なシステムでは、自然言語のみで要求を表現すると、曖昧さや解釈の違いが残りやすい。そのため、要件文、業務フロー、データモデル、画面イメージ、API 仕様、ユースケース、状態遷移、ルール表、テスト条件などを組み合わせて、要求を明確にする。

要求仕様化の代表的な成果物には、次のようなものがある。

表 7 要求仕様化の代表的な成果物

	内容
要件定義書	要求全体から要件を体系的に整理した文書
ユースケース	利用者とシステムのやり取りを整理したもの
業務フロー	業務手順、判断、承認、例外処理を示すもの
機能要件一覧	システムが提供すべき機能の一覧
非機能要件一覧	性能、可用性、セキュリティ、保守性等
データ定義	データ項目、意味、形式、品質、保存条件
API 定義	システム間連携の入出力仕様
受入条件	要求が満たされたことを確認する条件

表 7 に示す成果物は、それらを束ねる形で要求定義書としても、それぞれ独立した成果物として管理してもよい。独立して管理する場合には、要求 ID などを用いて、各成果物と対応する要求との関係を明確にする必要がある。ただし、本書上は、これらの成果物で表現される「図やモデルなどの自然言語以外の成果物を含む要求に関する情報セット」を「要求定義書」と便宜上、呼称する[6]。

良い要件文には、いくつかの条件がある。少なくとも、必要性があり、一意に解釈でき、実現可能で、検証可能で、矛盾がなく、適切な粒度であり、上位目的と下位実装へトレースできる必要がある[7]。

例えば、悪い要件文は次のようなものである。

- システムは使いやすいこと。

これは重要な要件に見えるが、「使いやすい」とは何かが不明確である。検証も難しい。

より良い要件文にするなら、次のように具体化する。

- 利用者は、初回ログイン後、3 分以内に申請入力画面に到達できること。

または、

- 申請入力画面では、必須項目が未入力の場合、保存前に未入力項目名と入力方法を表示すること。

このように、要件文は評価できる形で書く必要がある。

ただし、すべてを細かく書けばよいわけではない。要件の詳細度は、目的、リスク、制約、実現方法の自由度などに応じて適切に調整する必要がある。特に、安全性、コンプライアンス、監査、データ保護に関わる部分は詳細に書く必要がある一方で、新規サービスや探索的な AI 活用では現場の工夫や改善を妨げないように、目的や制約を明確にしつつ、実装方法には一定の柔軟性を残すことも必要である。

AI を利用する場合には、これらの基本に加え、AI の利用目的、入力・出力、求められる品質やその評価方法、人間による確認、想定外の出力への対応、運用時の監視や更新の条件などを要求仕様として明確にする必要がある。特に、AI の出力が一定しない場合には、期待する出力を一意に規定するだけでなく、許容範囲や評価基準を定めることが重要になる。

具体的な仕様化の方法については、8.6 節で詳しく解説する。

2.5 要求評価

要求評価とは、定義された要求が正しいか、必要か、実現可能か、検証可能かを確認する活動である[8]。要求評価を行わないと、間違った要求や曖昧な要求がそのまま開発や AI 導入に進んでしまう。

---

[6] 要求定義書及び要求管理台帳との関係については、第 16 章を参照されたい。

[7] ISO/IEC/IEEE 29148:2018 [1]に準拠している。より正確には、要件に必要な品質・特性として、Necessary、Appopriate、Unambiguous、Complete、Sigular、Feasible、Verifiable、Correct、Conforming が挙げられており、要件集合全体として、Complete、Consistent、Feasible、Comprehensible、Able to be validated が必要とされている。

[8] 要求工学の分野では、Verification And Validation（V＆V）と表現される。ISO/IEC/IEEE

要求評価の代表的な方法には、次のようなものがある。

表 8 要求評価の代表的な方法

	内容
レビュー	関係者が要求文書を確認する
ウォークスルー	業務シナリオに沿って要求を確認する
プロトタイプ評価	試作品を使って要求の妥当性を確認する
シナリオ検証	通常時、例外時、障害時の流れを確認する
ユーザーテスト	利用者が実際に操作して評価する
整合性チェック	要求間、法令・標準、データとの矛盾を確認する
受入基準確認	要求を満たしたと判断する条件を確認する

要求評価では、要求分析と同様に次の問いを確認する。

- その要求はどの目的に対応しているか
- その要求は本当に必要か
- 所与の条件を簡素化できないか
- ルールベース、既存システム、人、AI による対応等の代替手段を比較したか
- システムや AI に担わせる範囲はどこか
- 既存業務や既存ルールをそのまま実装していないか
- AI などの特定の技術や実現手段が目的化していないか
- 関係者の理解は一致しているか（また、合意がとれているか）
- 実現可能か
- 検証可能か
- 技術的制約はあるか
- 法令・標準・社内規則と整合しているか
- データは取得・利用できるか
- 運用後に監視・改善できるか
- 変更が発生した場合に影響を追跡できるか

要求評価は、要求そのものの正しさや実現可能性を確認するだけの活動ではない。要求の背景や目的、内容、優先順位、制約について関係者の理解が一致しているかを確認し、その実現に必要な役割

29148:2018 [1]では、Verification は指定された要求や特性を満たしていることを客観的に示すこと、Validation は特定の意図された用途や適用のための要求が満たされていることを客観的に示すことと定義している。つまり、平易に言えば、「要求を満たせているか」と「要求を正しく定義できているか」を確認することである。要求定義においては、主に後者の妥当性確認を実施するため、本書でもそちらを重視して記載している。

分担や協力について合意を形成することも重要である。要求が形式的に承認されていても、その意味や実現方法について関係者の認識が異なっていれば、設計、実装、運用の段階で齟齬が顕在化する可能性がある。そのため、要求評価を、関係者の共通理解と実現に向けたコミットメントを確認する機会として捉える必要がある。また、要求評価は、要求定義の最後に一度限りで行うものではない。要求獲得、分析、仕様化の各段階で繰り返し行う必要がある。

AI を利用するシステムでは、AI の出力は確率的であり、常に同じ結果になるとは限らないことから、要求評価においても追加的な観点が必要になる。単に「機能が動くか」に加え、精度、再現率、公平性、説明可能性、安全性、頑健性、誤用可能性、運用時の監視方法を確認する必要がある。

また、AI への要求では、人間の関与を評価することも重要である。AI が自動判断するのか、人間に推奨するのみなのか、人間が承認するのか、利用者が異議を申し立てられるのかによって、必要な要求は変わる。これらの観点は、すべての AI システムに一律に適用するものではなく、利用目的やリスクに応じて選択することが重要である。

2.6 要求管理

本書では、冒頭で解説したとおり、要求管理を広義的な意味で用いているが、本項においてのみ狭義的な意味で用いる。つまり、定義された要求をライフサイクルを通じて継続的に管理する活動という、要求工学の分野で定義される要求管理のことを指す。

要求は、一度決めたら固定されるものではない。事業環境、利用者ニーズ、法令、標準、データ、AI モデル、技術、リスクは変化する。そのため、要求も変化する。

要求管理では、要求の状態、版、変更履歴、優先度、関連要件、実装状況、検証状況を管理する。特に重要なのが、トレーサビリティである[9]。トレーサビリティとは、要求とその発生源、上位目的、下位要件、設計、コード、テスト、運用ログ、変更要求をつなぐことである。これにより、ある要求がなぜ存在するのか、どこに実装されているのか、どのテストで確認されるのか、変更時にどこへ影響するのかを把握できる。

ただし、トレーサビリティは、要求と設計や実装との対応関係を記録することだけを目的とするものではない。経営戦略、事業課題、ミッション等の上位目的から、要求・要件、業務、データ、設計、実装、テスト、運用までのつながりを維持することで、それぞれの活動が上位目的と整合しているかを継続的に確認できるようにすることが重要である。また、要求・要件の実現に支障が生じた場合や、関係者間の認識、上位目的、実装内容等に齟齬が見つかった場合には、それを速やかに関係者へ共有し、必要に応じて要求・要件だけでなく、その前提となる目的や業務まで立ち戻って見直す必要がある。このような問題の早期提起と影響分析も、継続的な要求管理の重要な役割である。

---

[9] BABOK [3]では、要求のライフサイクル管理として「トレース・維持・優先付け・変更評価・承認」という 5 つのタスクがあるとしている。「トレース」を中心に各タスクについては第 2 章や第 11 章で触れている。

それを踏まえ、要求管理台帳には、少なくとも次のような項目を持たせることが望ましい[10][11]。

表 9 要求管理台帳に持たせる項目の例

	内容
要求 ID	一意の識別子
要求名	要求の短い名称
要求文	明確に記述された要求
要求種別	業務、システム、AI、データ、ルール等
発生源	要望、法令、標準、会議、ログ等
根拠	背景、目的、法益、理由
優先度	Must、Should、Could 等
状態	候補、承認、実装中、検証済等
関連要求	上位・下位・依存関係
検証方法	テスト、レビュー、監査、シミュレーション
作成履歴	要求の作成者、作成日時
担当者	要求の管理責任者
変更履歴	変更理由、変更日時、承認者
バージョン	要求の版

なお、理想的には要求と要件は別々の台帳で管理する方がよい。承認者や確定時期、検証方法などが異なることが多々あるためである。また、契約や調達の境界になり、事業者レベルで分けざるを得ない場合もある。要求と要件を別々の台帳で管理する場合、管理台帳に持たせる項目はほとんど似たようなものになるが、要件管理台帳にはその要件を実現する対象（システム名、サブシステム名、機能名など）の項目もあるとよい。

一方で、例えば、次のような条件を満たすのであれば、同一管理とすることも視野に入れてよい。

- 関係者が少ない
- 承認者がほぼ同じ
- 対象システムが単純である
- 契約上の責任分解が複雑でない

---

[10] ISO/IEC/IEEE 29148:2018 [1]では、これとは別に、各要求仕様書（ビジネス要求仕様書、ステークホルダー要求仕様書。要求よりは要件に近くなるが、システム要求仕様書、ソフトウェア要求仕様書もある。）の項目例を記載しているため、要求管理台帳に持たせる項目の参考にされたい。

[11] 要求定義書を別途作成する場合、要求定義書上の情報の二重管理は極力避けることを推奨する。項目ごとに正本となる管理対象を定め、要求 ID、参照先又はツール間連携によって情報を関連付けることが望ましい。要求定義書及び要求管理台帳との関係については、第 15 章を参照されたい。

- 同一の変更管理手続で支障がない
- 要求と要件を ID 及び属性によって識別できる

要求管理が不十分だと、変更の影響範囲が分からなくなる。AI がバグや不整合を見つけたとしても、要求定義書がなく、要求と要件、要件とコードやデータの対応関係が分からなければ、修正には時間がかかる。逆に、要求、要件、設計、コード、テスト、運用ログがつながっていれば、原因究明と対策を迅速に行うことができる。

第 3 章　要求管理を支える考え方

3.1 要求管理は単独の方法論ではない

要求管理は、要求工学を基盤としつつ、組織や社会が実現したい価値や目的を、具体的な要求・要件、実現手段、評価・改善と結び付けて継続的に管理する活動である。そのため、要求工学だけでなく、目的や課題の明確化、システム全体の整合性確保、運用後の継続的改善などを扱う他の分野とも密接に関係している[12]。

こうした関係は、AI 時代になって新たに生まれたものではない。従来から、利用者の課題や体験を明らかにするデザイン思考や、達成すべきミッションを起点に必要な能力やシステムを考えるミッションエンジニアリングは、実現すべき目的や価値を明確にする上で役立ってきた。また、システムズエンジニアリング（SE）はシステム全体をライフサイクルにわたって捉え、その実践手段の一つである MBSE はモデルを中心に要求、構造、振る舞い、検証を接続する。さらに、アジャイルガバナンスは、環境やリスクの変化を踏まえ、ゴール設定、システム設計、運用、評価、改善を継続的に行う考え方を提供する。

AI 時代には、こうした分野との連携が一層重要になる。AI の活用によって、業務、データ、AI、ソフトウェア、人の役割分担などを一体として考える必要性が高まり、AI の振る舞いの不確実性や運用環境の変化に対応するため、要求・要件の整合性確保や継続的な見直しも重要になるためである。

---

[12] 本章は、ISO/IEC/IEEE 15288:2023 [2]で示されるビジネス又はミッション分析（Business or Mission Analysis）の考え方を一つの起点とし、AI 時代の要求管理に必要となる観点まで対象を広げて整理したものである。ビジネスやミッションの目的・課題を明確にするだけでなく、それらを要求・要件、システム、実装、運用へとつなげるため、関連する方法論との関係も扱っている。

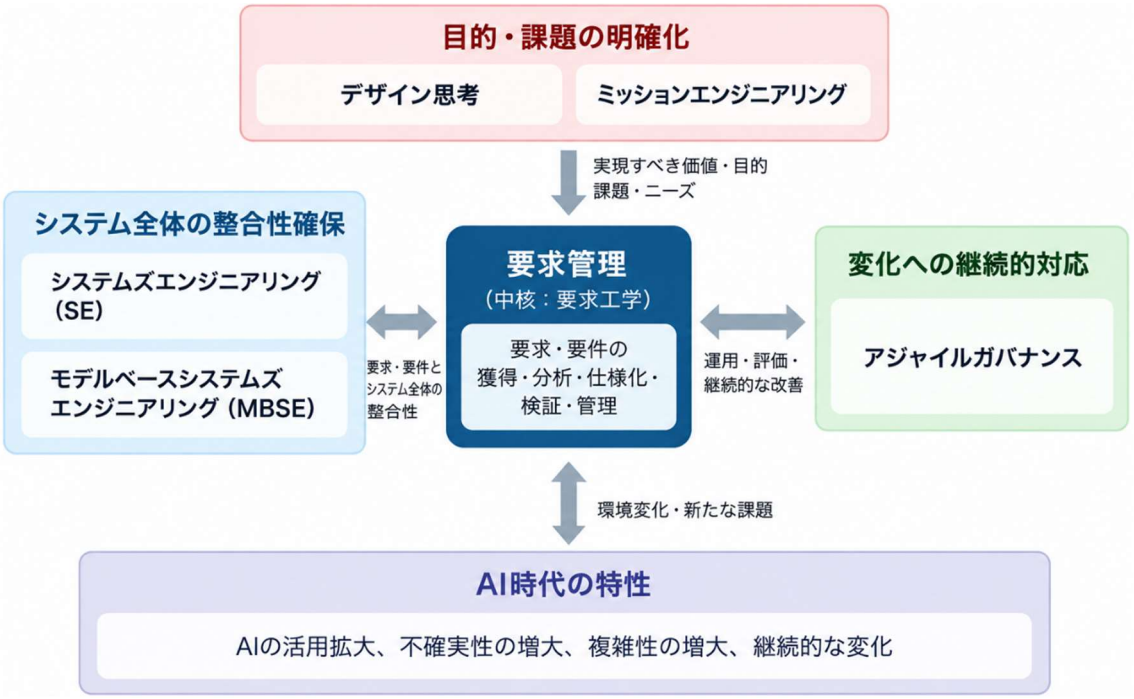


図 6 AI 時代の要求管理

ただし、すべての方法論を一律に適用する必要はない。対象となる業務やシステムの特性、複雑さ、リスクなどに応じて、それぞれの考え方を適切に活用することが重要である。
3.2 節以降で、各方法論と要求管理の関係及び AI 時代に特に重要となる観点を説明する。

3.2 デザイン思考との関係

デザイン思考は、人間中心の課題発見と解決策の探索を重視する考え方である。要求定義においては、利用者や現場が何に困っているのか、どのような体験を実現したいのかを明らかにする上で有効である。
AI 導入の現場では、「AI を使いたい」「チャットボットを入れたい」「自動化したい」という要望が先に出ることが多い。しかし、本当に重要なのは、利用者や業務担当者が何に困っているかである。問い合わせ件数が多いのか、回答の品質がばらつくのか、必要な情報にたどり着けないのか、申請者が手続を理解できないのかによって、要求と必要な要件は異なる。
デザイン思考は、要求管理の前段階で、問題の本質を明らかにするために役立つ。利用者観察、カスタマージャーニー、ペルソナ、プロトタイプなどを用いることで、表面的な要望ではなく、潜在的な要求を見つけやすくなる。
ただし、デザイン思考のみでは、実装可能で検証可能な要件にはならない。得られた気づきやアイデアを、要求工学により要求・要件へ展開し、システムズエンジニアリングや MBSE により構造化する必要が

ある。

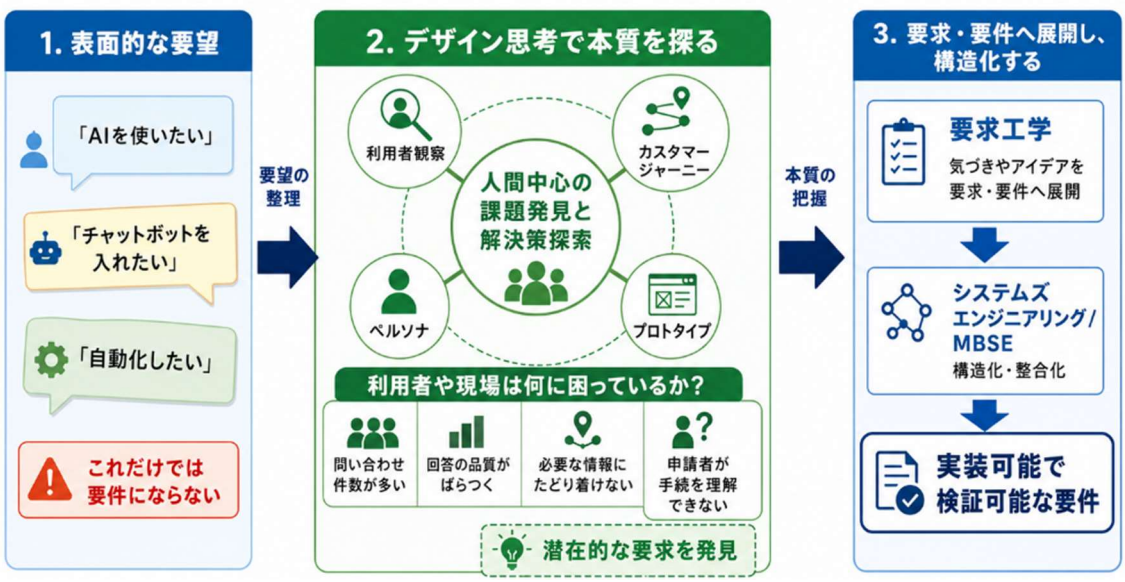


図 7 デザイン思考を用いた要求管理

3.3 ミッションエンジニアリングとの関係

ミッションエンジニアリングは、個別システムではなく、組織や社会が達成すべきミッションからシステム全体を考える方法である。要求管理において、個別機能の要求に加え、何を達成するためのシステムなのかを明確にすることは、AI の利用有無にかかわらず重要である。特に AI 時代には、AI が人の判断や業務の一部を担う場面が増えるため、AI 単体の機能や性能だけでなく、業務全体として達成すべき成果を明確にし、それに基づいて AI の適用範囲や人との役割分担、評価基準を定めることが一層重要となる。

例えば、ある企業が「問い合わせ対応 AI」を導入するとする。このとき、ミッションは「問い合わせ対応を自動化すること」ではないかもしれない。本当のミッションは、「顧客が必要な情報に早く到達できること」「担当者が高度な相談に集中できること」「回答品質を均一化すること」「顧客満足度を高めること」である可能性がある。

ミッションを明確にすると、要求・要件の優先順位が変わる。単に回答数を増やすのではなく、誤回答を避けること、問い合わせを適切にエスカレーションすること、顧客に不安を与えないこと、担当者が修正しやすいことが重要になるかもしれない。

ミッションエンジニアリングは、要求管理において、上位目的と成果を明確にする役割を果たす。要求工学は、そのミッションを実現するために、何を満たすべきかを具体化する。MBSE は、ミッション、要求・要件、機能、構造、検証の関係をモデルで接続する。

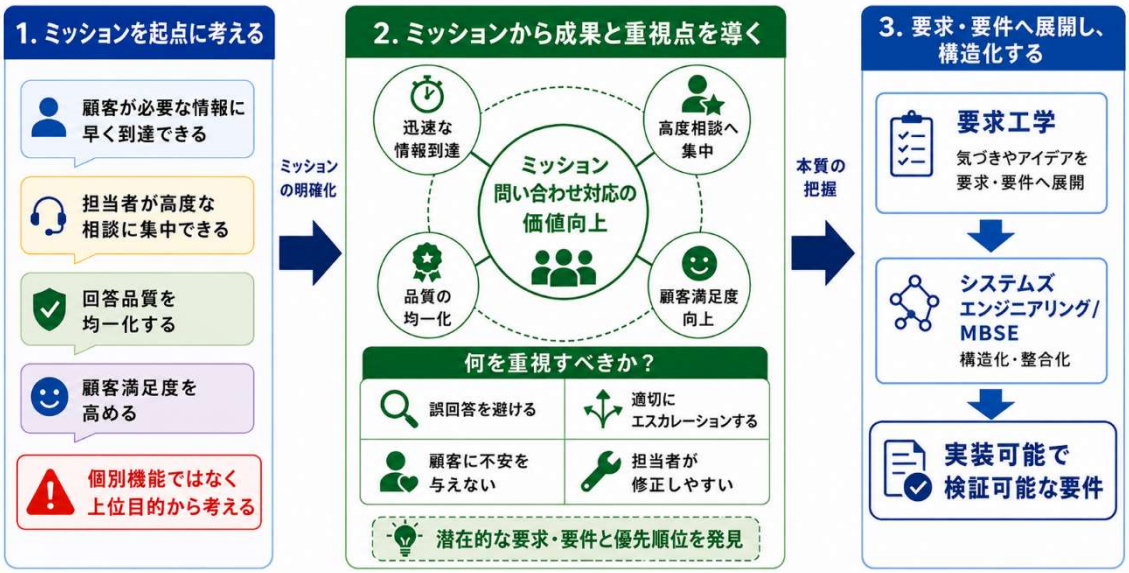


図 8 ミッションエンジニアリングを用いた要求管理

3.4 要求工学との関係

要求工学は、要求管理の中核となる方法論である。利用者や関係者のニーズを収集し、要求として分析し、仕様化し、評価し、変更を管理する。AI 時代には、この要求工学の対象が広がる。

従来の要求工学は、主にソフトウェアや情報システムを対象としてきた[13]。しかし、AI 時代には、データ要求、AI 要求、ルール要求、評価要求、運用要求も合わせて明確に扱う。法令や標準、社内規則も、単なる外部制約ではなく、業務やシステムが満たすべき要求源として扱う。

また、要求工学は、AI による支援の対象にもなる。AI は、要求候補の抽出、分類、要求・要件文の作成、矛盾検出、過不足確認、モデル生成を支援できる。ただし、AI の出力が常に正しいとは限らないことに留意して、AI が出力した結果は、人間が確認する必要がある。例えば、AI が出力した要求に対しては、目的や実装可能性、法令・標準との整合性を評価する必要がある。

3.5 システムズエンジニアリングとの関係

システムズエンジニアリングは、複雑なシステムを、ライフサイクル全体を通じて設計・統合・検証・運用する考え方である。要求管理において、個別機能だけでなく、業務、データ、ソフトウェア、クラウド、ネットワーク、セキュリティ、運用、法令、組織体制まで含めてシステム全体を考えることは、AI の利用有無にかかわらず重要であるため、この考え方は大いに役立つ。特に AI を利用する場合には、AI の振る舞いがデータや利用環境に左右されることに加え、人と AI の役割分担も業務の成果やリスクに影響するため、これら

[13] ISO/IEC/IEEE 29148:2018 [1]上では、情報システムやソフトウェアの要求を System/Software Requirements Definition として SyRS/SRS と定義している。

の要素の関係を一体として捉えるシステムズエンジニアリングの重要性が一層高まる。
AI が組み込まれたシステム（以降、AI システムと呼称する）は、単体の AI モデルのみで成立するものではない。入力データを収集する仕組み、データ品質を確認する仕組み、AI モデルを実行する基盤、結果を表示する画面、人間が確認する業務プロセス、ログを保存する仕組み、異常時に対応する運用体制が必要である。したがって、要求管理でも、システム全体を見なければならない。
システムズエンジニアリングの観点では、要求定義は、コンセプト定義、システム定義、実装、検証、妥当性確認、運用、保守、廃棄までのシステムライフサイクルの一部として位置づけられる[15]。要求・要件は、開発開始前に一度限りで定義されるものではなく、ライフサイクル全体を通じて管理される。
特に、複数の AI エージェントにより要求を実現する場合には、それぞれのエージェントの持つ機能やトレース性などを考慮することが望ましい。

3.6 MBSE との関係

MBSE（Model-Based Systems Engineering）は、文書中心ではなく、モデルを中心にシステムを定義・分析・設計・検証する考え方である。AI 時代の要求管理では、MBSE の考え方が重要になる。なぜなら、AI、データ、ソフトウェア、業務、ルール、評価がより複雑に関係するため、自然言語文書のみでは全体の整合性を管理しにくいためである。
MBSE を活用すると、要求、要件、機能、構造、振る舞い、インターフェース、検証項目をモデル上で接続できる。例えば、ある要件がどの機能で実現され、どのデータを使い、どのテストで検証されるのかを追跡できる。AI を利用する場合には、AI モデル、入力データ、評価基準、運用監視、更新条件をモデルに含めることで、AI の利用範囲とリスクを可視化できる。
MBSE は高度な専門知識を要するものと考えられがちであるが、最初から専門的なモデリング言語や専用ツールを導入する必要はない。まずは重要業務の業務フロー、主要データ、AI 利用箇所、評価項目などを図や表で可視化し、要求・要件との関係を整理することから始められる。その上で、その複雑さや管理の必要性に応じて、モデル間の関係や整合性をより体系的に管理する方法へと段階的に発展させることが現実的である。

3.7 アジャイルガバナンスとの関係

AI 時代には、制度、業務、技術、リスクが短い周期で変化する。そのため、最初に決めたルールや要求のみで長期間運用することは難しい。アジャイルガバナンスは、環境変化やリスク変化を継続的に把握し、ゴール設定、システム設計、運用、評価、改善を繰り返す考え方である。
要求管理も、このアジャイルガバナンスの一部として捉える必要がある。要求を定義し、実装し、運用し、ログ、問い合わせ、インシデント、監査結果、利用者フィードバックを分析し、必要に応じて要求を見直

---

[15] INCOSE のシステムズエンジニアリングハンドブック [24]によると、システム定義の 1 つにシステム要求定義プロセスがあるとされている。

す。この循環により、システムや AI を社会や業務の変化に合わせて更新できる。
この考え方は、DevOps とも接続する。DevOps は、開発と運用を分断せず、運用から得られるフィードバックを開発や改善に反映する考え方である。AI 時代の要求管理では、要求定義、実装、運用、監視、フィードバック、変更管理、再実装、再評価を一連のサイクルとして扱う。

3.8 各方法論の関係整理

これまで説明した各方法論を簡潔にまとめると次のようになる。

表 10 AI 時代の要求管理を支える各方法論

	主な焦点	従来からの要求管理との関係	AI 時代に特に重要となる観点
デザイン思考	利用者体験、人間中心の課題発見	そもそも解くべき問題を明確にする	AI 導入ありきではなく、解決すべき課題を見極める
ミッションエンジニアリング	組織や社会が達成すべき目的	上位目的や達成すべき成果と要求を結び付ける	AI を含む複数の手段の役割を、目的や成果から判断する
要求工学	要求の獲得、分析、仕様化、評価、管理	要望を要求、要件へ具体化し、変更管理する	AI 固有の要求・要件への対応と AI による要求管理支援
システムズエンジニアリング	複雑なシステム全体の統合	システム全体の整合性をライフサイクルにわたって確保する	AI、データ、人、業務、運用を一体として捉える
MBSE	モデルを中心にした設計・検証・一貫性管理	要求、要件、構造、振る舞い、検証をモデルで接続する	複雑な依存関係の可視化、変更影響分析、再検証
アジャイルガバナンス	変化への継続的対応	運用データやリスク変化を要求へ反映する	AI の性能・リスク・利用環境の変化に応じた継続的な評価・改善

これらの方法論を実務に適用する際には、個々の方法論を独立して実践するのではなく、それぞれの考え方を組み合わせ、得られた成果を要求管理につなげることが重要である。
例えば、問い合わせ対応 AI の導入を検討する場合、各方法論を次のように活用することが考えられる。

表 11 問い合わせ対応 AI を導入する例における各方法論の活用例

	具体的な活用例
デザイン思考	利用者や担当者へのヒアリング等を通じて、問い合わせ対応における

	課題や潜在的なニーズを明らかにする。
ミッション エンジニアリング	顧客の自己解決率向上や回答品質の改善など、組織として達成すべき成果を明確にする。
要求工学	実現すべき要求を明確にし、回答品質、有人対応への引継ぎ条件、評価基準などを具体的な要件に展開する。
システムズ エンジニアリング	業務フローや人と AI の役割分担、必要なデータやシステムの構成を検討し、AI を適用すべき範囲を見極める。
MBSE	要求・要件と業務フロー、データ、システム構成、検証項目などの関係をモデルで表現し、整合性や変更の影響を確認できるようにする。
アジャイルガバナンス	運用後の問い合わせ履歴や利用者の評価をもとに、達成状況を確認し、必要に応じて要求・要件や実現方法を見直す。

このように、利用者の課題や組織が達成すべきミッションを明確にし、要求・要件を具体化し、その実現方法や評価方法をモデルなどで関連付け、運用後のフィードバックで更新することができる。重要なのは、各方法論を決められた順序で適用することではなく、必要な視点を組み合わせながら、目的、要求・要件、実現手段、評価結果の一貫性を維持することである。これを組織能力として定着させることが、変化に強い企業や社会を作る基盤となる。AI 時代には、この重要性が一層高まる。

第 4 章　グローバル経営時代の要求管理

4.1 経営・事業変革において要求管理が重要になる理由

企業や行政が変革を進める際には、経営戦略や政策上の目的を掲げるだけでは十分ではない。その目的を、実際の業務、組織、データ、AI、ソフトウェア、ルール、取引関係等が満たすべき事項へ具体化し、関係者間で共有し、実現状況を継続的に確認する必要がある。要求管理は、このように、実現したい価値と具体的な実行を接続する役割を持つ。

このとき重要なのは、既存の業務やシステムを所与のものとして、そのまま要求・要件へ置き換えないことである。事業上・業務上の課題や実現したい価値を分析し、必要に応じて業務そのものを見直した上で、何を実現すべきかを明確にする必要がある。このような検討には、ビジネス分析や BPR（Business Process Re-engineering）[17]等の考え方も活用できる。

---

[17] ビジネス分析は、事業や業務上のニーズ、課題、価値、ステークホルダー等を明らかにし、変革の方向性を検討するための活動である。また、既存の業務プロセスそのものを抜本的に見直す場合には、BPR 等のアプローチが用いられる。

第 2 章や第 3 章でも触れているとおり、要求管理は、これらの活動と分断されたものではない。ビジネス分析や BPR によって明らかになった目的や変革内容を要求として具体化する一方、要求分析の過程か

特に、複数の組織、企業、国・地域が関係する場合には、目的、責任、業務、データ、ルール等の前提が異なるため、この役割はより重要となる。グローバル経営は、その典型的な場面の一つである。

企業活動は、国内の一企業、一拠点、一システムの中のみで完結しなくなっている。製品、サービス、データ、業務プロセス、サプライチェーン、顧客対応、規制対応は、国境を越えてつながっている。製造業では、部品調達、生産、物流、販売、保守が複数国にまたがる。サービス業では、顧客接点、データ処理、外部委託、クラウド利用、問い合わせ対応が国際的に分散する。ソフトウェアや AI を利用する企業では、開発拠点、データ保管場所、モデル提供者、API 提供者、運用委託先が世界中に存在する。

このようなグローバル経営の時代において、要求管理は、単なるシステム開発のための活動ではない。経営戦略や事業方針を具体的な要求・要件につなげ、その実現を支える経営・事業活動の一部である。要求定義書は、国、地域、組織、部門、取引先の間で、何を実現するのか、どの品質を満たすのか、どのデータを使うのか、どのルールに従うのかを共有するための共通言語となる。

グローバルな取引では、関係者の文化、言語、制度、商習慣、技術水準が異なる。その中で、口頭の合意や暗黙知に依存すると、誤解や手戻りが発生しやすい。例えば、「品質を確保する」「適切に管理する」「迅速に対応する」「安全に利用する」といった表現は、一見分かりやすいが、国や組織によって解釈が異なる可能性がある。要求定義では、これらを、実装可能で、評価可能で、責任範囲が明確な形に落とし込む必要がある。

そのため、要求管理をシステム担当者のみに任せるのではなく、経営層や事業責任者が、実現すべき事業価値や優先順位、許容するリスク、組織間の役割分担などの判断に主体的に関与することが重要である。こうした経営・事業上の判断を、業務部門やシステム担当者と協働して具体的な要求・要件に落とし込み、関係者間で共有・管理していく必要がある。

経営・事業変革、とりわけ多様な組織や国・地域が関係するグローバル経営における要求定義の意義は、次の四点に整理できる。

1. 関係者間の共通理解を醸成する

   異なる国、言語、部門、企業が関わる場合、要求定義は、目的、範囲、責任、制約、成果物を明確にするための共通基盤となる。

2. 迅速なサプライチェーン構築を可能にする

   部品、サービス、データ、システム、AI モデルを国際的に組み合わせるには、インターフェース、品質基準、データ形式、セキュリティ、監査要件を事前に明確し、相互に合意する必要があり、要求定義はそれに役立つ。

3. M&A や業務統合の基準になる

   企業を買収・統合する際には、相手企業の業務、システム、データ、ルール、リスク、内部統制が、自社の基準とどこまで整合しているかを確認する必要がある。要件が適切に定義され、可視化されていれば、統合すべき点、残すべき点、変更すべき点を判断しやすくなる。

ら、業務や事業の前提そのものを見直す必要が明らかになることもある。相互に行き来しながら進めることが重要である。

4. 文化的背景や制度の違いを吸収する

国や地域によって、労働慣行、個人情報保護、契約、商習慣、リスク許容度、説明責任の考え方は異なる。要求定義は、その違いを可視化し、違いを無理に一つの要求に押し込むのではなく、共通化すべき部分とローカルに適応すべき部分を分けるための手段となる。

4.2 節から 4.5 節にかけてこの 4 点について詳細に説明する。

4.2 グローバルな取引を例に見る関係者間の共通理解

グローバルな取引では、契約書、仕様書、SLA、品質基準、セキュリティ要件、データ利用条件など、多くの文書が作成される。しかし、それらが相互に整合していなければ、実務では混乱が生じる。契約では「高品質なサービスを提供する」と書かれていても、要求定義で品質指標、測定方法、報告方法、改善手順が明確でなければ、トラブル時に解釈が分かれる。

要求管理は、グローバルな取引における共通理解を具体化する役割を持つ。特に、次のような項目を明確にすることが重要である。

表 12 グローバルな取引において重要な要求管理項目の例

	明確にすべき内容
目的	何のために取引・連携・開発を行うのか
対象範囲	どの業務、拠点、システム、データが対象か
役割分担	誰が何を担当し、誰が責任を持つか
品質基準	何をもって品質を満たしたと判断するか
データ要求	データ項目、定義、形式、品質、利用条件
セキュリティ要件	アクセス制御、ログ、暗号化、事故対応
インターフェース	API、データ連携、ファイル形式、更新頻度
評価方法	テスト、受入条件、監査、報告方法
変更管理	仕様変更、法令変更、障害時の対応
ローカル差異	国・地域ごとの例外や追加要求

このように要求・要件を明確にしておけば、発注者、受注者、委託先、現地法人、外部サービス提供者が、同じ前提で作業を進めやすくなる。特に、AI やデータを扱う場合には、データの意味、利用条件、保存場所、再利用可否、説明責任を明確にしておかなければ、後から大きな問題になりやすい。

グローバルな取引では、言語の違いも重要である。英語で書かれた要求が、日本語に訳されたときに意味がずれることがある。また、日本語の「適切に」「必要に応じて」「速やかに」といった表現は、英語や他言語に置き換える際に曖昧さが増す場合がある。したがって、重要な要求は、自然言語に加え、表、定義、モデル、受入条件、測定指標を併用して表現することが望ましい。

4.3 迅速なサプライチェーン構築と要求管理

グローバル経営では、サプライチェーンを迅速に構築し、必要に応じて組み替える能力が競争力になる。部品、素材、物流、クラウド、ソフトウェア、AI、データサービスを、複数の国や企業から組み合わせる場合、各要素の要件が明確でなければ、接続や切替に時間がかかる。

例えば、新しい海外サプライヤーを追加する場合、価格や納期の他にも、品質管理、トレーサビリティ、セキュリティ、データ連携、監査対応、環境・人権対応などの要件を確認する必要がある。これらが文書化され、評価可能な形になっていれば、候補企業の比較、契約交渉、システム連携、監査準備が速くなる。

サプライチェーンにおける要求管理では、次のような観点が重要である。

表 13 サプライチェーンにおける要求管理で重要な観点の例

	要求管理上の意味
品質	製品・サービスが満たすべき基準を明確にする
納期	リードタイム、遅延時対応、報告義務を定義する
データ連携	注文、在庫、出荷、品質情報の形式と頻度を定義する
トレーサビリティ	部品、材料、工程、出荷履歴を追跡可能にする
セキュリティ	情報管理、アクセス権、委託先管理を定義する
レジリエンス	災害、紛争、規制変更時の代替手段を定義する
コンプライアンス	輸出管理、個人情報、環境、人権等への対応を定義する
変更管理	仕様変更、供給停止、代替品利用時の手順を定義する

AI 時代には、サプライチェーンの監視や最適化にも AI が使われる。需要予測、在庫最適化、異常検知、調達先評価、リスク予測などである。その場合、AI が利用するデータの品質、出典、更新頻度、利用条件、説明可能性も要件として定義する必要がある。

迅速なサプライチェーン構築のためには、要件を一から毎回作るのではなく、共通テンプレート、標準データ項目、共通 API、評価基準、チェックリストを整備しておくことが重要である。要件が標準化されていれば、新しい取引先やサービスを追加する際にも、比較、審査、接続、監査を迅速に行える。

4.4 M&A における要求定義の役割

M&A では、財務、法務、人事、事業、顧客、技術、システム、データ、セキュリティ、内部統制など、多面的な確認が行われる。AI 時代には、これに加えて、AI 利用状況、データ品質、モデル管理、外部サービス依存、サイバーリスク、規制対応、社内ルールの整備状況も重要になる。

要求定義は、M&A において二つの意味を持つ。

1. デューデリジェンスの基準として機能する。

買収対象企業の業務、システム、データ、AI、セキュリティ、コンプライアンスが、自社の要求水準を満たしているかを確認するためには、基準となる要件が必要である。例えば、顧客データ管理、アクセス制御、ログ保存、AI 利用ルール、外部委託管理、インシデント対応について、自社が求める要件が明確であれば、対象企業との差分を把握しやすい。

2. 統合計画の基準として機能する。

M&A 後には、業務プロセス、基幹システム、データ、社内規則、セキュリティ基準、AI 利用方針を統合する必要がある。このとき、どこを共通化し、どこを地域・事業特性に応じて残すかを判断するためにも、要求定義が必要になる。

M&A で確認すべき要求の例は、次のとおりである。

表 14 M&A で領域ごとに確認すべき要求の例

	確認すべき要求
業務	主要業務プロセス、承認、例外処理、責任分担
システム	基幹システム、外部サービス、API、運用体制
データ	データ定義、品質、正本、アクセス権、保存期間
AI	利用目的、モデル管理、入力データ、評価、監視
セキュリティ	認証、権限、ログ、脆弱性管理、事故対応
コンプライアンス	法令、規制、契約、業界標準、監査証跡
社内ルール	規程、ガイドライン、マニュアル、教育、改定履歴
変更管理	システム変更、ルール変更、データ変更の管理方法
ベンダー管理	委託先、SLA、契約、再委託、データ利用条件

M&A における要求定義の目的は、対象企業を自社のやり方に一方的に合わせることではない。重要なのは、統合後に実現したい経営目的を明確にし、共通化すべき要件、ローカルに残すべき要件、段階的に移行すべき要件を整理することである。特にグローバル M&A では、各国の制度や文化を無視して一律統合を進めると、現場の反発や業務停滞が起きる。要求定義は、共通基盤と地域適応のバランスを取るための道具である。

4.5 文化的背景や制度の違いへの対応

グローバル経営では、文化的背景や制度の違いを前提に要求定義を行う必要がある。国や地域によって、意思決定の方法、責任の考え方、契約への期待、リスク許容度、個人情報保護、労働慣行、顧客対応、監査文化が異なるためである。

例えば、ある国では詳細な手順を文書化し、担当者がその手順に従うことが重視される。一方、別の国では、現場の裁量や判断が重視されることがある。また、個人情報や従業員データの扱いについても、国ごとに法制度や社会的期待が異なる。顧客対応においても、即時回答を重視する文化もあれば、慎重

で正式な回答を重視する文化もある。
このような違いを無視して、単一の要件をすべての国・地域に適用すると、現場に合わない仕組みになる可能性がある。一方で、すべてをローカルに任せると、グループ全体の統制、品質、セキュリティ、データ連携が難しくなる。
したがって、グローバル要求定義では、共通要件とローカル要件を分けることが重要である。

表 15 グローバル経営を見据えた要件区分の例

	内容
グローバル共通要件	グループ全体で統一すべき目的、品質、セキュリティ、データ、監査、倫理
リージョン要件	地域の規制、商習慣、言語、顧客対応に応じた要件
ローカル要件	国・拠点・事業固有の運用、例外、手順
例外要件	一時的または限定的に認める例外とその管理方法

この整理により、グローバル統一とローカル適応を両立しやすくなる。重要なのは、ローカル差異を「例外」として曖昧に扱うのではなく、理由、適用範囲、責任者、見直し時期を明確にした要件として管理することである。

4.6 言語の違いを吸収する要求定義

グローバル経営では、言語の違いが要求定義の品質に大きな影響を与える。要求は、英語、日本語、現地語の間で翻訳されることが多い。しかし、翻訳によって意味がずれると、実装、契約、監査、運用に影響する。
特に注意すべきなのは、曖昧な表現である。例えば、「速やかに」「必要に応じて」「適切に」「合理的な範囲で」「原則として」といった表現は、文脈によって解釈が変わる。日本語では自然に読めても、英語や他言語に訳すと、責任範囲や期限が不明確になることがある。
言語の違いを吸収するためには、次の工夫が有効である。

1. 用語集を作る
   重要な用語について、英語、日本語、現地語の対応を整理し、定義を明確にする。特に、顧客、利用者、事業所、担当者、承認者、個人情報、機密情報、AI 出力、ログ、インシデントなどは、定義のずれが生じやすい。
2. 要求文を短く、一意に書く
   一つの要求文に複数の条件や例外を詰め込むと、翻訳時に意味が崩れやすい。要求文は、主語、動詞、対象、条件、評価方法を明確にして書くことが望ましい。
3. 自然言語のみに依存しない
   表、業務フロー、データモデル、ルール表、受入条件、テストケースを併用すれば、言語の違いによる誤解を減らせる。

4. AI 翻訳や生成 AI を使う場合にも、専門家レビューを行う

AI 翻訳は便利だが、法令、契約、技術、業務用語の微妙な差異を誤る可能性がある。重要な要求は、現地の法務、業務担当、技術担当が確認する必要がある。

4.7 グローバル要求定義で追加すべき観点

グローバル経営時代の要求定義では、国内プロジェクト以上に、次の観点をより重点的に確認することが望ましい。

表 16 グローバル要求定義で追加すべき観点

	確認すべき内容
法制度	各国・地域の法令、規制、許認可、報告義務
データ越境	データの保存場所、移転、アクセス、再利用条件
言語	用語定義、翻訳、現地語対応、表示言語
文化	顧客対応、承認、責任、リスク許容度
契約	責任分担、SLA、知財、再委託、データ利用
セキュリティ	各国基準、グループ基準、委託先管理
監査	証跡、報告形式、内部統制、外部監査対応
サプライチェーン	取引先、再委託、代替手段、地政学リスク
標準化	共通 API、共通データモデル、共通評価基準
ローカル適応	地域差、例外、段階的移行

これらを要求定義の初期段階から確認しておけば、後から「この国では使えない」「現地規制に合わない」「データを移転できない」「現地語対応が必要だった」といった手戻りを減らすことが期待できる。

4.8 グローバル経営時代の要求管理の進め方

グローバルな要求管理では、最初からすべてを一律に決めようとすると、現場の実態に合わなくなる。一方で、各国・各拠点に任せきりにすると、グループ全体の統制やデータ連携が難しくなる。そのため、共通化とローカル適応を分けて進めることが重要である。

進め方の例は、次のとおりである。

1. グローバル共通の目的と原則を定義する

経営層や事業責任者が、経営戦略や事業方針に基づき、グローバルで実現すべき価値や優先順位、共通化すべき範囲、現地に委ねる判断の範囲などを明確にする。例えば、顧客データを安全に扱う、AI 利用時には人間の関与を明確にする、重要業務では監査証跡を残す、といった基本原則を明確にする。

2. 共通化すべき要件を定義する

   データ定義、セキュリティ、認証、ログ、API、品質基準、AI 利用ルールなど、グループ全体で統一した方がよい要件を整理する。

3. ローカル要件を収集する

   各国の法令、商習慣、言語、顧客対応、労働慣行、現地システムの制約を確認する。

4. 共通要件とローカル要件の差分を管理する

   差分は、単なる例外として放置せず、理由、適用範囲、責任者、見直し時期を明確にする。特に、事業方針やグループ全体の統制に影響する差分については、経営層や事業責任者が必要な判断を行う。

5. モデルやテンプレートで共有する

   業務フロー、データモデル、ルール表、トレーサビリティ表を使い、国や言語を越えて確認できる形にする。

6. 運用後のフィードバックを取り込む

   現地での問い合わせ、監査指摘、事故、法令変更、顧客反応を分析し、グローバル共通要件とローカル要件を見直す。

このように、グローバル経営時代の要求管理は、標準化と多様性の調整である。すべてを一律にそろえることでも、すべてを現地任せにすることでもない。経営層や事業責任者が経営・事業上の方針や判断基準を示し、それに基づいて共通化すべき部分を明確にするとともに、地域差を要件として管理し、変化に応じて更新できる仕組みを作ることが重要である。この意味で、要求管理は、システム開発の枠を超えて、経営戦略や事業方針の実現を支える活動といえる。

本章では、グローバル経営を中心として、経営・事業変革において要求管理が果たす役割を示した。ここで示した考え方は、グローバルな取引や M&A に限定されるものではない。新規事業、業務改革、AI 導入、制度変更等においても、目的を明確にし、ステークホルダーを特定し、要求を獲得・分析・評価し、実現手段へつなげ、運用後の変化を再び要求へ反映するという基本的な考え方は共通する。

この一連の活動を実務でどのように進めるかについては、第 7 章「AI 時代の要求管理プロセス」で整理する。

第 2 部　AI 時代に要求管理はどう変わるか

AI の急速な進展により、要求管理を取り巻く環境は大きく変化している。これまでの要求定義は、主に人間が利用者や業務部門から要望を聞き取り、文書として整理し、システム開発へ引き渡す活動として行われてきた。しかし、AI 時代には、要求定義そのものが AI によって支援されるようになる。同時に、AI を組み込んだシステムの要件を定義する必要も生じる。

つまり、AI 時代の要求管理には二つの側面がある。第一に、AI を使って要求定義を高度化することである。AI は、問い合わせ、議事録、マニュアル、ログ、クレーム、事故報告、既存仕様書などを分析し、要求候補の抽出、分類、要約、矛盾確認、モデル化を支援できる。第二に、AI を使うシステムそのものの要件を適切に定義することである。AI は確率的に振る舞い、学習データや利用環境の影響を受け、運用後も変化する。そのため、通常のソフトウェア以上に、目的、利用範囲、データ、評価、監視、人間の関与、更新、説明責任を明確にしなければならない。

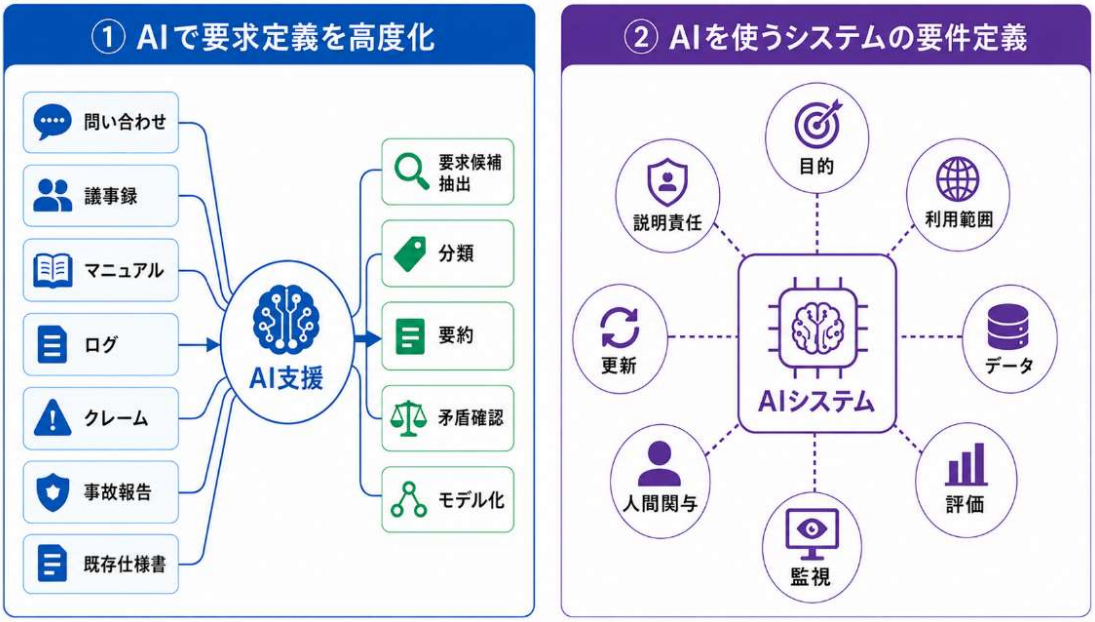


図 9　AI 時代の要求管理の 2 つの側面

第 2 部では、この二つの観点から、AI 時代に要求管理がどのように変わるのかを整理する。まず、第 1 部で説明した要求工学の基本を前提としつつ、前述の二つの側面に関して AI が要求管理の前提をどう変えていくか（第 5 章）に焦点を当てる。その後、AI が要求管理をどう支援するか（第 6 章）、AI 時代の要求管理プロセスについて、プロセス全体をどう進めるか（第 7 章）と AI システム特有の要求獲得・分析・仕様化の変化（第 8 章）に焦点を当てる。

第 5 章　AI が変える要求管理の前提

5.1 変化速度が上がる時代の要求管理

AI 時代の最も大きな特徴は、変化の速度が上がることである。技術は急速に進化し、新しいサービスが次々に登場し、業務や社会の仕組みも短期間で変化する。従来は、制度、標準、業務、システムは年単位で見直されることが多かった。しかし、AI を中心とするデジタル技術の進展により、四半期単位、場合によってはそれより短い周期で見直しが必要になる。

この変化は、要求管理に大きな影響を与える。これまでのように、最初に要件を詳細に決め、それに基づいてシステムを開発し、完成後に運用するという一方向の流れのみでは対応しにくくなる。AI モデルは更新され、利用データは変化し、利用者の行動も変わる。法令、標準、ガイドライン、社内規則も変化する。したがって、要求管理は、開発開始前に一度限りで行う活動ではなく、運用後も継続的に見直される活動になる。

変化速度が上がる時代には、「正しい要件を最初にすべて決める」ことよりも、「要件を継続的に検証し、必要に応じて更新できる状態にする」ことが重要になる。もちろん、最初の要求定義が不要になるわけではない。また、最初の要求定義が不十分なまま開発を開始し、後から修正すればよいという意味でもない。特に、業務の目的、評価基準、法令・社内規則、例外処理など、運用開始前に明確化できる事項については、関係部門が十分に整理し、要求・要件として定義しなければ、運用後の改善も場当たり的になる。一方で、新しいプロダクトやサービス、探索的な AI 活用など、実際の利用や市場から得られる情報によって初めて妥当性を十分に確認できる事項については、目的、仮説、評価基準及び見直し条件をあらかじめ定め、運用から得られる知見に基づいて要求・要件を発展させていくことが重要である。

ここで重要なのは、変化に応じて要求・要件を更新するための判断軸として、上位の目的・ゴールを明確にしておくことである。要求・要件を継続的に見直す場合であっても、何を実現するための要求・要件なのかが不明確であれば、目的と手段が混在し、個々の改善が全体としてどの価値につながるのか分からなくなるおそれがある。

また、変更の過程で上位の目的・ゴールそのものを見直す必要が生じた場合には、個別の要求・要件をそのまま修正し続けるのではなく、一度最上流に戻る必要がある。実現すべき価値や成功基準を再確認し、それを実現するビジネス・業務モデル、機能の分担、必要なデータ、評価方法等を再整理した上で、要求・要件を改めて導出する。このように、継続的な変更と、必要に応じて上流から再構成する活動を組み合わせることが、AI 時代の要求管理では重要となる。

以上を踏まえて、AI 時代の要求管理では、次のような発想が求められる。

表 17　AI 時代に求められる要求管理の考え方

従来によく見られた 固定的な考え方	AI 時代に求められる ライフサイクルを通じた考え方

要件は開発前に確定する	要件は目的との整合を保ちながら運用後も更新される （ただし、開発前に必要な水準まで要件を定義・合意する）
文書化して引き渡す	モデル化し、共有し、更新する
機能を定義する	目的から、機能、データ、評価、運用まで一貫して定義する
開発完了が最終目標	価値実現と継続改善を行う
変更は例外	変更を前提に目的との整合を確認しながら管理する （ただし、最初の要求定義が不十分でもよいという意味ではない）
人間がすべて読んで理解する	人間と AI・ツールが利用できる形にする

この変化に対応するためには、要求を「静的な文書」ではなく「生きた管理対象」として扱う必要がある。ただし、要求を継続的に管理することは、人がすべての変更を都度確認し、文書を手作業で更新し続けることを意味しない。更新の頻度が高まるほど、要求・要件をモデルや構造化された情報として管理し、AI やツールを活用して、要求間の整合性確認、変更影響の分析、関連する設計・実装・テストの特定、評価基準や制約への適合状況の監視を「支援・自動化」することが重要になる。
そのため、AI 時代の要求管理では、要求、要件、業務、データ、設計、実装、テスト、運用ログ等の関係を明示し、変更が生じた場合に、その影響をライフサイクル全体へ追跡できる状態を整えておく必要がある。これにより、要求の継続的な見直しと、変更に伴う再検証・再評価を一体として進めやすくなる。

5.2 AI システムは変化し続ける

従来のソフトウェアは、設計された機能を比較的安定して実行することを前提としていた。もちろん、ソフトウェアにも改修や機能追加はあるが、基本的には仕様に基づいて動作する。一方、AI を含むシステムは、データ、モデル、利用環境、利用者行動、外部サービスの変化により、性能やリスクが変化する。
例えば、問い合わせ対応 AI を考える。導入時には十分な回答精度があったとしても、新しい商品、制度改正、新しい問い合わせ傾向、社会情勢の変化により、回答の妥当性が低下する可能性がある。AI モデルを更新すれば、過去には正しく回答していた質問に対して、異なる回答を出す可能性もある。生成 AI を利用する場合は、出力の揺らぎ、ハルシネーション、機密情報の混入、著作権上の懸念、誤用のリスクも考慮する必要がある。
したがって、AI を含むシステムでは、要求定義の対象が広がる。例えば、単に「A 機能を実装すること」のような要件では不十分であり、次のような項目を AI 要求として定義する必要がある。

表 18 AI 要求で定義すべき項目の例

	内容
AI 利用目的	AI を何のために使うのか
利用範囲	どの業務・場面で使ってよいか
禁止範囲	どの用途では使ってはいけないか

入力データ	どのデータを使うか、品質はどう確保するか
出力要件	どのような形式・根拠・制約で出力するか
人間の関与	誰が確認・承認・差戻しを行うか
評価指標	精度、安全性、公平性、説明可能性をどう測るか
監視要件	運用後の性能劣化や異常をどう検知するか
更新要件	モデルやデータをいつ、どう更新するか
証跡要件	ログ、根拠、判断履歴をどう残すか
インシデント対応	誤判定や事故が起きたときにどう対応するか

AI 時代の要求定義は、機能要求に加え、データ要求、AI 要求、ルール要求、評価要求、運用要求、説明責任要求、変更管理要求を含むものになる。これらの要求は、構造化されパラメータ化されることで変更容易性が高まる。変更発生時に当該部分の変更により迅速に対応が可能になる。

5.3 AI システムにおける不確実性を要件として定義する

従来のソフトウェア要件定義では、多くの場合、システムは決定論的に動作することを前提としてきた。すなわち、同じ入力に対して同じ処理を行い、同じ出力を返すことが期待される。もちろん、例外処理や障害は存在するが、基本的には、仕様で定めた条件を満たせば、システムの振る舞いは予測可能である。一方、AI システム、とりわけ機械学習や生成 AI を用いるシステムは、確率的に振る舞う。入力が似ていても出力が変わることがある。学習データ、モデル、プロンプト、参照文書、利用環境、外部サービスの更新により、出力品質やリスクが変化することもある。生成 AI では、事実と異なる情報をもっともらしく出力するハルシネーションが発生する可能性もある。また、学習時に想定していない入力、すなわち Out-of-Distribution、略して OOD と呼ばれる予期せぬ入力に対して、AI が不安定な出力を行う場合もある。

したがって、AI 時代の要件定義では、「AI は正しく分類すること」「精度 90%以上で回答すること」といった単純な記述のみでは不十分である。AI の不確実性を前提に、どの程度の確率で、どの範囲で、どの条件において、どのような振る舞いを期待するのかを定義する必要がある。さらに、AI が十分な確信を持てない場合、誤りが疑われる場合、想定外の入力が与えられた場合に、どのように安全側に倒すのかを要件として定義しなければならない。

AI システムの不確実性に関する要件では、少なくとも次の観点を確認する。

表 19 AI システムの不確実性に関する要件での確認観点

	確認すべき内容
精度	どの評価データで、どの指標を、どの水準以上にするか
信頼区間	評価結果のばらつきや不確かさをどう扱うか
確信度	AI が自分の出力にどの程度自信を持っているかをどう扱うか

誤分類	誤りが発生した場合、どの種類の誤りを特に避けるべきか
ハルシネーション	事実と異なる出力が発生した場合、どう検知・抑制・対応するか
OOD 対応	想定外の入力や分布外データに対してどう振る舞うか
フェイルセーフ	AI が迷った場合、安全側に倒す仕組みがあるか
人間へのエスカレーション	AI のみで判断せず、人間に確認させる条件は何か
継続的評価	運用後に性能劣化やリスク変化をどう確認するか
再評価条件	モデル、データ、プロンプト、業務条件が変わった場合に再評価するか

例えば、問い合わせ分類 AI を導入する場合、「問い合わせを正しく分類すること」のみでは要件として不十分である。従来のルールベース型のシステムでは、「件名に『請求書』を含む場合は経理部門に振り分ける」といったように、あらかじめ定めた条件とそれに対応する処理を要件として記述できる。一方、AI を用いて問い合わせの内容から分類する場合は、表現の揺れや想定外の入力などによって誤分類が生じる可能性がある。そのため、対象とする問い合わせの範囲や分類ごとの評価指標・達成水準に加え、AI が判断に迷った場合や誤分類の影響が大きい場合の対応も定める必要がある。具体的には、次のように要件化することが考えられる。

- 問い合わせ分類 AI は、評価用データセットに対して、主要分類ごとの再現率が 90%以上となることを月次で評価すること
- 分類の確信度が一定値を下回る場合、AI は自動分類せず、担当者確認に回すこと
- 苦情、障害報告、法的問い合わせなど高リスク分類については、誤分類を防ぐため、人間による確認を必須とすること
- AI が分類不能と判断した問い合わせは、「その他」に自動分類するのではなく、未分類として管理し、分類ルールまたは学習データの改善対象とすること
- 運用中に新しい問い合わせ傾向が増加した場合、分類体系の見直しを行うこと

このように記述すると、AI の性能を単一の平均精度のみで評価するのではなく、分類ごとのリスク、確信度、人間の関与、運用後の改善まで含めて管理できる。

特に重要なのが、「AI が迷ったときにどう振る舞うか」を要件として定義することである。人間であれば、自信がない場合に上司や専門家に確認する、回答を保留する、追加情報を求めるといった対応ができる。AI システムでも同じように、確信度が低い場合、不明確な入力を受けた場合、高リスクな判断を求められた場合には、無理に回答や判断を行うのではなく、安全な動作を選ぶ必要がある。

AI が迷ったときの振る舞いは、次のように要件化できる。

表 20 AI が迷った時の振る舞いに対する要件化の例

	要件化の例
確信度が低い	自動処理せず、人間の確認に回す
入力が不十分	追加情報を求める
入力が想定外	処理を停止し、例外として記録する

高リスク業務	AI 出力を参考情報に限定し、最終判断は人間が行う
根拠が不明	回答を生成せず、参照資料不足を表示する
複数の解釈がある	候補と前提条件を提示し、人間に選択させる
ハルシネーションの可能性	原資料との照合を必須にする
セキュリティ上の疑義	応答を拒否し、ログを記録する

例えば、生成 AI を使った社内問い合わせシステムでは、次のような要件が考えられる。

- AI は、承認済み文書に根拠を確認できない場合、推測で回答してはならない
- AI は、回答根拠となる社内規則、マニュアル、FAQ の名称と該当箇所を表示すること
- AI は、法務判断、人事評価、懲戒、契約解除、個人情報開示に関する質問については、一般的な案内にとどめ、専門部署への確認を促すこと
- AI は、利用者の質問が曖昧な場合、回答を生成する前に追加質問を行うこと
- AI は、回答の確信度が低い場合、その旨を表示し、人間による確認を求めること

また、AI システムでは、平均精度に加え、評価結果の不確かさも考慮することが望ましい。評価データが少ない場合、ある時点で高い精度が出ていても、実運用で同じ性能が出るとは限らないためである。したがって、可能な場合には、単一の数値に加え、評価データの範囲、信頼区間、分類別性能、重要ケースでの誤り率、運用後の再評価頻度を定義することが望ましい。

例えば、次のような要件文が考えられる。

- AI モデルの性能評価では、全体精度に加え、主要カテゴリ別の精度、再現率、誤分類率を確認すること
- 評価結果を報告する際には、評価データの件数、対象期間、データの偏り、信頼区間またはばらつきの情報を併記すること
- 利用環境、入力データ、業務ルール、モデル、プロンプトを変更した場合、再評価を行うこと
- 運用中の実績値が評価時の性能を一定以上下回った場合、改善要求を起票すること

OOD 対応も重要である。AI は、学習時や評価時に想定していない入力を受けると、不適切な出力を行う可能性がある。例えば、社内規則に関する AI に対して、未登録の新制度、他国の制度、古い規則、誤った文書、悪意あるプロンプトが入力される場合がある。このような場合に、AI が自信ありげに回答してしまうと、利用者は誤った判断をしてしまう。

そのため、OOD 対応では、次に関する要件を定義する。

- 想定する入力範囲
- 想定外の入力を検知する方法
- 想定外の場合に回答を控える条件
- 追加確認や人間へのエスカレーション方法
- 想定外入力のログの保存と、それを改善対象とすること
- 新しい入力傾向が増えた場合に要件、データ、モデルを見直すこと

AI システムにおける不確実性の仕様化は、AI の弱点を認めるためのものではない。むしろ、不確実性を

前提に、安全で信頼できる使い方を設計するためのものである。AI が常に正しいと仮定するのではなく、AI が誤る場合、迷う場合、想定外に遭遇する場合を要件として定義することで、AI を実務に安全に組み込むことができる。
これまでの話をまとめると、AI 時代の要件定義では、「AI が何をできるか」に加え、「AI ができないとき、迷ったとき、誤ったときにどう振る舞うか」を明確にする必要がある。これにより、AI システムは、単に高性能なものではなく、運用上信頼でき、説明可能で、改善可能なものになる。

5.4 プロンプトエンジニアリングも要求工学の一部である

生成 AI の利用が広がるにつれて、プロンプトエンジニアリングという言葉が使われるようになった。プロンプトとは、AI に対する指示である。AI に何をしてほしいのか、どの情報を使うのか、どの形式で出力するのか、何を禁止するのかを記述する[18]。
この意味で、プロンプトエンジニアリングは、要求工学の一部と考えることができる。なぜなら、プロンプトは AI に対する要求を記述しているためである。
例えば、単に「議事録を要約してください」と指示するのみでは、何を重視するのかが分からない。意思決定事項を抽出したいのか、未決事項を整理したいのか、ToDo を担当者別に並べたいのか、リスクや懸念点を抽出したいのかによって、必要なプロンプトは異なる。
良いプロンプトには、次のような要素が含まれる。

表 21 良いプロンプトに含まれる要素

	内容
目的	何のために AI を使うのか
役割	AI にどの立場で作業させるのか
入力	どの情報を参照させるのか
制約	何をしてはいけないのか
出力形式	表、箇条書き、文章、JSON 等

[18] AI に適切な指示を与えるためには、プロンプトの記述だけでなく、AI が参照する情報やその背景となる文脈（コンテキスト）を適切に設計・管理することも重要である。このような考え方は、コンテキストエンジニアリングと呼ばれる。
例えば、AI を用いて要求候補を抽出・分析する場合、単に会議録や要望一覧を入力するだけでなく、プロジェクトの目的、関係者間で共有したコンセプト、業務上の制約、既存の要求・要件などを、必要に応じて参照できるようにすることが有効である。ただし、情報を多く与えればよいわけではなく、情報の正確性や最新性、利用目的との関連性にも配慮する必要がある。
このように、要求管理で明確化した目的や背景情報を AI にも適切に伝えることは、AI による要求管理支援の品質を高める上でも重要となる。

評価基準	何を満たせばよい出力とするのか
確認事項	不明点、前提、リスクを明示させる
根拠	出典や判断理由を示させる

プロンプトは一回限りの文章ではなく、要求仕様の一部として管理することが望ましい。ただし、プロンプトが要求・要件のすべてを代替するわけではない。プロンプトは、システムや AI に求める要件のうち、AI に実行させる内容を表現する手段の一つである。業務上の目的、人と AI の役割分担、データ、セキュリティ、法令・ルール、評価基準、監視、責任分界などは、プロンプトのみではなく、システム全体の要件として管理する必要がある。

また、業務で AI を利用する場合、プロンプト、参照データ、出力形式、レビュー手順、利用制限、ログ、評価基準をセットで管理すると良い。これはシステム開発[19]に限った話ではない。既製の生成 AI を業務で直接利用する場合でも、プロンプトは AI の出力内容や品質、リスクを左右する。例えば、同じ業務であっても、使用するプロンプトや参照データ、出力条件が変われば、得られる結果も変わり得る。そのため、繰り返し利用するプロンプトは、業務手順やテンプレートと同様に、組織として共有し、評価し、必要に応じて更新できる状態にしておくことが重要である。

したがって、AI 時代の要求管理では、プロンプトも管理対象になる。プロンプトの変更が出力品質やリスクに影響する場合、その変更履歴、承認、評価結果を残す必要がある。

5.5 AI の社会的受容性と倫理的影響を要件として扱う

AI 時代の要求定義では、組織内の業務効率、コスト削減、品質向上のみを確認すればよいわけではない。AI システムは、利用者、顧客、従業員、取引先、地域社会、将来の利用者など、広い範囲のステークホルダーに影響を与える。そのため、システムに AI を組み込む場合には、そのシステムが社会的に受け入れられるものか、倫理的に問題がないか、特定の人々に不利益を与えないかを、要求分析の段階で確認する必要がある。

従来の要求定義では、社会的影響や倫理的影響は、理念や方針として扱われることが多かった。しかし、AI 時代には、それらを抽象的なスローガンにとどめてはならない。公平性、透明性、説明可能性、頑健性、安全性、人間の尊厳、異議申立ての機会、社会的受容性を、具体的な要件として定義する必要がある。

例えば、AI を採用選考、与信、保険、教育、医療、行政サービス、顧客対応、従業員評価に利用する場合、AI の判断や推奨が特定の属性、地域、年齢層、言語能力、障害の有無、経済状況に不利に働く可能性がある。AI が直接判断を行わない場合でも、担当者に提示する候補やスコアが、結果とし

[19] 実コード生成用などのシステム開発のプロンプトは、設計情報として構成管理されるものであり、要求・要件定義の時点ではプロンプトとして認識はされない。要求管理のライフサイクルの中で、要求定義工程で整理された要求が、設計工程でプロンプトとなる。

て人間の判断に強い影響を与えることがある。このため、AI の社会的影響を「運用後に問題が起きたら対応するもの」と考えるのではなく、要求定義の段階で検討すべきである。
社会的インパクト評価では、少なくとも次の観点を確認する。

表 22 AI の出力結果の社会的インパクト評価の観点の例

	確認すべき内容
影響を受ける人	利用者、顧客、従業員、取引先、地域社会など、誰に影響するか
不利益の可能性	特定の集団に不利な結果が生じないか
バイアス	学習データ、入力データ、評価指標に偏りがないか
差別・排除	AI の判断により、特定の人が不当に排除されないか
説明可能性	判断や推奨の理由を利用者や担当者に説明できるか
異議申立て	利用者が結果に疑義を持った場合、確認・修正・再審査できるか
心理的影響	AI による評価や応答が、利用者に不安、萎縮、不信感を与えないか
経済的影響	AI の判断が収入、取引機会、サービス利用機会に影響しないか
人間の関与	重要な判断に人間が関与する仕組みがあるか
継続的監視	運用後に影響を測定し、改善できるか

公平性や頑健性は、単なる技術要件として扱うのみでは不十分である。
公平性とは、統計的な偏りが少ないことに限らない。利用者や社会から見て、不当な扱いを受けていないと理解され、納得できることも含まれる。例えば、AI の分類精度が全体として高くても、特定の属性を持つ利用者に対して誤判定が多い場合、そのシステムは社会的に受け入れられない可能性がある。
同様に、頑健性も、技術的にエラーに強いことに限らない。社会状況、利用者行動、言語、文化、制度、業務条件が変化しても、不適切な判断や過度な不利益を生まないことが求められる。AI が想定外の入力、誤ったデータ、偏った情報、悪意ある操作に対してどのように振る舞うかは、技術上の問題であると同時に、社会的信頼に関わる問題である。
AI の社会的受容性を要件として扱う場合、次のような要求文に具体化できる。

- AI の判断または推奨が、特定の属性を持つ利用者に不利な結果を生じさせていないかを、定期的に評価すること
- 重要な判断に AI を利用する場合、最終判断者、確認手順、異議申立て手順を明確にすること
- AI の出力を利用者に提示する場合、AI が生成または推奨したものであることを明示すること
- AI の判断結果により利用者の権利、利益、取引機会に影響が生じる場合、判断根拠または確認方法を説明できること
- AI の利用により不利益を受けた可能性がある利用者からの問い合わせ、苦情、異議申立てを記録し、改善要求として管理すること
- AI モデル、入力データ、評価指標を変更した場合、特定の集団への影響を再評価すること

社会的インパクト評価は、法務部門や AI 担当部門のみで行うものではない。業務部門、利用者代表、顧客対応部門、データ担当、セキュリティ担当、監査担当、必要に応じて外部有識者を含め、複数の視点で確認することが望ましい。特に、AI の影響を受ける人々が要求定義の議論に含まれていない場合、そのシステムは重要なリスクを見落とす可能性がある。
AI 時代の要求定義では、「AI で何ができるか」に加え、「AI を使うことが社会的に妥当か」を確認する必要がある。公平性、頑健性、説明可能性、安全性は、技術的な性能指標であると同時に、社会的受容性を支える要件である。AI が組織の目的を達成していても、利用者や社会から信頼されなければ、長期的な価値は生まれない。
したがって、AI システムの要求定義では、社会的インパクト評価を要求分析の一部として位置づけるべきである。AI の利用目的、対象者、入力データ、判断内容、影響範囲、人間の関与、説明方法、異議申立て、継続的監視を要件として定義することで、AI システムを技術的に有効にするのみならず、社会的にも受け入れられるものに近づけることができる。

5.6 レジリエンスと継続的アシュアランスを要求として定義する

AI 時代の要求管理では、システムが運用開始時に安全で信頼できることのみを確認すればよいわけではない。システムは、利用データ、外部環境、業務ルール、利用者行動、法令・標準、AI モデル、プロンプト、外部サービスの変化により、運用後も性能やリスクが変化する。そのため、システムには、変化に対応しながら安全性、信頼性、説明責任を維持するためのレジリエンスが求められる。
レジリエンスとは、障害や変化が起きても、重要な機能や安全性を維持し、必要に応じて縮退運転、代替手段、復旧、改善を行える能力である。システムでは、性能劣化、データ分布の変化、ハルシネーションの増加、想定外入力、外部 API 障害、セキュリティ攻撃、法令変更、社会的批判などが発生し得る。これらに対して、単に障害対応手順を用意するのみでは不十分である。障害対応手順は必要であるが、それのみではレジリエンスを実現できない。これは次のような理由である。

- システムで生じる性能劣化やリスクの変化は、システム停止やエラーのような明確な障害として現れるとは限らず、正常に稼働しているように見える状態で徐々に進行することがある。
- データ分布、法令、業務ルール、利用者行動等の変化は、従来の障害対応手順では検知や対応の対象とならない可能性がある。

そのため、要求定義の段階で、何を監視し、どの条件で警告し、どの条件で安全モードに切り替え、誰が確認し、どのように改善するかを定義する必要がある。
この考え方を支えるのが、継続的アシュアランスである。アシュアランスとは、システムが安全で信頼できることを、根拠に基づいて説明・論証する考え方である。AI 時代には、開発時に一度限りで安全性や品質を確認するのではなく、運用後も継続的に監視し、評価し、根拠を更新し続けることが求められる。すなわち、システムが「現時点でも要求を満たしている」と説明できる状態を維持することが重要である。
従来の品質保証は、リリース前のテストや受入確認に重点が置かれていた。しかし、システムでは、リリース後にデータや環境が変化する。導入時には十分な精度があった AI が、半年後には新しい問い合わ

せ、制度改正、顧客行動の変化に対応できなくなることがある。生成 AI では、モデル更新や参照文書の変更により、出力の傾向が変わる可能性もある。したがって、システムのアシュアランスは、開発時点の確認から、運用中の継続的確認へと拡張する必要がある。
継続的アシュアランスで確認すべき観点は、次のとおりである。

表 23 継続的アシュアランスで確認すべき観点の例

	確認すべき内容
性能監視	精度、再現率、誤分類率、回答品質が維持されているか
安全性	誤判断、危険な出力、不適切な自動処理が発生していないか
公平性	特定の属性、地域、言語、利用者層に不利益が生じていないか
頑健性	想定外入力、データ欠損、外部環境変化に耐えられるか
セキュリティ	プロンプトインジェクション、データ漏えい、不正利用に対応できるか
説明責任	判断根拠、参照データ、ログ、承認履歴を説明できるか
データ品質	入力データや参照データの品質が維持されているか
モデル更新	モデル変更時に再評価、承認、影響分析を行っているか
運用体制	異常時の責任者、対応手順、エスカレーションが明確か
改善サイクル	監視結果や事故情報を要求更新に反映しているか

継続的アシュアランスでは、「安全である」と宣言するのみでは不十分である。なぜ安全と判断できるのか、どの要求を満たしているのか、どの証拠に基づいているのかを説明できなければならない。そのためには、アシュアランス・ケースの考え方が有効である。アシュアランス・ケースとは、あるシステムが安全で信頼できると主張するために、主張、根拠、証拠、前提条件を構造化して示す考え方である。
AI システムにおけるアシュアランス・ケースでは、例えば次のような構造を持つ。

主張

この AI 問い合わせ支援システムは、社内規則に基づく回答案を安全に提示できる。

根拠

- AI は承認済み文書のみを参照する
- AI は根拠文書を表示する
- AI の回答は利用者へ直接送信されず、担当者が確認する
- 高リスク質問は専門部署へエスカレーションする
- 回答品質と誤回答件数を月次で監視する

証拠

- 参照文書（要求、設計など）
- テスト結果
- 回答ログ

- 担当者レビュー記録
- 改善履歴

このように、システムが要求を満たしていることを、要求、設計、テスト、運用ログ、監査証跡と接続して説明できるようにすることが重要である。アシュアランス・ケースは、システムの安全性や信頼性を一時点で示すことのみを目的とするものではない。運用後の変更、モデル更新、データ更新、法令変更、事故対応のたびに見直すべき管理対象である。
レジリエンスを要件として定義する際には、異常時や性能劣化時の振る舞いを明確にする必要がある。AI が期待どおりに動作しない場合に、システムが危険な状態に進むのではなく、安全側に倒れる設計、すなわちフェイルセーフが重要である。
例えば、次のような要件が考えられる。

- AI の回答品質が一定水準を下回った場合、AI による自動回答を停止し、担当者確認モードに切り替えること
- AI の確信度が一定値を下回る場合、回答を確定せず、人間にエスカレーションすること
- 参照文書の根拠を確認できない場合、AI は推測で回答しないこと
- 外部 AI サービスに障害が発生した場合、業務継続に必要な代替手段を提供すること
- 高リスク分類の誤判定が一定件数を超えた場合、当該機能を停止し、原因分析を行うこと
- データ分布の変化が検知された場合、モデル性能を再評価すること

このような要件を定義しておくことで、システムが不確実性や変化に直面した場合でも、業務や利用者に重大な影響を与える前に制御しやすくなる。
また、デジタルツインを活用した事前シミュレーションも有効である。AI を含む業務やシステムのモデルを作成し、運用前に通常時、例外時、障害時、データ変化時、法令変更時、アクセス集中時、攻撃時の振る舞いを検証することで、リスクを事前に把握できる。特に、重要インフラ、行政サービス、金融、医療、製造、交通などでは、実環境で試すことが難しいシナリオをデジタルツイン上で検証することが重要になる。
デジタルツインを用いた安全性シミュレーションでは、次のような要件を定義できる。

- 重要業務に AI を適用する場合、運用開始前に通常時、例外時、障害時の業務シナリオをモデル上で確認すること
- AI の誤分類、誤回答、応答遅延、外部サービス停止が業務に与える影響をシミュレーションすること
- 高リスク業務では、AI が停止した場合の手動代替手順を確認すること
- 制度変更、データ項目変更、モデル更新が業務フローに与える影響を事前に確認すること
- シミュレーション結果を要求管理台帳、リスク管理台帳、変更管理記録に反映すること

継続的モニタリングでは、何を監視するかに加え、どの条件で対応を開始するか、すなわち更新のトリガー条件を定義する必要がある。監視項目を集めていても、基準がなければ、異常を検知しても対応が遅れる。そのため、AI 時代の要求管理では、監視指標、しきい値、対応責任者、対応手順、再評価条件をあらかじめ定めておくことが重要である。

更新のトリガー条件の例は、次のとおりである。

表 24 更新トリガー条件の要件の例

	要件化の例
性能劣化	精度、再現率、採用率、誤回答率が基準値を下回った場合、再評価を行う
データ分布変化	入力データの傾向が評価時データと大きく異なる場合、モデル影響を確認する
苦情・異議申立て	利用者からの苦情や異議申立てが一定数を超えた場合、原因分析を行う
インシデント	誤判断、情報漏えい、不適切出力が発生した場合、利用停止または制限を検討する
法令・標準変更	関連法令、ガイドライン、社内規則が変更された場合、要求と運用手順を見直す
モデル更新	AI モデル、プロンプト、参照文書を変更する場合、再テストと承認を行う
外部サービス変更	API 仕様、SLA、データ利用条件が変更された場合、影響分析を行う
監査指摘	監査で不備が指摘された場合、改善要求として管理する

継続的アシュアランスを現実的な負担で実施するためには、これらの確認を人手のみに依存させないことも重要である。AI やツールを用いて、運用ログ、評価結果、変更情報等を継続的に分析し、あらかじめ定義した基準・制約からの逸脱を検知するとともに、影響を受ける要求・要件や再評価が必要な項目を提示する仕組みを構築することが考えられる。人間は、その結果を踏まえて変更の承認や安全性・妥当性に関する判断を行う。

なお、ISO/IEC 42001 のような AI マネジメントシステムの考え方を踏まえると、AI システムの要求定義は、個別プロジェクトの技術仕様にとどまらない。組織として、AI に関するリスクと機会を管理し、方針、役割、プロセス、評価、改善の仕組みを持つことが求められる。したがって、継続的アシュアランスは、AI 開発チームのみの作業ではなく、経営、業務、AI、データ、セキュリティ、法務、監査、運用が連携して行う組織的活動である。

例えば、次のような要件が考えられる。

- AI システムの運用開始後、性能、安全性、公平性、セキュリティ、データ品質を継続的に監視すること
- 監視指標、しきい値、対応責任者、エスカレーション手順を定義すること
- AI モデル、参照データ、業務ルールなどを変更する場合、変更前後の影響を評価すること
- 高リスクな AI 利用については、アシュアランス・ケースを作成し、要求、根拠、証拠、前提条件を管理すること
- 運用中に重大な性能劣化または安全性低下が確認された場合、安全モード、手動運用、利用停止などの防御的措置を実行できること
- 監視結果、事故、監査指摘、利用者フィードバックを要求管理台帳に反映し、継続的改善につなげること

レジリエンスと継続的アシュアランスを要求として定義することにより、システムは、変化に対して脆弱な仕組みではなく、変化を検知し、安全に対応し、改善し続ける仕組みになる。AI 時代に重要なのは、最初に完全なシステムを作ることではなく、変化しても信頼を維持できるシステムを作ることである。そのためには、要求定義の段階から、監視、更新、フェイルセーフ、シミュレーション、アシュアランス・ケース、改善サイクルを組み込む必要がある。

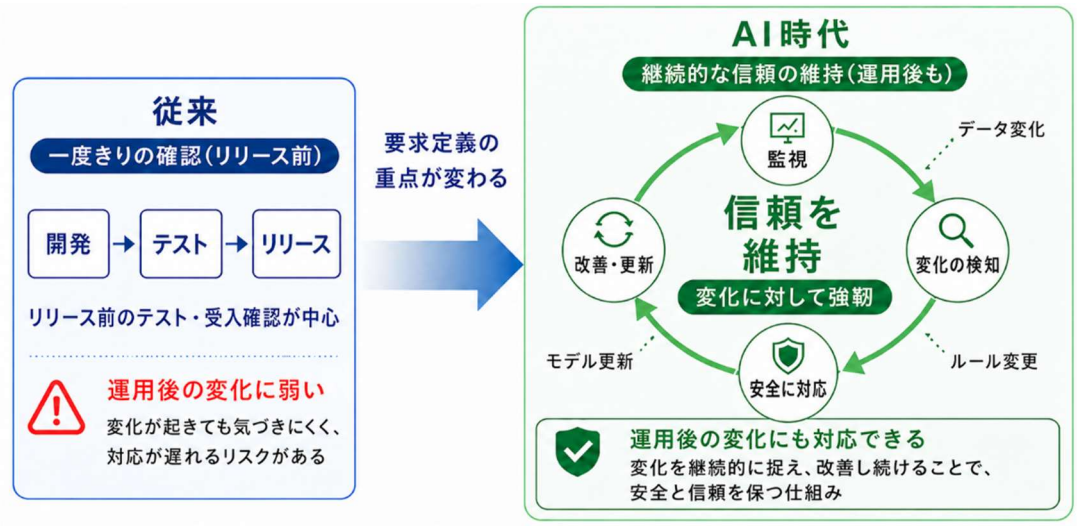


図 10 AI 時代に求められるレジリエンス・継続的アシュアランス要求

第 6 章 AI による要求管理支援

6.1 AI は要求管理の作業を支援する

AI は、要求管理の多くの作業を支援できる。例えば、会議録を要約し、関係者の発言から要求候補を抽出することができる。既存の仕様書やマニュアルを読み込み、機能一覧や業務フローのたたき台を作ることができる。問い合わせ履歴やクレームを分類し、利用者が困っている点を整理することもできる。さらに、既存システムのコードやログを分析し、システム構造や影響範囲を推定することも可能である。

特に重要なのは、AI が対話型で要求定義を支援できる点である。人間が最初から十分な要求を整理できていない場合でも、AI は不足している項目を質問し、ヒアリングを補助できる。例えば、AI は「利用者は誰か」「入力データは何か」「禁止すべき用途は何か」「例外処理はあるか」「受入基準は何か」「ログは必要か」といった問いを投げかけることができる。これにより、要求定義の経験が少ない担当者でも、一定の構造に沿って検討を進めやすくなる。

ただし、AI による支援は、人間の判断を置き換えるものではない。AI は、要求管理の作業を補助し、検討漏れを減らし、文書化やモデル化を効率化する道具である。最終的な目的、優先度、リスク許容度、法令・標準との整合、事業上の判断は、人間が責任を持って確認する必要がある。

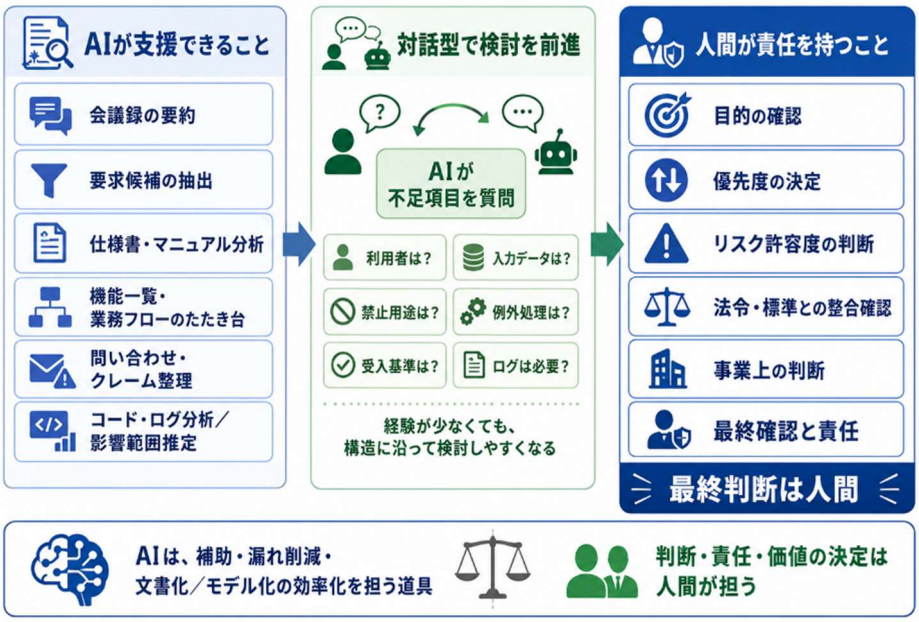


図 11 AI による要求定義支援と人間の責務

6.2 AI によるヒアリング支援

AI によるヒアリング支援では、担当者や利用者の発言を受けて、要求として不足している項目を聞き出す。従来、熟練した要求定義担当者が経験に基づいて行っていた質問を、AI が一定程度支援できる。例えば、利用者が「問い合わせ対応を AI で効率化したい」場合、AI は次のような質問を提示できる。

- 問い合わせの種類は何か
- 対象となる利用者は誰か
- AI は回答を直接返すのか、担当者向けの回答案を作るのか
- 参照する文書や FAQ は何か
- 個人情報や機密情報を含むか
- 回答できない場合はどうするか
- 人間が確認すべき条件は何か
- 誤回答が発生した場合の対応は何か
- 回答品質をどのように評価するか
- ログは何を保存するか
- 社内規則やマニュアルとの関係は何か
- 外部に回答する前に承認は必要か

このように、AI は要求定義の抜け漏れを減らすための質問リストを動的に提示できる。ヒアリングの場では、AI を議事録係や要約係に加え、要求定義の観点から「まだ確認していない事項」を提示する補助者として活用できる。

AI によるヒアリング支援を有効にするためには、事前に質問の観点を整理しておくことが望ましい。例え

ば、業務、データ、AI、法令・標準、運用、評価、セキュリティ、監査といった観点をプロンプトに組み込むことで、AI はより体系的に不足項目を確認できる。

6.3 AI によるドキュメンテーション支援

AI は、ヒアリング結果、会議録、既存文書、チャットログから、要求一覧、要求定義書、業務フロー、データ項目一覧、AI 要求管理、受入条件のたたき台を作成できる。
従来、要求定義のドキュメンテーションは、担当者の大きな負担であった。会議で合意された内容を整理し、要求として書き直し、関係者が理解できる文書にするには、多くの時間がかかる。AI を活用すれば、会議録から要求候補を抽出し、分類し、一覧化することができる。また、既存の規程やマニュアルから、義務、禁止、承認、記録、例外処理を抽出し、要求定義の材料にすることもできる。
ただし、AI が作成した文書は、そのまま正式な要求定義書として採用してはならない。AI は、文脈の誤解、重要事項の省略、過度な一般化、根拠不明な補完を行うことがある。したがって、AI が作成した成果物は、関係者レビューの対象とし、発生源、根拠、優先度、検証方法を確認する必要がある。
AI による要求定義のドキュメンテーション支援では、次の考え方が重要である。

表 25 AI による要求定義のドキュメンテーション支援における考え方

AI が支援できること	人間が確認すべきこと
会議録から要求候補を抽出する	重要な発言や意図が落ちていないか
要求を分類する	分類が業務実態と合っているか
要求定義書案を作る	実装可能・評価可能か
業務フロー案を作る	現場の例外処理を反映しているか
データ項目案を作る	データの意味、品質、利用条件が正しいか
受入条件案を作る	本当に確認可能か
社内規則との対応を示す	根拠規則や条項が正しいか

AI は、文書化の効率化に加え、要求の可視化にも役立つ。例えば、AI が要求一覧を作成し、各要求に「発生源」「関連業務」「関連データ」「検証方法」を付与すれば、関係者は確認すべき点を早く把握できる。

6.4 AI による過不足・矛盾の確認

AI は、定義された要求（特に、要件）に過不足がないか、矛盾がないか、曖昧な表現がないかを確認する支援にも使える。
例えば、AI は次のような観点で確認できる。

- 主語が明確か

- 何を実現するかが明確か
- 入力と出力が明確か
- 制約条件が明確か
- 例外処理が定義されているか
- 受入基準があるか
- ログや証跡が必要な場合に定義されているか
- 上位目的と対応しているか
- 法令・標準・社内規則と矛盾していないか
- データ要求と AI 要求が整合しているか
- 人間の関与が必要な場面が定義されているか
- 変更時の影響範囲を追跡できるか

ただし、このような AI による確認は、レビューの代替ではなく、レビューの前処理とする。AI により抜け漏れ候補や矛盾候補を抽出し、人間が確認することで、レビューの効率と品質を高めることができる。
特に AI システムでは、通常の機能要件に加え、入力データ、利用制限、説明責任、ログ、監視、モデル更新、人間の関与、インシデント対応が抜けやすい。AI を使って、これらの観点が要求定義書に含まれているかを確認することは有効である。

6.5 AI による要求のモデル出力

AI が作成した要求は、単なる文章ではなく、人間が理解しやすく、確認しやすいモデル形式で出力することが望ましい。例えば、業務フロー、データモデル、ユースケース、ルール表、トレーサビリティ表、リスク一覧として出力することで、関係者が確認しやすくなる。
代表的な出力形式は次のとおりである。

表 26 AI による要求のモデル出力形式例

	用途
Markdown	要求定義書、議事録、要求一覧の軽量な文書化
表形式	要求一覧、データ項目一覧、受入条件、リスク一覧
Mermaid	簡易なフロー図、シーケンス図、状態遷移図
BPMN	業務プロセスの標準的な表現
UML	ユースケース、クラス、シーケンス等
SysML	要求、構造、振る舞い、検証の接続
ArchiMate	業務、アプリ、データ、技術基盤の関係
JSON/XML	ツール連携、構造化データとしての利用
ReqIF	要求管理ツール間の要求情報交換

AI によるモデル出力では、標準的な形式を使うことが重要である。標準的な形式で出力すれば、人間による確認が容易になるとともに、要求管理ツール、モデリングツール、テストツールとの連携もしやすくなる。
例えば、会議録から抽出した要求を Markdown の表で出力し、重要な業務の流れを Mermaid や BPMN で可視化し、要求とテストの対応をトレーサビリティ表にする。さらに、要求管理ツールと連携する場合には、ReqIF 形式[20]で要求情報を交換する。このような流れを作ることで、AI を単なる文章作成支援ではなく、要求定義プロセス全体を支援する道具として活用できる。

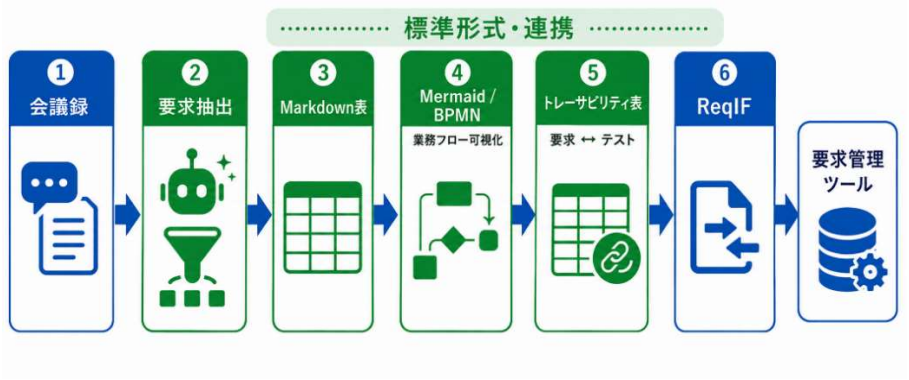


図 12 AI でのモデル出力による要求定義支援の例

モデル化の目的は、要求を分かりやすく可視化することのみではない。要求・要件、業務、データ、設計、実装、テスト、運用等の関係を機械的に扱える形で表現することにより、変更時の整合性確認、影響分析、再検証等を AI やツールによって支援・自動化するための基盤となる。

6.6 AI による変更影響分析と再検証支援

AI 時代の要求管理では、要求や要件を一度定義して終わりにするのではなく、業務、法令・標準、利用環境、データ、AI モデル等の変化に応じて継続的に見直すことが重要になる。一方で、要求や要件を変更するたびに、その影響を人手ですべて確認することは大きな負担となる。
このため、要求・要件と、業務、データ、設計、実装、テスト、ルール、運用情報等との関係をあらかじめ整理し、AI やツールを用いて変更影響の分析や再検証を支援することが有効である。
その支援パターンは大きく次の 3 つが考えられる。

1. 変更の影響範囲分析の支援
   ある要求や要件を変更した場合、その影響は当該要求のみにとどまるとは限らない。関連する業務プ

[20] ReqIF（Requirements Interchange Format）は、要求管理ツール間で要求情報を交換するための標準形式である。詳細は 8.5 節を参照されたい。

ロセス、データ、システム機能、AI モデル、プロンプト、法令・標準、テストケース、運用手順等にも影響する可能性がある。

要求・要件とこれらの関連情報がモデルやトレーサビリティによって接続されていれば、AI は、変更内容をもとに影響を受ける可能性のある要素を探索し、確認すべき対象の候補を提示できる。

例えば、次のような支援が考えられる。

- 法令の変更から、影響を受ける要求、業務プロセス、システム機能及びテストケースを抽出する
- 要求の変更から、修正が必要となる可能性のある設計や実装を特定する
- データ項目の変更から、そのデータを利用する AI モデル、業務、インターフェース及び評価項目を抽出する
- AI モデルやプロンプトの変更から、再確認すべき要求、評価指標、利用場面及びリスクを特定する
- 変更によって他の要求や制約との矛盾が生じていないかを確認する

AI による影響分析によって、変更範囲を自動的に確定できるとは限らない。重要なのは、人間が確認すべき対象を効率的に絞り込めることである。

2. 変更後に必要な再検証・再評価の支援

要求や要件を変更した場合には、変更内容を設計や実装へ反映するだけでなく、変更後も要求が満たされていることを再確認する必要がある。

AI やツールを活用することで、変更内容と既存のテストケース、評価基準、シミュレーション条件等との関係を分析し、再実施すべき検証項目の候補を提示できる。例えば、変更された要求に対応する回帰テストの抽出、新たに必要となるテストケースの候補生成、評価指標や受入条件との整合性確認、デジタルツインやシミュレーションによる影響確認等が考えられる。

また、システム上、機能が仕様どおりに動作することを確認するだけでは十分でない場合がある。変更によって、利用者の目的や業務上の価値、安全性、公平性、説明可能性等が損なわれていないかについても再評価する必要がある。

ただし、こうした妥当性の判断には、事業目的、利用者の期待、社会的影響等、人間による判断を必要とする要素が含まれる。そのため、AI は評価対象や確認すべき論点を提示し、分析やシミュレーションを支援する役割を担い、最終的な判断や承認は関係者が行うことが基本となる。

3. 継続的アシュアランスの支援

変更後の再検証は、変更時に一度実施すれば終わりというものではない。システムに関するデータ、利用環境、外部サービス、法令・標準等は継続的に変化するため、運用中も要求や制約が満たされているかを確認し続ける必要がある。

そのため、要求や要件に評価基準、監視指標、しきい値、テスト、運用ログ等を関連付け、AI やツールによってその状態を継続的に確認できるようにすることが重要である。例えば、性能、安全性、データ品質、セキュリティ等の指標が基準を外れた場合に、関連する要求や要件を特定し、再評価や変更

の必要性を提示する仕組みが考えられる。
このように、要求変更、影響分析、設計・実装への反映、再検証、運用監視を相互に接続することで、要求管理を継続的なサイクルとして実施しやすくなる。
この仕組みを実現するためには、要求や要件を単独の文書として管理するのではなく、業務、データ、モデル、設計、実装、テスト、運用ログ等との関係を明示しておくことが重要である。モデルやトレーサビリティは、人間の理解や説明責任を支えるだけでなく、AI が変更の影響を分析し、必要な再検証を支援するための基盤にもなる。これらについては、第 9 章及び第 11 章で詳しく説明する。

6.7 AI 活用時の注意点

AI を要求管理に活用する場合、次の点に注意する。

- AI の出力を要求候補として扱い、そのまま採用しない
- 出典、根拠、前提条件を確認する
- 専門家、現場、利用者によるレビューを行う
- 少数意見や例外処理を確認する
- 法令、標準、社内規則との整合性を確認する
- 実装可能性と評価可能性を確認する
- AI が見落としやすい新しい選択肢を人間が補う
- AI が作成したモデルや文書にも版管理を行う
- AI に入力してよい情報と入力してはいけない情報を明確にする
- AI による要件案の責任主体を明確にする

AI 時代には、要求管理の能力が不要になるのではない。むしろ、AI の出力を評価し、修正し、使いこなすために、要求工学や業務分析の基礎がより重要になる。

6.8 複数の AI エージェントの連携による要求管理

複数の AI エージェントが役割を分担し、要求の獲得、分析、仕様化、検証及び変更管理を協調して行うことも可能である。単一の AI エージェントでこれらの活動を行う場合と比べ、専門的な役割分担や成果物の相互確認が可能となる。
ただし、本書執筆時点（2026 年 10 月時点）では、複数の AI エージェントを連携させたとしても、AI のみで要求の妥当性判断や正式な承認まで完結させる実践は確立しているとは言えず、専門 AI エージェントによる成果物の作成・検査と、人間による意味確認、優先順位付け及び承認を組み合わせる形が妥当である。
なお、本章でいう複数の AI エージェントの連携は、次の 2 つに分けて考えることができる。

表 27 複数の AI エージェントの連携の形態

連携形態	用途

組織内連携	要求候補の抽出、分析、仕様化、レビュー等の専門エージェントが互いに連携する
組織・システム間連携	発注者、供給者、法務、検証等の異なる組織・システム間エージェント基盤が作業を委任し、成果物を交換する

例えば、組織内連携として「模擬ステークホルダーエージェント」と「要求ヒアリングエージェント」を連携させるということが考えられる。ただし、ハルシネーション等のリスクを考慮し、その成果物は、実際の関係者の意思又は合意を代替するものではなく、確認すべき論点や質問を増やすために用いることが妥当である。つまり、「6.2 AI によるヒアリング支援」での実施内容を補強するものに過ぎない。
いずれの連携形態にせよ、成果物品質をより安定させるために、複数の AI エージェントの連携は自然言語による会話ではなく、構造化された要求成果物を連携の中心とすることが望ましい。要求情報としては、「表 9　要求管理台帳に持たせる項目の例」で整理した項目以外に「作成・レビューに用いた AI エージェント及びモデル等」も記録・管理できるとなおよい。
AI エージェントの役割は、要求候補の抽出、重複・矛盾・曖昧性の検出、モデル化、受入基準の生成、影響分析及び差分作成を中心とする。要求の正式採用、優先順位、リスク受容、法的判断、ベースライン化及び変更承認は、人間の責任者が行う。例えば、要求候補レビューエージェントを設ける場合も、同一モデルによる検査のみに依存せず、ルール検査、スキーマ検証、異なる知識源及び人間レビューを組み合わせる。
MCP（Model Context Protocol）を活用した場合でも変わらない。MCP は、AI エージェントが要求管理ツール、文書、法令、設計モデル、テスト結果等の外部資源を参照し、登録、検索又は分析等のツールを実行するための接続層として利用できるが、MCP Server を接続するのみで要求管理が成立するわけではない。こうした外部資源との接続を有効に活用するためには、要求・要件だけでなく、その背景にある目的、判断根拠、制約条件なども AI が適切に参照できるようにすることが重要である。MCP はそれらの情報にアクセスするための手段の一つであり、情報自体の品質や、AI に与えるコンテキストの適切性については別途確保する必要がある。
したがって、AI エージェントを活用しても、要求管理のプロセスは大きくは変わらない。AI エージェントが、構造化された要求成果物を介して協働し（ただし、AI エージェントが承認済み要求を直接上書きしない構成とする）、人間が意思決定責任を保持し、すべての変更根拠と承認履歴を追跡できる仕組みとすることが重要である。

第 7 章　AI 時代の要求管理プロセス

7.1 AI 時代の要求管理プロセスの全体像

AI 時代の要求管理は、要求定義書を作成して終わるものではない。目的の確認から運用、フィードバック、変更管理までを含む循環的な活動である。

INCOSE が定義するシステムのライフサイクル[21]の要求定義に関する部分や、REBOK [27]で整理されているプロセスを参考にすると、標準的な流れは次のように整理できる。

1. 目的・ミッションの確認
2. ステークホルダーの特定
3. 要求獲得
4. 要求分析
5. 要求仕様化
6. モデリング
7. 要求評価・妥当性確認
8. 要求ベースライン化
9. 実装・調達・開発への接続
10. 運用・監視
11. フィードバック・変更管理
12. 廃止・移行

この流れは、必ずしも一方向に進むものではない。例えば、要求分析の結果、追加の要求獲得が必要になることがある。他にも、モデリングの結果、業務やデータの不整合が見つかることや、運用後のログや利用者の声から、新しい要求が発生することもある。

AI 時代の要求管理では、要求の新規発生や更新がより高頻度に起こりうることを踏まえ、このような繰り返しを前提にする必要がある。重要なのは、最初からすべてを完全に決めることではなく、要求を管理し、変化に応じて更新できる状態にしておくことである。

7.2 各工程の考え方

7.2.1 目的・ミッションの確認

最初に確認すべきことは、「何を作るか」ではなく、「何を実現したいか」である。AI を使いたい、システムを刷新したい、業務を自動化したいという要望の前に、事業上・業務上・社会上の目的を明確にする必要がある。ここでは、何の価値を実現したいのか、誰の課題を解決するのか、どの業務やサービスを変えるのか、何を成功とみなすのか、どのリスクを避けるべきかを確認する。なお、この段階で整理する目的や実現したい価値は、必ずしも確定したものではなく、後続のステークホルダーとの対話を通じて確認・具体化し、必要に応じて見直すものである。

第 3 章で示したデザイン思考やミッションエンジニアリングが参考になる。

---

[21] INCOSE のシステムズエンジニアリングハンドブックによると、一般的なシステムライフサイクルは、「1. コンセプトステージ」「2. 開発ステージ」「3. 生産ステージ」「4. 利用ステージ」「5. サポートステージ」「6. 廃棄ステージ」であるとしている。 [24]

7.2.2 ステークホルダーの特定

要求定義では、関係者（ステークホルダー）を早い段階で整理する必要がある。経営層、事業部門、利用者、現場担当、IT 部門、AI 担当、データ担当、法務、セキュリティ、監査、調達、外部ベンダーなど、ステークホルダーは多岐にわたる。AI を使う場合には、AI の出力を利用する人、AI の判断を確認する人、データを管理する人、事故時に対応する人も明確にする必要がある。

また、ステークホルダーを特定するだけでなく、それぞれが置かれている状況や抱えている課題、問題意識、制約条件など、要求の背景となるコンテキストを相互に理解することが重要である。その上で、7.2.1 で整理した目的や実現したい価値について認識を擦り合わせ、プロジェクトが目指す基本的な方向性（コンセプト）を共有する。こうした共通理解が不十分なまま要求・要件を定義すると、個々の要件が正確に記述されていても、プロジェクト全体として期待する価値を実現できない可能性がある。

7.2.3 要求獲得

要求獲得では、インタビュー、ワークショップ、現場観察、既存文書調査、問い合わせ分析、ログ分析、クレーム分析などを通じて、要求候補を集める。AI を使えば、大量の文書やログから要求候補を抽出できるが、AI の出力は候補として扱い、人間が確認する必要がある。

第 6 章で示した AI による要求獲得支援、第 8 章で示す AI 時代の要求獲得の考え方、第 12 章や第 13 章で示すルールや制度を要求定義に活かす考え方が参考になる。

7.2.4 要求分析

要求分析では、集めた要求候補を分類し、重複、矛盾、優先度、実現可能性、リスクを整理する。AI 時代には、機能要求に加え、データ要求、AI 要求、ルール要求、評価要求、運用要求も明確にする。また、各要求について、それをどのような手段で実現するかを検討する。システム、AI、人、業務プロセス、データ整備、組織・権限、ルール・制度等の選択肢を比較し、必要に応じて組み合わせる。その上で、システムや AI に担わせる範囲を明確にし、後続の要求仕様化につなげる。

第 8 章で示す AI 時代の要求分析の考え方が参考になる。

7.2.5 要求仕様化

要求仕様化では、分析した要求を、関係者が理解でき、実装や評価に使える形で記述する。自然言語の文章に加え、表、図、業務フロー、データモデル、受入条件などを組み合わせる。

第 8 章で示す AI 時代の要求仕様化の考え方が参考になる。

7.2.6 モデリング

モデリングでは、業務、データ、AI 利用、ルール、システム構成、評価の関係を可視化する。すべてを詳細にモデル化する必要はないが、重要業務、高リスク業務、AI 利用部分、法令・標準に関係する部分は優先的に可視化する。

第 9 章で示す要求の可視化方法、第 14 章で示す文書の構造化と標準形式が参考になる。

7.2.7 要求評価・妥当性確認

要求評価では、定義した要求が本当に必要か、実現可能か、検証可能か、関係者の理解が一致して

いるかなどを確認する。AI を利用する場合は、入力データ、出力、人間の関与、評価指標、監視方法、ログ、更新条件も確認する。
第 2 章で示した要求評価の基本、第 6 章で示した要求の妥当性確認目的での AI によるチェックの考え方が参考になる。

7.2.8 要求ベースライン化

関係者が合意した要求は、要求ベースラインとして管理する。ベースラインを設定することで、どの要求を前提に設計、開発、調達、評価を行うのかが明確になる。
第 2 章で示した（狭義の）要求管理の基本が参考になる。

7.2.9 実現手段への接続

前段までで整理した要求を、要求分析で検討した要求の実現手段へ接続する。要求の実現手段は必ずしもシステム開発に限らないが、システムや AI によって実現する要求については、設計、開発、AI モデル選定、データ整備、調達仕様、契約、テスト計画へ引き継ぐ。要求定義書を作るのみで終わらせず、実装や調達をはじめとした要求の実現手段に確実に接続することが大切である。
第 11 章で示すトレーサビリティの考え方が参考になる。

7.2.10 運用・監視

運用開始後は、システムや AI が期待どおりに機能しているかを確認する。利用状況、エラー、問い合わせ、ログ、AI 出力の品質、監査結果を確認し、改善につなげる。
第 5 章で示したレジリエンスと継続的アシュアランスの考え方が参考になる。

7.2.11 フィードバック・変更管理

運用データ、利用者の声、法令・標準の変更、業務変更、AI モデル更新に応じて、要求を見直す。変更要求を受け付け、影響分析を行い、承認後に新しいベースラインへ反映する。
この際、要求変更の影響を要求間だけでなく、関連するモデル、設計、実装、テスト、評価基準、運用手順まで追跡し、必要な再検証・再評価につなげる。AI やツールを活用して、影響範囲の抽出、整合性確認、テスト対象の特定等を支援・自動化することも有効である。
なお、変更が上位の目的・ゴールそのものに及ぶ場合には、個別の要求変更として処理するのではなく、ビジネス又はミッション分析まで戻り、業務、機能、データ等を再整理した上で、新たな要求ベースラインを設定する。
第 11 章で示す変更管理の考え方が参考になる。

7.2.12 廃止・移行

システムや AI サービスを廃止・移行する場合にも、要求定義が必要である。データ移行、ログ保存、契約終了、利用者通知、代替手段、リスク対応を整理する。

第 8 章　AI 時代の要求獲得・分析・仕様化

8.1 要求獲得の対象の拡大

従来の要求獲得では、主に利用者、業務部門、管理者、経営層、システム担当者から要求を聞き取っていた。具体的な手法としては、インタビュー、ワークショップ、現場観察、既存文書調査などが中心であった。

AI 時代には、要求獲得の対象が広がる。人に聞くのみでなく、組織や社会に蓄積されたデータから要求候補を抽出できるようになるためである。問い合わせ履歴、クレーム、事故報告、障害報告、監査指摘、FAQ、チャット、メール、会議録、操作ログ、SNS、パブリックコメントなどが、要求獲得の重要な情報源になる。

例えば、問い合わせ履歴を分析すれば、利用者がどの手続でつまずいているかが分かる。クレームを分析すれば、利用者が不満を持つ原因が見えてくる。障害報告を分析すれば、運用上の弱点が分かる。操作ログを分析すれば、実際には使われていない機能や、途中で離脱されている画面が見つかる。監査指摘を分析すれば、コンプライアンス上の改善要求が見える。

このように、AI 時代の要求獲得では、利用者や関係者の声に加え、データに表れた行動、エラー、例外、リスクを読み取ることが重要になる。

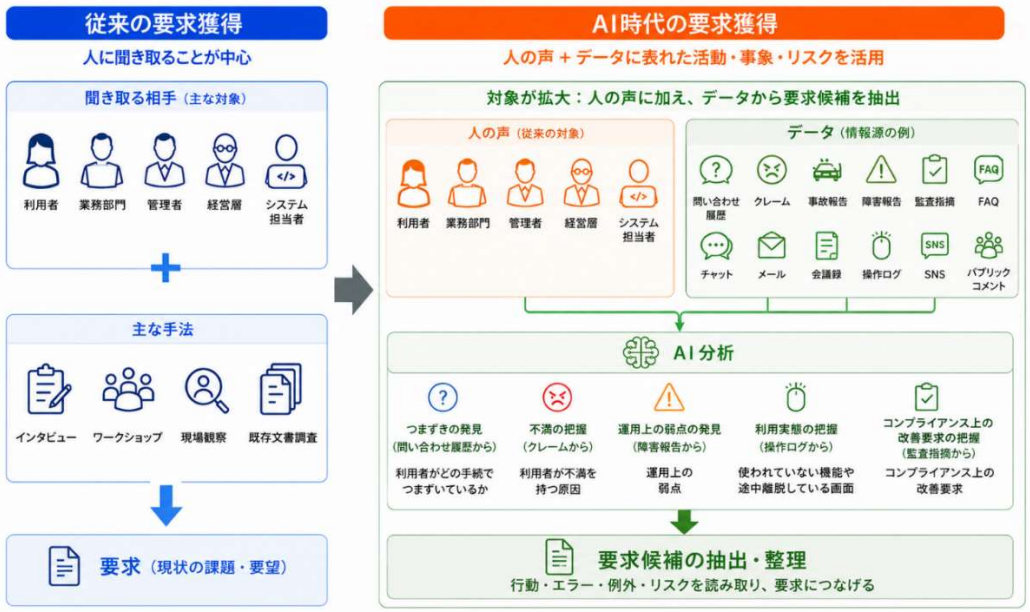


図 13　要求獲得の対象の拡大

8.2 マスの意見と「光る原石」を区別する

AI は、大量の情報を整理することが得意である。そのため、多くの利用者が同じように困っていること、頻

繁に発生する問い合わせ、よくあるエラー、繰り返されるクレームを見つけることができる。これは、マスの意見を把握するうえで有効である。
しかし、要求定義では、多数意見のみを見ればよいわけではない。少数の意見の中に、重要な発見が含まれることがある。まだ多くの人が気づいていない新しいニーズ、重大事故につながる小さな兆候、専門家のみが指摘する本質的な課題、将来のビジネスチャンスにつながる発想が含まれている場合がある。
このような少数だが重要な要求候補を、本書では「光る原石」と呼ぶ。AI は、光る原石を見つける支援はできるが、何が本当に重要かを判断するには、人間の専門性、現場知、経営判断が必要である。
例えば、ある問い合わせが全体の 1%しかなくても、それが高リスクな顧客、重要な制度変更、重大事故の兆候、新しい市場ニーズに関係している場合は、無視してはならない。逆に、多数の要望があっても、事業目的や法令、リスク管理の観点から採用すべきでない場合もある。
AI 時代の要求獲得では、次の二つを分けて考えることが望ましい。

表 28 「マスの意見」と「光る原石」

	意味	対応
マスの意見	多くの利用者に共通する要望や不満	AI で分類・集計し、優先度を検討する
光る原石	少数だが重要な兆候や新しい要求	専門家が深掘りし、価値やリスクを評価する

AI に全面的に委ねた場合、平均的な要求に寄りやすい。変化に強い組織を作るには、マスの意見を把握しつつ、重要な少数の兆候を見逃さない仕組みが必要である。

8.3 潜在要求を見つけるための問い

潜在要求とは、利用者自身がまだ明確に言葉にできていない要求である。利用者は、自分が知っている手段を前提に要望を表現することが多い。しかし、本当に実現したい価値は、その手段とは別のところにある場合がある。
例えば、利用者が「紙の申請書を PDF で提出できるようにしたい」と言ったとする。この要求の背後には、「窓口に行きたくない」「入力を簡単にしたい」「審査を早くしてほしい」「手続状況を知りたい」という要望があるかもしれない。もし目的を深掘りしなければ、紙の手続を PDF 化するのみで終わり、根本的な改善にはならない。
AI を使えば、利用者の発言から要求候補を抽出できる。しかし、潜在要求を見つけるには、人間が問いを立てる必要がある。
潜在要求を探るための問いは、次のとおりである。

- 利用者は、なぜそれを求めているのか
- その要求の背景にある困りごとは何か
- 本当に必要なのは、その手段なのか、それとも別の価値なのか
- 現在の制度や業務を前提にしすぎていないか

- もっと簡単な方法、安い方法、安全な方法はないか
- データ連携や既存サービスで代替できないか
- 法令や標準の目的は何か
- その目的を別の技術で実現できないか
- 少数意見の中に将来重要になる兆候はないか
- AI が一般的な回答に寄りすぎていないか

AI 時代の要求獲得では、AI に質問することに加え、人間が AI に対して良い問いを投げることが重要である。良い問いを持たない組織は、AI を使っても表面的な要求の整理にとどまる。良い問いを持つ組織は、AI を使って潜在要求を掘り起こし、新しい価値を発見できる。

8.4 データ供給チェーンの透明性とトレーサビリティ

AI 時代の要求管理では、データを単なる入力情報として扱ってはならない。AI の性能、リスク、公平性、説明可能性、社会的受容性は、利用するデータに大きく左右される。どれほど高度な AI モデルを利用しても、データの出所が不明確で、品質が低く、偏りがあり、利用権限や著作権処理が不十分であれば、信頼できる AI システムにはならない。

したがって、AI を利用する要求定義では、データ項目を定義することに加え、データがどこから来て、どのように収集され、どのように加工され、誰が確認し、どの目的で利用され、どのように更新・削除されるのかを、データ供給チェーンとして管理する必要がある。これは、製造業において部品や材料の調達先、品質、加工履歴、検査結果を管理することに似ている。AI におけるデータは、製品における材料であり、データ供給チェーンの不備は、AI システムの品質不良やリスクの原因になる。

データ供給チェーンとは、AI やシステムで利用されるデータが、発生、収集、取得、加工、クレンジング、アノテーション、統合、保管、利用、評価、更新、削除されるまでの一連の流れである。要求定義では、この流れを可視化し、各段階で必要な管理要件を明確にする必要がある。



図 14 製造業の一般的なプロセスと AI 時代のデータ供給チェーンの比較

まず確認すべきなのは、データの出自、すなわち Data Provenance である。どのデータが、いつ、どこで、どの目的で、誰によって収集されたのかを明確にする必要がある。個人情報を含む場合には、当初の収集目的、同意、利用条件、第三者提供、越境移転の有無も確認しなければならない。外部データを利用する場合には、ライセンス、契約、著作権、再利用条件、商用利用可否、二次利用の制約を確認する必要がある。

次に、学習データ、検証データ、テストデータの構成を確認する必要がある。AI モデルの性能は、どのデータで学習し、どのデータで評価したかに依存する。評価データが偏っていれば、実運用では期待した性能が出ない。特定の地域、年齢層、言語、業務パターン、例外ケースが不足していれば、AI はそれらの状況で誤った出力をする可能性がある。

そのため、AI 時代の要求管理では、次のようなデータセットの構成要件を明確にする必要がある。

表 29 AI 時代の要求管理でのデータセット構成要件の確認観点

	要求定義で確認すべき内容
出典	データの取得元、取得方法、取得時期は明確か
利用目的	当初の収集目的と AI 利用目的は整合しているか
権利処理	著作権、ライセンス、契約、同意は確認されているか
多様性	利用者、地域、言語、業務パターンが適切に含まれているか
代表性	実運用で想定されるデータ分布を反映しているか

完全性	必要な項目やケースが欠落していないか
正確性	誤記、重複、古い情報、不正確な情報が含まれていないか
バイアス	特定の属性や集団に偏ったデータになっていないか
分割方法	学習、検証、テストの分割が適切か
更新頻度	データがどの頻度で更新されるか
保存・削除	保存期間、削除条件、利用停止条件が明確か

また、クレンジングやアノテーションの品質保証も重要である。AI モデルの開発では、データの欠損を補完し、重複を除去し、表記ゆれを統一し、ラベルを付与する作業が行われる。この過程で誤った補正や不適切なラベル付けが行われると、AI モデルの判断に直接影響する。

例えば、問い合わせ分類 AI で、苦情、質問、申請、障害報告、その他といったラベルを付与する場合、ラベル付けの基準が曖昧であれば、担当者によって分類が異なる。ある担当者は苦情と分類し、別の担当者は質問と分類するかもしれない。このようなばらつきがあると、AI は一貫した判断を学習できない。

したがって、アノテーションに関しては、次のような要件を定義する必要がある。

- アノテーション基準を文書化すること
- 複数のアノテーターが作業する場合、判断基準を共有し、サンプルレビューを行うこと
- 高リスク分類については、専門家または責任者による確認を行うこと
- アノテーション結果のばらつきを定期的に確認すること
- ラベル定義を変更した場合、既存データへの影響を確認すること
- アノテーション履歴、担当者、確認者、変更理由を記録すること

データ供給チェーンでは、データの加工履歴も重要である。元データがどのように加工され、どの項目が削除され、どの値が補正され、どのように匿名化または仮名化されたのかを追跡できる必要がある。加工履歴が不明確なデータは、問題が起きたときに原因を特定できない。また、後から法令や契約上の問題が判明した場合に、どの AI モデルやシステムが影響を受けるのかを把握できない。

データ供給チェーンのトレーサビリティでは、次の関係を管理する。

表 30 データ供給チェーンのトレーサビリティで管理する関係

	目的
データソースと利用目的	目的外利用を防ぐ
データソースとライセンス	権利処理や再利用条件を確認する
元データと加工データ	加工内容や品質変化を追跡する
加工データと学習データ	モデルが何を学習したかを確認する
学習データと AI モデル	モデルの性能やバイアスの原因を追跡する
評価データと評価結果	評価の妥当性を確認する
運用データと改善要求	実運用での性能劣化や新しい要求を把握する
データ変更とモデル更新	データ更新が AI 出力に与える影響を管理する

AI 時代には、データ品質管理を一部のデータ担当者のみの作業として扱うべきではない。データ品質は、AI の安全性、信頼性、公平性、コンプライアンス、事業価値に直結するため、組織の必須能力として位置づける必要がある。特に、欧州 AI 法 [28]のような規制動向を踏まえると、高リスク AI システムでは、学習データ、検証データ、テストデータの品質、出典、適切性、バイアス管理、データガバナンスを説明できることが重要になる。

データ品質管理要件は、次のように具体化できる。

- AI システムで利用するデータについて、データソース、取得方法、取得目的、利用条件を記録すること
- 学習データ、検証データ、テストデータの構成、件数、期間、対象範囲、代表性を記録すること
- 個人情報、機密情報、著作物を含むデータについて、利用権限、同意、契約、ライセンスを確認すること
- データクレンジング、匿名化、アノテーション、統合、除外の処理履歴を記録すること
- データセットに偏りがないかを確認し、特定の属性や集団への影響を評価すること
- 運用開始後も、入力データの分布変化、欠損、異常値、品質劣化を監視すること
- データ品質に問題が見つかった場合、モデル、プロンプト、業務手順、評価基準への影響を分析すること
- データソース、利用条件、法令、契約が変更された場合、影響を受ける AI モデルとシステムを特定すること

データ供給チェーンの透明性は、AI の説明責任にも直結する。AI が不適切な出力をした場合、モデルのみを調べても原因が分からないことがある。学習データに偏りがあったのか、評価データが実運用を反映していなかったのか、アノテーション基準が不明確だったのか、古いデータが残っていたのか、外部データの利用条件が変わったのかを確認する必要がある。そのためには、データの出自、加工、利用、評価、更新の履歴を残しておく必要がある。

また、データ供給チェーンは、サプライチェーンリスクの一部でもある。AI モデルや AI サービスを外部から調達する場合、利用されているデータの出典、学習データの範囲、権利処理、評価方法が不明確なことがある。外部 AI サービスを利用する場合には、サービス提供者がどのようなデータ管理を行っているのか、自社の入力データが学習に使われるのか、ログがどこに保存されるのか、データ削除に対応できるのかを確認しなければならない。

このように、AI 時代の要求管理では、データを「入力項目」として定義するのみでは不十分である。データの出自、権利、品質、加工、アノテーション、評価、運用、更新、削除までを、データ供給チェーンとして要件化する必要がある。データ品質管理は、AI システムの付随作業ではなく、AI ガバナンス、リスク管理、コンプライアンス、価値創出を支える中核的な組織能力である。

なお、本節では、AI 時代の要求定義において考慮すべきデータマネジメントの主な観点を取り上げたが、これらがデータマネジメントのすべてではない。データマネジメントには、データ品質管理やトレーサビリティのほか、データガバナンス、データアーキテクチャ、メタデータ管理、データセキュリティなど、幅広い領域が含ま

れる。実際の要求定義に当たっては、利用目的やリスクに応じて必要な事項を検討するとともに、詳細については DAMA-DMBOK 等の専門的な知識体系や関連する国際標準を参照することが望ましい。

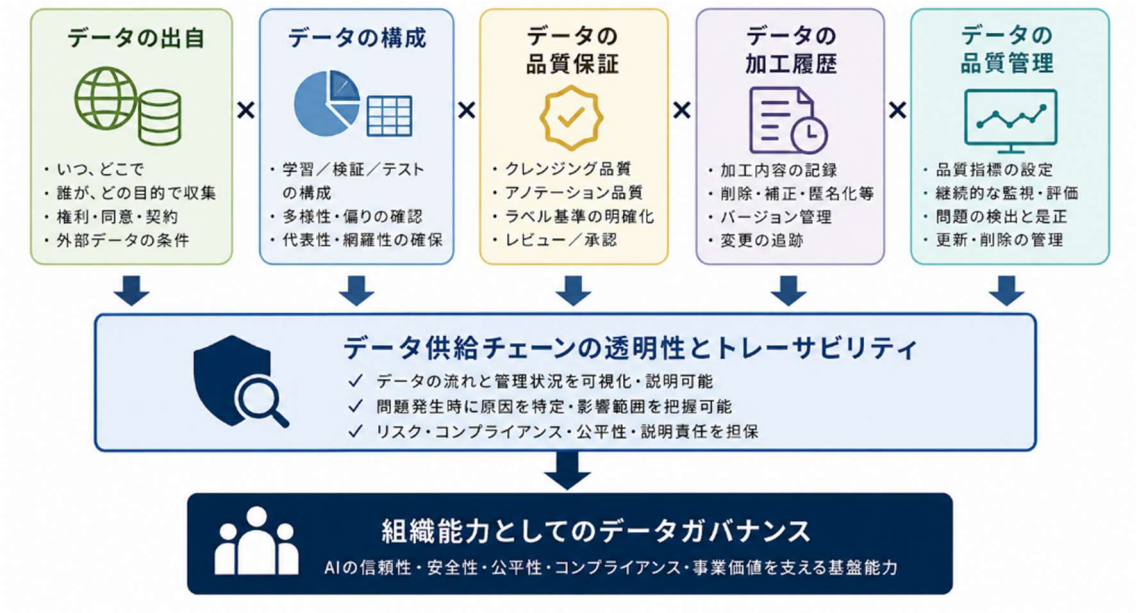


図 15 データ供給チェーンの透明性とトレーサビリティ

8.5 AI 時代の要求分析

AI 時代の要求分析において重要なことは、最初から AI の利用を前提としないことである。「AI を使いたい」「生成 AI を導入したい」「AI で自動化したい」という要望があった場合でも、まず、その背景にある目的や課題を確認し、何を実現する必要があるのかを明確にする必要がある。

その上で、要求を実現する手段として AI が適切かを分析する。要求の実現方法は AI のみではない。ルールベースの処理、従来型のソフトウェア、既存システム、SaaS や外部サービス、人による業務、あるいはそれらと AI との組合せなど、複数の実現方法が考えられる。

特に、処理条件や判断基準を明確なルールとして定義でき、同じ入力に対して同じ結果を求める業務では、AI を利用する必要がない場合がある。このような業務を AI で実現すると、データの整備、AI の評価、出力の監視、モデルやプロンプトの更新、人による確認等が追加で必要となり、ライフサイクル全体のコストが高くなる可能性もある。

一方、大量の非構造データを扱う場合、あらかじめすべての判断ルールを定義することが困難な場合、画像・音声・自然言語の認識や生成を行う場合など、AI を利用することで要求を効果的に実現できる領域もある。

したがって、要求分析では、AI を利用するか否かをシステム全体で一律に判断するのではなく、業務や機能ごとに、要求の性質、代替手段、期待される効果、リスク及びライフサイクル全体のコストを比較し、AI を利用する必要性と適用範囲を見極めることが重要である。

AI を要求の実現手段として利用する場合には、要求分析の対象を従来以上に広く捉える必要がある。従来のシステム開発では、機能要件と非機能要件を中心に分類することが多かった。しかし、AI を利用するシステムでは、データ、AI モデル、運用、評価、法令・標準、説明責任が密接に関係するため、第 2 章で示したビジネス要求、ステークホルダー要求、システム要求、ソフトウェア要求に加え、特にデータ要求、AI への要求、ルール要求、評価要求、運用要求を明確に区別し、それぞれの関係を分析することが重要となる。ただし、これらは AI システムだけに必要な要求ではない。データ要求、ルール要求、評価要求、運用要求は従来のシステムにも存在する。AI を利用する場合には、AI の性能やリスクがこれらに強く依存するため、従来以上に明示的に扱う必要がある。

その上で、AI を利用する部分については、特に次の 4 点を重点的に確認する。

1. AI に任せる範囲と人間が担う範囲を明確にする

   AI が自動的に処理・判断するのか、担当者に候補や推奨を提示するのみなのか、人間による確認や承認を必要とするのかを明確にする。特に、誤りが利用者や業務に大きな影響を与える場合には、AI の判断のみで処理を完結させるのではなく、人間へのエスカレーション条件や責任分担を分析する必要がある。

2. データ要求を明確にする

   AI は利用するデータに性能やリスクが左右される。そのため、必要なデータを取得できるか、品質や代表性は十分か、利用条件や権利上の問題はないか、運用後も継続して取得・更新できるかを確認する。必要なデータを適切に確保できない場合には、AI を利用するという判断自体を見直す必要がある。

3. 評価要求を明確にする

   AI の出力は確率的であり、単純な合否のみでは評価できない場合がある。精度、再現率、公平性、説明可能性、安全性、利用者満足度、業務効率、誤回答率など、目的に応じた評価指標と許容水準を設定する。また、AI を利用しない実現方法と比較して、AI の利用によって期待する価値が得られるかを確認する。

4. 運用後の監視と変更管理を考慮する

   AI システムは導入時に適切でも、時間とともに性能が劣化したり、利用状況が変化したりする。そのため、運用後に何を監視するか、どの条件で再評価するか、どのような場合に AI の利用方法を変更・停止するかを要求として明確にする必要がある。

これらの分析を通じて、当初は AI の利用が適切と判断した場合であっても、必要なデータを確保できない、要求する品質を達成できない、リスクに対する対策が困難である、ライフサイクル全体のコストが期待する効果に見合わない、といったことが判明する場合がある。その場合には、AI の適用範囲を縮小したり、ルールベースの処理や人間による判断等、別の実現方法へ戻って検討したりすることも必要である。

8.6 AI 時代の要求仕様化

要求仕様化では、分析した要求を、実装や評価に使える形で記述する。そのため、自然言語の要件文

に加え、モデル、表、データ定義、評価基準、ログ要件、運用ルールなどを組み合わせて仕様化することが望ましい。
AI 時代かに関わらず、一般的に、要求仕様には次の観点を含める[22]。

表 31 要求仕様に含めるべき観点

	内容
目的	何を実現するための要求か
対象	どの業務、利用者、システムが対象か
入力	何を入力とするか
処理	どのように処理するか
出力	何を出力するか
制約	何をしてはいけないか
例外	通常と異なる場合の扱い
評価	どう確認するか
監視	運用後に何を観測するか
変更	いつ、どう見直すか
証跡	何を記録するか

しかし、特に AI に関する要求仕様では、「正しい出力」の定義が難しい場合がある。そのため、単純な合否に加え、許容範囲、レビュー方法、再評価条件、人間の判断との関係を明確にする必要がある。
例えば、生成 AI による回答案作成では、次のように仕様化できる。

- AI は問い合わせ内容、FAQ、過去回答例を参照し、担当者向けの回答案を作成すること
- AI の回答案は利用者に直接送信しないこと。
- 担当者は回答案を確認し、必要に応じて修正したうえで送信すること。
- AI が参照した文書、生成時刻、担当者の修正有無をログとして保存すること。
- 月次で、回答案の採用率、修正率、誤回答件数を確認し、FAQ、プロンプト、参照文書を見直すこと。

この要求仕様では、目的、対象、人間の関与、禁止事項、証跡、評価、改善が明確になっている。

[22] REBOK [27]では要求仕様を 5W1H の観点に基づいて構造の視点、挙動の視点、機能の視点などから整理することを推奨している。

第 3 部　モデリングとトレーサビリティ

AI 時代の要求管理では、要求を文章として書くのみでは不十分である。社会や組織の活動は、業務プロセス、データ、AI モデル、ソフトウェア、法令、標準、社内規則、契約、運用ルールなどが複雑に関係して成り立っており、これらを自然言語の文書のみで管理しようとすると、関係者間で解釈がずれ、変更の影響範囲が見えにくくなり、AI やツールによる活用も難しくなるためである。

このため、AI 時代の要求管理では、要求を可視化し、構造化し、関係性を追跡可能にすることが重要である。その中心となるのが、モデリングとトレーサビリティである。

モデリングとは、業務、システム、データ、ルール、リスク、評価などを、図や表、構造化された形式で表すことである。モデリングにより、関係者は複雑な対象を共通の形で理解し、議論し、確認できる。さらに、モデルが機械可読な形で整備されれば、AI やツールを使った分析、シミュレーション、影響分析、テスト、監査にも活用できる。

トレーサビリティとは、目的、要求、要件、モデル、データ、設計、コード、テスト、運用ログ、改善要求の関係を追跡できるようにすることである。トレーサビリティが確保されていれば、ある要件がなぜ必要か、どこに実装されているか、どのテストで確認されるか、変更した場合にどこへ影響するかなどを把握できる。

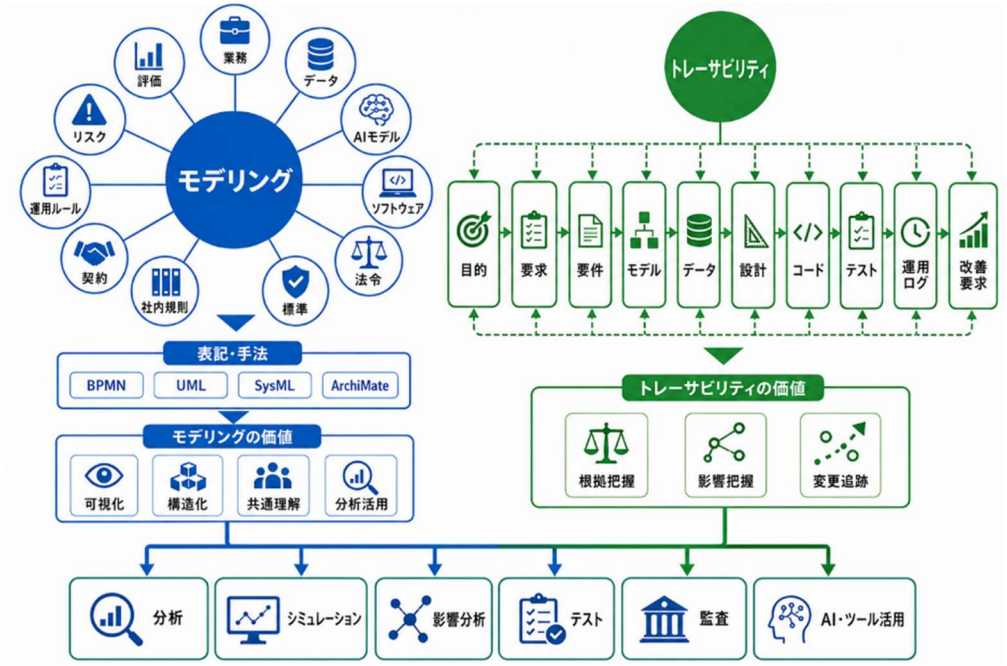


図　16　AI 時代の要求管理におけるモデリングとトレーサビリティ

第 3 部では、AI 時代の要求管理において、なぜモデリングとトレーサビリティが重要になるのか（第 9 章、第 11 章）と、モデルをシステムの実現につなげる手段の一例としてのビルディングブロックの活用（第 10 章）について整理する。

第 9 章　要求を可視化する

9.1　なぜ要求を可視化するのか

要求定義の現場では、関係者が同じ言葉を使っていても、実際には異なる意味で理解していることが多い。経営層、業務部門、IT 部門、AI 担当、データ担当、法務、監査、ベンダーは、それぞれ異なる関心や前提を持っている。文章のみで要求を共有すると、読み手ごとに解釈が異なり、後工程で認識違いが明らかになることがある。

例えば、「顧客情報を適切に管理したい」という要求があったとする。この表現のみでは、何をもって適切とするのかが分からない。顧客情報とはどのデータ項目を指すのか。誰が閲覧できるのか。どのシステムで保管するのか。AI に入力してよいのか。保存期間はどれくらいか。削除条件は何か。アクセスログは必要か。法令や契約上の制約は何か。こうした点を明らかにしなければ、実装も評価もできない。

可視化の目的は、単に見やすい図を作ることではない。関係者が同じ対象を同じ構造で理解し、議論し、検証できるようにすることである。業務フローを描けば、誰が、いつ、何を判断し、どのデータを使い、どの例外処理があるかが見える。データモデルを描けば、データ項目の意味、関係、重複、欠落が見える。システム構成図を描けば、AI、アプリケーション、データベース、外部 API、認証基盤、ログ基盤の接続関係が見える。リスクモデルを描けば、どの脅威がどの業務やデータに影響するかが見える。

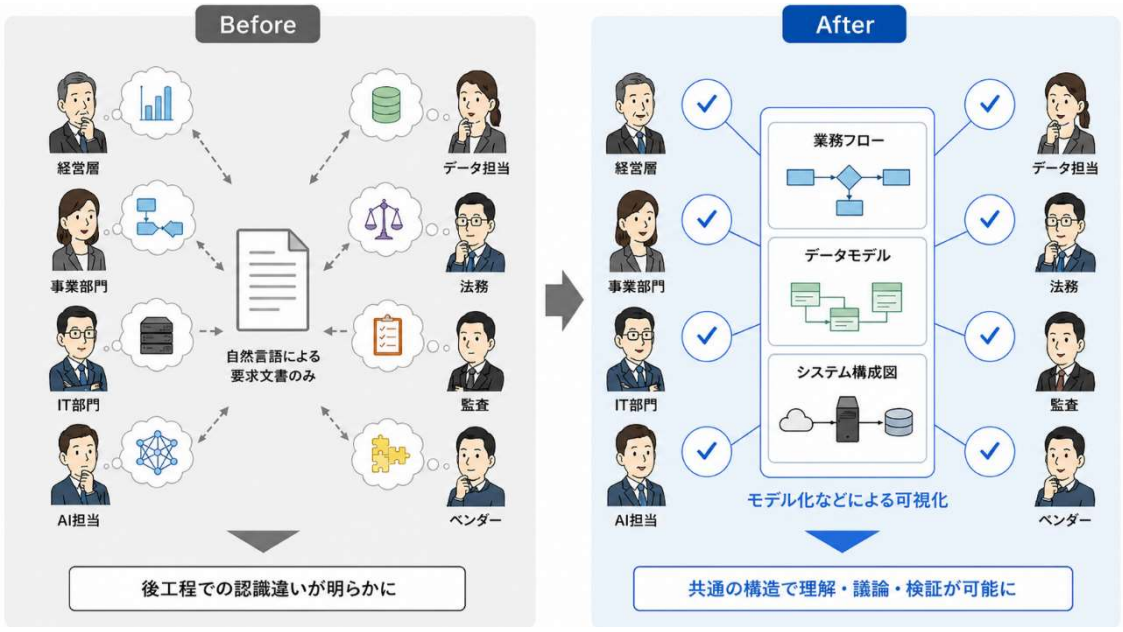


図 17　要求の可視化の重要性

AI 時代には、可視化の重要性がさらに高まる。AI の出力は、データ、モデル、プロンプト、参照文書、運用環境に依存する。そのため、AI をどこで使い、何を入力し、何を出力し、人間がどこで確認し、どのログ

を残し、どのように評価するのかを可視化する必要がある。これができていないと、AI の誤出力や事故が発生したときに、原因究明や改善が難しくなる。

9.2 Human Readable から Machine Understandable へ

従来の要求定義書は、人間が読むことを前提に作られてきた。自然言語で背景、目的、機能、制約、画面、帳票、非機能要件などを記述し、関係者がレビューする形である。しかし、人間が読めることと、複雑な業務や要求、その関係性を十分に理解できることは同じではない。文章のみでは、関係者によって解釈が異なったり、要求間の矛盾や抜け漏れを見落としたりする可能性がある。
そのため、まずは業務、要求、判断基準、データなどをモデル化して可視化し、関係者が共通の理解を持ち、議論・合意できるようにすることが重要である。そのうえで、合意された内容を AI やツールによる分析、整合性確認、変更管理などに活用できるよう、必要に応じて機械可読な形式で構造化することが求められる。このような観点から、要求の記述・共有方法は、次のように整理できる。

表 32 要求文書の記載レベル

	意味	例
Human Readable	人間が読める	自然言語の要求定義書
Human Understandable	人間が理解しやすい	図、表、業務フロー、用語集
Machine Readable	機械が読み取れる	構造化データ、タグ、ID、メタデータ
Machine Understandable	機械が意味を扱える	データモデル、オントロジー、ルールモデル、トレーサビリティ

ただし、Human Readable な文書は今後も変わらず必要である。要求の妥当性や優先順位、リスクの許容範囲などについて判断し、合意するためには、関係者がその内容や根拠を理解できる必要があるためである。一方、人間の理解のみを目的とした文書では、AI やツールを使った影響分析、整合性確認、変更管理、テスト生成、監査支援には限界がある。そのため、人間による理解・合意形成と、AI やツールによる処理の双方を支えることが重要である。
例えば、要求に ID を付与し、発生源、上位目的、関連法令、関連データ、関連機能、検証方法をメタデータとして持たせれば、ツールで検索・集計・影響分析ができる。業務フローを BPMN などの標準的な形式で表せば、業務の流れや例外処理を機械的に解析しやすくなる。データモデルやオントロジー[23]を

[23] オントロジーとは、システム、要求、要件、機能、構造、振る舞い、インターフェース、検証などの概念と関係を、共有可能かつ機械処理可能な形で明示した意味モデルである。Gruber による定義 [33]も参照されたい。

用いれば、同じ言葉が異なる意味で使われていないかを確認できる。
Machine Understandable に近づけることは、要求を機械に丸投げすることではない。むしろ、人間の判断を支援するために、要求の構造、意味、関係を明確にすることである。AI 時代の要求管理では、人間が理解でき、機械も扱える形を目指す必要がある。

9.3 代表的なモデル・モデリング手法・構造化手法

モデリング・構造化には多様な手法がある。すべてを使う必要はなく、目的に応じて使い分けることが重要である。

表 33 代表的なモデル・モデリング手法・構造化手法

	主な用途	活用場面
業務フロー	業務手順、役割、例外処理	業務改善、要求定義、教育
BPMN	標準的な業務プロセス記述	手続、承認、部門間連携
UML	ユースケース、クラス、シーケンス	アプリケーション設計
SysML	要求、要件、構造、振る舞い、検証	複雑なシステム、MBSE
ArchiMate	業務、アプリ、技術基盤の関係	エンタープライズアーキテクチャ
ER 図	データ項目と関係	データベース、データ連携
データモデル	データ定義、制約、コード体系	AI、データ基盤、相互運用性
オントロジー	用語、概念、意味関係	分野横断、ルール、AI 活用
ルール表	条件と処理の対応	業務ルール、審査、判定
リスクマトリクス	リスクの発生可能性と影響	安全性、セキュリティ、AI リスク
トレーサビリティマトリクス	要件と実装・評価の対応	変更管理、監査、説明責任

例えば、BPMN は業務プロセスを標準的に記述するのに適している。承認、分岐、例外、部門間の受け渡しを整理する場合に有効である。UML は、ユースケース、クラス、シーケンスなど、アプリケーションの構造や振る舞いを整理するのに向いている。SysML は、要求、要件、構造、振る舞い、検証の関係を扱えるため、複雑なシステムや MBSE に向いている。ArchiMate は、組織、業務、アプリケーション、技術基盤を横断的に整理する場合に有効である。
AI 時代には、データモデルとオントロジーも重要になる。AI はデータの意味に依存するため、用語やデータ項目の意味が曖昧だと、誤った判断や不適切な出力につながるためである。特に、法令、標準、業務、データが接続される領域では、用語の意味や参照関係を明確にすることが重要である。

9.4 建築や製造から学ぶ可視化

建築や製造の分野では、設計図、CAD、BIM、デジタルモックアップ、シミュレーションが広く使われてい

る。施主、設計者、施工者、設備担当、監理者は、図面や 3D モデルを使って完成物のイメージを共有し、事前に確認し、変更の影響を把握する。
住宅建築を考えると分かりやすい。施主は、完成した家を直接見る前に、間取り図、外観図、3D パース、設備仕様、見積、工程表を確認する。最近では、3D ウォークスルーにより、完成後の空間を体感することもできる。耐震性や日当たり、動線もシミュレーションできる。これにより、施主と設計者は完成物のイメージを共有し、早い段階で認識違いを減らすことができる。
一方、ソフトウェアや AI の要求定義では、完成物のイメージが共有されにくい。業務改革と並行してシステムを作る場合、発注者自身も完成形を明確に描けていないことが多い。さらに、AI システムでは、データ、モデル、判断、運用、監査が絡むため、自然言語の要求定義書のみでは全体像を把握しにくい。
だからこそ、ソフトウェアや AI でも、建築や製造と同じように、モデルを使った可視化、シミュレーション、変更管理、再利用を進める必要がある。要求定義は、文章で説明することに加え、関係者が見て、確認し、体感し、合意できる形にすることが重要である。

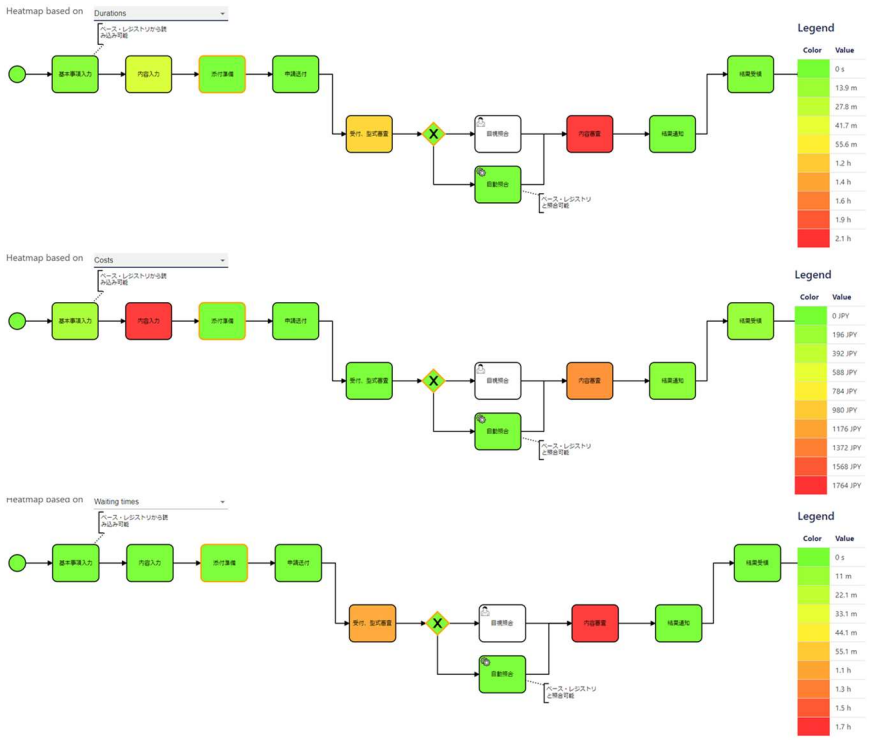


図 18 BPMN によるモデルを使ったシミュレーション例（時間、コスト、待ち時間）

9.5 要求情報とモデルを接続する

AI 時代の要求管理では、要求情報を特定の文書や特定のツールに閉じ込めないことが重要である。要求は、発注者、受注者、システム部門、ベンダー、監査部門、テスト担当、モデリング担当、運用担当の間で共有される。また、要求は単独で存在するものではなく、設計、モデル、コード、テスト、変更要求、

障害情報、運用ログ、監査証跡と結びついて管理される。
業務フロー、データモデル、システム構成図、SysML モデルなどを作成しても、それぞれのモデルがどの要求に対応しているかが不明確であれば、要求の実現状況や変更の影響範囲を十分に確認することはできない。そのため、要求 ID と業務フロー、データモデル、設計モデル、テストケースなどを関連付け、要求情報とモデルの対応関係を追跡できるようにすることが重要である。
要求情報とモデルを接続することで、ある要求を変更した場合に、どの業務、どのデータ、どの設計、どのテスト、どの運用手順に影響するかを確認しやすくなる。また、モデルから要求へ遡ることで、そのモデル要素が何のために存在し、どの要求を実現するものかを説明しやすくなる。
要求情報をツール間で交換し、要求と設計、テスト、変更要求などのライフサイクル情報を連携するための標準的な形式及び考え方として、ReqIF と OSLC がある。これらについては第 11.6 節で説明する。

9.6 モデリングを導入する現実的な進め方

モデリングの重要性は理解されていても、現場ではなかなか定着しない。理由として、専門人材がいない、ツールが使いにくい、作成コストが高い、メリットが分かりにくい、既存の文書文化から移行できない、といった課題がある。
そのため、モデリングは一気に全面導入するのではなく、次のように段階的に進めることが望ましい。

1. 重要業務の業務フローの可視化
   まずは、対象業務の開始から終了までを図にし、誰が、何を、どの順番で行うのかを確認する。例外処理、承認、差戻し、手作業、二重入力などを見えるようにする。
2. 主要データ項目の整理
   業務で使うデータ、AI に入力するデータ、外部連携するデータ、法令や規則で保存が求められるデータを一覧化し、意味、形式、出典、品質、保存期間を定義する。
3. 要件とテスト、要件とデータ、要件と業務プロセスの対応関係の整理
   すべてを完璧に接続する必要はない。まずは、重要な要件について、どの業務、どのデータ、どのテストで確認するのかを明確にする。
4. 変更影響分析に利用
   制度変更、業務変更、AI モデル更新、データ項目変更が発生したときに、どの要件、どのシステム、どのテスト、どの運用手順に影響するかを確認できるようにする。
5. AI やツールによる支援を取り入れる
   自然言語の文書から業務フローやデータ項目を抽出する、既存仕様書から要求一覧を作る、モデル間の不整合を検出する、変更影響を候補として提示する、といった活用が考えられる。

モデリング導入の目的は、きれいな図を作ることではない。関係者の共通理解を作り、抜け漏れを減らし、変更に強くし、AI やツールによる支援を受けやすくすることである。

9.7 形式手法による要求の厳密化と検証

AI 時代の要求管理では、要求や要件を分かりやすく可視化することに加え、重要な部分については、より厳密に記述し、矛盾や抜け漏れを検証できるようにすることが重要である。そのための方法の一つが、形式手法である。

形式手法とは、数学的または論理的な記法を用いて、システムの要求、仕様、状態、振る舞い、制約を厳密に記述し、その性質を検証するための方法である。自然言語による要求定義は人間にとって読みやすい一方で、曖昧さ、解釈の違い、矛盾、抜け漏れが残りやすい。形式手法はすべての要求を数学的に記述するためのものではなく、特に安全性、セキュリティ、コンプライアンス、重要な業務判断、状態遷移、権限管理、例外処理など、誤りが大きな影響を及ぼす領域で要求の厳密性を高めるために用いる。

例えば、「利用者は適切な権限を持つ場合のみ、顧客情報を閲覧できること」という要件は、一見明確に見える。しかし、実際には、利用者の役割、所属、担当顧客、委任、例外承認、退職・異動、緊急時対応、ログ記録など、多くの条件が関係する。自然言語のみで記述すると、例外条件や優先順位が曖昧になりやすい。形式的な記述やルール表、状態遷移モデルを用いれば、どの条件でアクセスを許可し、どの条件で拒否し、どの条件で承認を求めるのかを明確にできる。

形式手法が有効な領域には、次のようなものがある。

表 34 形式手法が有効な領域

	形式手法が有効な理由
安全性要求	危険状態に遷移しないことを確認できる
セキュリティ要求	権限、アクセス制御、情報流通制約を厳密に扱える
業務ルール	条件、例外、優先順位、禁止事項を明確にできる
状態遷移	申請、承認、差戻し、取消し、終了などの状態を検証できる
並行処理	複数処理の競合、順序、デッドロックを確認できる
AI 利用制限	AI が判断してよい範囲、人間に委ねる範囲を明確にできる
フェイルセーフ	異常時、安全側に倒れる条件を検証できる
コンプライアンス	法令・規則に基づく義務、禁止、例外を構造化できる

形式手法は、要求すべてを難解な数式に置き換えるものではない。実務では、「軽量な形式手法」として、状態遷移図、決定表、ルール表、制約条件、時系列の振る舞い、入出力条件を明確にするところから始めることが現実的である。例えば、申請業務に対し、申請中、審査中、差戻し、承認済、却下、取消しといった状態を定義し、どの操作でどの状態に遷移できるかを整理する。この整理のみでも、例外処理、二重承認、取消し後の再申請、承認後の修正可否などの曖昧さを減らすことができる。

より高度な領域では、モデル検査、定理証明、形式仕様記述、時間論理、制約充足などの手法を使うこともある。例えば、重要な制御システム、金融取引、本人確認、アクセス制御、データ連携、法令遵守が厳しく求められる業務では、要求や仕様が特定の性質を満たしているかを機械的に確認することが有

効である。代表的な性質としては、「禁止された状態に到達しないこと」「必要な承認を経ずに処理が完了しないこと」「権限のない利用者が情報にアクセスできないこと」「障害時に安全な状態へ移行すること」などがある。

AI システムにおいても、形式手法は重要である。ただし、AI モデルそのものの出力を完全に形式的に証明することは難しい場合が多い。AI は確率的に振る舞い、学習データや利用環境に依存するためである。そのため、AI 時代の形式手法では、AI モデルの内部をすべて証明するというよりも、AI を取り巻くシステム全体の安全制約、利用範囲、人間へのエスカレーション条件、フェイルセーフ条件、ログ要件、権限管理、データ利用制約を形式的または半形式的に記述・検証することが現実的である。

例えば、生成 AI を使った社内問い合わせシステムでは、次のような制約を形式的に扱うことができる。

- 承認済み文書に根拠が存在しない場合、AI は確定的な回答を返してはならない
- 法務、人事、懲戒、契約解除に関する質問は、専門部署へのエスカレーション対象とする
- 機密情報を含む入力は、外部 AI サービスに送信してはならない
- 確信度が一定値を下回る場合、AI 出力は参考情報として扱い、人間の確認を必須とする
- 高リスク回答を利用者に提示する場合、根拠文書、確認者、確認日時を記録する

このような制約を明確にしておけば、AI が自由に回答する範囲と、人間が確認すべき範囲を分けやすくなる。また、後から監査や事故調査を行う場合にも、どの要件に基づいて AI の動作を制限していたのかを説明しやすい。

形式手法は、継続的アシュアランスとも関係する。AI システムや業務システムが運用後に更新される場合、変更のたびに、安全性、セキュリティ、コンプライアンスが維持されているかを確認する必要がある。形式的な制約やモデルが整備されていれば、仕様変更、ルール変更、モデル更新、データ変更が発生したときに、既存の制約を破っていないかを確認しやすくなる。これは、手作業のレビューのみに依存するよりも、継続的な品質保証を実現しやすい。

形式手法を要求定義に取り入れる場合、次のような進め方が現実的である。

1. 形式化すべき対象を選ぶ

   すべての要求を形式化する必要はない。安全性、セキュリティ、法令遵守、AI 利用制限、権限管理、重要な状態遷移など、誤りの影響が大きい要求を優先する。

2. 自然言語の要求を整理する

   形式化の前に、目的、条件、例外、禁止事項、責任者、評価方法を明確にする必要がある。自然言語の要求が曖昧なままでは、形式化しても意味のある検証はできない。

3. 軽量なモデルから始める

   状態遷移図、決定表、ルール表、制約一覧、入出力条件など、人間が理解しやすい形で記述する。必要に応じて、モデル検査や形式仕様記述へ進む。

4. 要求、モデル、テスト、監査証跡をつなぐ

   形式化した制約が、どの要求に対応し、どの設計やテストで確認され、運用後にどのログで監視されるのかをトレーサビリティで管理する。

5. 変更時に再検証する

法令、業務ルール、AI モデル、データ、プロンプト、システム構成が変わった場合、形式化された制約に影響がないかを確認する。

形式手法を導入する際には、専門性の高さにも注意が必要である。高度な形式手法は、専門知識を必要とし、すべての現場で直ちに利用できるわけではない。そのため、組織としては、まず軽量な形式化を普及させ、重要領域について専門家が支援する体制を整えることが望ましい。形式手法を特定の専門家に限った活動にすると、現場から切り離されてしまう。要求工学、MBSE、テスト、セキュリティ、監査と接続し、実務上の価値が見える形で導入することが重要である。

AI 時代の要求管理では、自然言語、モデル、データ、ルール、テスト、運用ログをつなぐことが求められる。その中で形式手法は、曖昧さを減らし、重要な制約を検証し、変更後も安全性や信頼性を維持するための有力な手段である。すべてを形式化する必要はないが、重要な要求については、形式的または半形式的に記述し、検証可能な状態にすることが、AI 時代の要求管理をより強固なものにする。

第 10 章　ビルディングブロックやサービスによるシステムの実現

10.1 モデリングから実現手段へ

要求管理においてモデリングを行う意義は、業務やシステムを分かりやすく可視化することに限らない。モデリングにより、実現すべき機能、データ、業務プロセス、ルール、インターフェース、評価条件を分解し、それぞれをどのような単位で実現するかを考えやすくなる。本章では、その実現手段を検討する際の選択肢の一つとして、ビルディングブロックや既存サービスの活用を取り上げる。

複雑なシステムや業務をそのまま一つの大きな仕組みとして作ろうとすると、設計も開発も運用も難しくなる。変更が必要になった場合にも、どこを修正すればよいか分かりにくい。特に AI 時代には、技術、データ、外部サービス、規制、セキュリティリスクが短い周期で変化するため、一度作った仕組みを長期間そのまま使い続けることは難しい。

そこで重要になるのが、機能やサービスを「ビルディングブロック」として捉える考え方である。ビルディングブロックとは、システムや業務を構成する部品であり、一定の役割、入力、出力、責任範囲、インターフェースを持つ単位である。モデリングによって、業務やシステムを適切な単位に分解できれば、それぞれの機能をどのように実現するかを柔軟に選択できる。

例えば、本人確認、認証、決済、通知、検索、データ連携、ログ管理、AI による分類、レポート作成、権限管理などは、それぞれビルディングブロックとして捉えることができる。これらを一つずつ内製で開発する場合もあれば、OSS を利用する場合もある。商用アプリケーションやクラウドサービス、SaaS、API サービスを組み合わせて実現する場合もある。

要求・要件をシステムとして実現する際には、「何を実現したいか」を明確にしたうえで、「それをどの単位に分け、どう実現するか」を考えることが重要である。その際、既存のビルディングブロックやサービスの活用可能性を検討し、必要に応じて新規開発と組み合わせることが有効である。



図 19 ビルディングブロックのイメージ

10.2 ビルディングブロックとは何か

ビルディングブロックとは、システムや業務を構成する再利用可能な機能単位である。建築でいえば、部材、設備、配管、電気系統、部屋、構造体などが組み合わさって建物を構成するように、情報システムや AI システムも、複数の機能やサービスの組み合わせで構成される。
ビルディングブロックには、さまざまな大きさがある。非常に小さな単一機能の場合もあれば、業務サービス全体を担う大きなサービスの場合もある。

表 35 ビルティングブロックの粒度

	例	説明
単一機能	認証、通知、ログ出力、データ変換	特定の機能のみを担う小さな部品
共通機能	権限管理、検索、帳票出力、ワークフロー	複数業務で共通利用される機能
データ連携機能	API 連携、ファイル連携、データ変換、コード変換	システム間・組織間のデータ連携を担う機能
AI 機能	分類、要約、翻訳、異常検知、回答生成	AI モデルや AI サービスを使った機能
業務サービス	顧客管理、契約管理、購買管理、経費精算	業務単位で提供されるアプリケーションや SaaS
基盤サービス	ID 管理、監査ログ、セキュリティ監視、データ基盤	組織全体で共通利用される基盤機能

重要なのは、ビルディングブロックの大きさを固定的に考えないことである。あるプロジェクトでは、通知機能を小さなビルディングブロックとして扱う場合がある。別のプロジェクトでは、問い合わせ対応全体を一つの SaaS として導入し、その中に FAQ、回答生成、チケット管理、分析機能が含ませる場合もある。

最初から「すべてを開発する」「すべてを既製サービスで置き換える」と決めるのではなく、要求を分解し、それぞれのビルディングブロックについて、内製、OSS、商用製品、SaaS、外部 API、既存システムの活用など、最適な実現手段を検討することが重要である。

10.3 機能ブロックの実現方法

モデリングによって機能ブロックが明確になると、それぞれの機能をどのように実現するかを選択できる。実現方法は一つではない。主な選択肢は次のとおりである。

表 36 機能ブロックの実現方法

実現方法	内容	向いている場合
新規開発	自社要件に合わせてコードを書く	差別化領域、特殊要件、高度な制御が必要な場合
既存システム活用	既存機能を再利用・改修する	既に安定運用され、要件に合う場合
OSS 活用	オープンソースソフトウェアを利用する	標準的機能、透明性、拡張性が必要な場合
商用アプリケーション	パッケージ製品を導入する	一般的な業務機能を短期間で導入する場合
SaaS 利用	クラウドサービスとして利用する	標準業務、迅速導入、運用負荷軽減が必要な場合
外部 API 利用	特定機能を API で利用する	認証、決済、翻訳、地図、AI などを外部サービスで使う場合
AI サービス利用	生成 AI、分類 AI、異常検知 AI などを利用する	高度な AI 機能を短期間で利用する場合

例えば、データ連携機能を考える。単純なファイル連携で十分な場合もあれば、API ゲートウェイを使う場合もある。国際的なデータ連携や高頻度なリアルタイム連携が必要であれば、メッセージング基盤やデータスペース基盤を検討する必要がある。要件が異なれば、選ぶべきビルディングブロックも変わる。
同様に、AI 機能についても、自社でモデルを開発する場合、既存の AI モデルをファインチューニングする場合、外部の生成 AI サービスを利用する場合、商用 AI アプリケーションを導入する場合がある。重要なのは、「AI を使う」という抽象的な要求実現ではなく、どの業務で、どの入力に対して、どの出力を求め、どの品質で、どの責任範囲で利用するかを明確にしたうえで、要求を真に実現する方法を選ぶことである。

10.4 ビルディングブロック化の利点

ビルディングブロック化には、既存機能の再利用による開発生産性の向上、実績のある部品の活用による品質の安定化、そして、変更容易性の向上などの利点がある。特に、変更に強くなることは最大の利点である。システム全体が一体化している場合、ある機能を変更するのみでも広い範囲に影響が及ぶ。一方、機能が適切に分離され、インターフェースが明確であれば、特定のブロックのみを交換したり、改善したりすることができる。

例えば、ある時点では自社開発の検索機能を使っていたが、後からより高性能な商用検索サービスが登場した場合、検索機能がビルディングブロックとして分離されていれば、比較的容易に置き換えを検討できる。AI による分類機能についても、当初は簡易なルールベース分類で開始し、その後、より高精度な AI モデルや商用 AI サービスに置き換えることができる。

以上も踏まえて、ビルディングブロック化には次のような利点がある。

- 技術更新に対応しやすい
  AI、セキュリティ、データ連携、クラウドサービスは急速に進化する。機能が分離されていれば、新しい技術やサービスが登場したときに、必要な部分のみを更新できる。
- 目的に合ったサービスを選びやすい
  同じ機能でも、安さを重視する場合、高性能を重視する場合、セキュリティを重視する場合、規制対応を重視する場合で、選ぶべきサービスは異なる。ビルディングブロックとして要件を整理しておけば、目的に合った選択がしやすい。
- 再利用によって生産性や品質を高めやすい
  認証、通知、ログ、データ連携、承認ワークフローなどは、複数のシステムで共通利用できる。共通ビルディングブロックとして整備すれば、個別システムごとに同じ機能を作り直す必要がなくなり、開発の効率化や品質の安定化につなげることができる。
- 調達や委託がしやすい
  機能単位で要件、インターフェース、品質、責任範囲を定義しておけば、外部サービスや委託先を比較しやすい。SLA や受入条件も明確になる
- 障害やインシデント時の影響範囲を把握しやすい
  どのビルディングブロックがどの業務、データ、システムに影響するかが分かっていれば、原因究明や代替手段の検討が速くなる

ビルディングブロック化は、単なる技術設計の話ではない。要求定義の段階で、機能の役割、境界、入力、出力、責任範囲、評価方法を明確にすることにより、将来の変更可能性を確保するための考え方である。

10.5 セキュリティ対策におけるビルディングブロック

ビルディングブロックは、セキュリティ対策においても有効である。セキュリティは、システム全体に関わる横断

的な要件であるが、実際には認証、認可、暗号化、ログ、監視、脆弱性管理、インシデント対応など、複数の機能ブロックによって実現される。
例えば、認証機能を個別システムごとに作り込むのではなく、共通 ID 基盤や認証サービスとして整備すれば、セキュリティ水準を統一しやすくなる。アクセスログも、各システムが独自に保存するのではなく、共通ログ基盤に集約すれば、監査や異常検知がしやすくなる。脆弱性管理やセキュリティ監視についても、共通サービスとして整備すれば、組織全体の対応力が高まる。
以上も踏まえて、セキュリティに関する主なビルディングブロックには、次のようなものがある。

表 37 セキュリティに関する代表的なビルディングブロック

	ビルディングブロックの例
認証	ID 管理、多要素認証、シングルサインオン
認可	権限管理、ロール管理、属性ベースアクセス制御
暗号化	通信暗号化、保存データ暗号化、鍵管理
ログ管理	アクセスログ、操作ログ、AI 利用ログ
監視	異常検知、セキュリティ監視、アラート
脆弱性管理	SBOM、脆弱性スキャン、パッチ管理
インシデント対応	通報、封じ込め、復旧、報告、再発防止
データ保護	マスキング、匿名化、利用制御、持ち出し制御

AI 時代には、AI 利用そのものにもセキュリティ対策が必要になる。例えば、機密情報を生成 AI に入力しないための入力制御、プロンプトインジェクション対策、AI 出力の検証、AI 利用ログの保存、モデル更新時の検証などである。これらもビルディングブロックとして整理し、各システムで共通に利用できるようにすれば、個別対応よりも安全性と効率性を高めることができる。

10.6 ビルディングブロックを選定するための要件

ビルディングブロックを選定する際には、単に機能があるかどうかのみを見てはいけない。特に、OSS や商用アプリケーション、SaaS、外部 API を利用する場合には、機能要件に加え、非機能要件、運用要件、契約要件、セキュリティ要件、データ要件を確認する必要がある。
主な確認観点は次のとおりである。

表 38 ビルディングブロックを選定するための主な確認観点

	確認内容
機能適合性	必要な機能を満たしているか
性能	処理速度、同時利用数、応答時間を満たすか
可用性	障害時に業務を継続できるか

拡張性	利用者数やデータ量の増加に対応できるか
相互運用性	API、データ形式、標準に対応しているか
セキュリティ	認証、権限、暗号化、ログ、脆弱性管理は十分か
データ管理	データの保存場所、所有権、利用条件、削除条件は明確か
コンプライアンス	法令、業界標準、社内規則に適合するか
運用性	監視、保守、障害対応、サポート体制は十分か
変更可能性	将来、他サービスへ置き換え可能か
コスト	初期費用、運用費用、ライセンス、従量課金は妥当か
ベンダーロックイン	特定事業者への依存が過度でないか
監査可能性	証跡、レポート、監査対応が可能か

特に AI サービスや SaaS を利用する場合、データの扱いを確認することが重要である。入力データが学習に使われるのか、どの国・地域に保存されるのか、ログはどれくらい保存されるのか、削除要件に対応できるのか、障害や情報漏えい時にどのように通知されるのかを要件として明確にする必要がある。
また、OSS を利用する場合には、ライセンス、メンテナンス状況、脆弱性対応、コミュニティの継続性、自社での保守能力を確認する必要がある。OSS は有効な選択肢であるが、導入後の管理責任を明確にしなければ、脆弱性や保守停止のリスクが残る。

10.7 ビルディングブロックと標準化・相互運用性

ビルディングブロック化を有効にするためには、インターフェースとデータの標準化が重要である。機能が分離されていても、接続方法が独自仕様であれば、交換や再利用は難しい。API 仕様、データ形式、認証方式、ログ形式、エラーコード、メタデータを標準化することで、ビルディングブロックを組み合わせやすくなる。
例えば、顧客管理、契約管理、請求、問い合わせ対応、AI 分析の各サービスを連携させる場合、顧客 ID、契約 ID、商品コード、問い合わせ番号などのキー情報が統一されていなければ、データ連携は難しい。同じ「顧客」という言葉を使っていても、部門ごとに意味が異なれば、AI 分析の結果も不安定になる。
したがって、ビルディングブロック化には、次のような標準化が必要である。

表 39 ビルディングブロック化の標準化対象の例

	内容
API	入出力、認証、エラー処理、バージョン管理
データ	項目定義、コード体系、データ品質、正本
認証・認可	ID、ロール、権限、属性
ログ	操作ログ、アクセスログ、AI 利用ログ、監査ログ

メタデータ	サービス説明、データ説明、利用条件、更新頻度
評価基準	性能、品質、セキュリティ、AI 評価指標
契約条件	SLA、責任分担、データ利用、終了時対応

標準化は、自由度を奪うものではない。むしろ、標準化により、個別機能を自由に組み合わせやすくなる。共通のインターフェースやデータ定義があれば、新しいサービスやより高性能な部品が出てきたときに、置き換えやすくなる。

10.8 ビルディングブロック化と調達・契約

ビルディングブロック化は、調達や契約のあり方にも影響する。従来は、大きなシステムを一括で委託し、ベンダーがまとめて設計・開発・運用する形が多かった。しかし、ビルディングブロック化を進めると、機能単位、サービス単位、API 単位で調達や契約を行うことが増える。
この場合、要求管理では、各ビルディングブロックの責任範囲を明確にする必要がある。どの機能を提供するのか、どの入力を受け取り、どの出力を返すのか、性能や可用性はどの水準か、障害時の責任はどこにあるのか、データの管理責任は誰にあるのかを定義する。
また、複数のビルディングブロックを組み合わせる場合、個々のサービスが正常でも、全体として期待どおりに動かないことがある。したがって、調達・契約では、個別サービスの SLA に加え、サービス間連携、全体性能、障害時の切り分け、統合テスト、運用責任分担を明確にする必要がある。
AI サービスを利用する場合には、さらに、モデル更新、出力品質、データ利用、説明責任、利用ログ、インシデント通知、サービス終了時のデータ返却・削除も確認する必要がある。

10.9 ビルディングブロック化の注意点

ビルディングブロック化は有効であるが、次のような注意点もある。

- 分割しすぎると複雑になる
  細かく分ければ柔軟性は高まるが、連携点が増え、管理が難しくなる。どの粒度で分けるかは、業務の安定性、変更頻度、リスク、再利用可能性を踏まえて判断する必要がある。
- インターフェース設計が重要になる
  ビルディングブロック間の接続が曖昧だと、障害時の責任分担やデータ不整合が問題になる。入力、出力、エラー処理、バージョン管理、ログを明確に定義する必要がある。
- ベンダーロックインに注意する必要がある
  商用サービスや SaaS は便利であるが、独自仕様に依存しすぎると、将来の置き換えが難しくなる。データのエクスポート、API、契約終了時の対応を確認する必要がある。
- セキュリティと責任分担を明確にする必要がある
  複数のサービスを組み合わせると、どこで認証し、どこでログを保存し、どこで監視し、どこが障害対応するのかが不明確になりやすい。

- 全体最適を忘れてはならない
  個々のビルディングブロックが優れていても、全体として業務目的に合っていなければ意味がない。要求管理では、個別機能に加え、業務全体、利用者体験、運用、監査、変更管理まで見て判断する必要がある。
- SBOM の導入を考える必要がある
  ビルディングブロック化によってシステムの再利用性・交換可能性が高まる一方、構成や依存関係は複雑化する。このため、各ブロックを構成するソフトウェア部品とその依存関係を SBOM によって機械可読な形で把握し、脆弱性対応、影響分析、ライセンス管理、サプライチェーン管理につなげられるよう、あらかじめ考慮しておく必要がある。

第 11 章　トレーサビリティと DevOps 型要求管理

11.1 なぜトレーサビリティが重要か

AI 時代の要求管理では、トレーサビリティが極めて重要である。トレーサビリティとは、要求がどこから来て、どの要件になり、どの設計やコードに反映され、どのテストで確認され、運用後にどのように評価されるのかを追跡できる状態を指す。
トレーサビリティがない場合、次のような問題が起きる。

- なぜその機能が必要なのか分からない
- どの法令や標準に基づく要件なのか分からない
- ある要件がどこに実装されているのか分からない
- 変更時にどこへ影響するのか分からない
- テストがどの要件を確認しているのか分からない
- AI の出力がどのデータやルールに基づくのか説明できない
- 障害や事故の原因究明に時間がかかる
- 運用データを改善に結びつけられない

AI がバグや不整合を見つけたとしても、要求定義書がなく、コードが複雑化しており、要件と実装の対応関係が分からなければ、修正には時間がかかる。AI でパッチを作ることはできても、どの要件を満たすべきか、どの業務や法令に影響するかが分からなければ、安心して修正を適用することはできない。
逆に、要求、要件、設計、コード、テスト、運用ログがつながっていれば、AI は影響分析や修正案の作成を支援しやすくなる。トレーサビリティは、人間の説明責任に加え、AI による開発支援や運用改善の前提でもある。

11.2 何をつなぐべきか

トレーサビリティでは、単に要求、要件とコードをつなげばよいわけではない。AI 時代には、目的、要求、

要件、モデル、データ、AI モデル、ルール、評価、運用までをつなぐことが望ましい。
つなぐ対象の基本的な流れは次のとおりである。

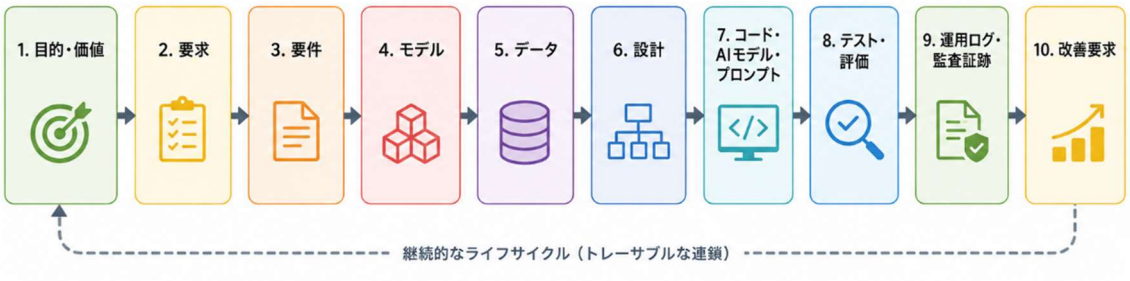


図 20 トレーサビリティのためにつなぐ対象の流れ

この流れを明確にしておくことで、各要求、要件の意味と影響範囲を把握できる。
特に AI 時代には、要求や要件の次の関係を管理することが重要である。

表 40 AI 時代の要求や要求のトレーサビリティ確保のためにつなぐ対象

	目的
目的と要求	要求が何の価値を実現するのかを明確にする
要求と要件	抽象的な要求を実装・評価可能にする
要件と業務プロセス	要件がどの業務で実行されるかを確認する
要件とデータ	必要なデータ、品質、出典を明確にする
要件と AI モデル	AI が何を担うのかを明確にする
要件とプロンプト	生成 AI への指示を管理する
要件と法令・標準	コンプライアンス根拠を明確にする
要件とテスト	どのテストで確認するかを明確にする
要件とログ	運用後に何を観測するかを明確にする
ログと改善要求	運用データを改善に接続する

このようなつながりがあると、変更への対応が容易になる。例えば、法令が変わった場合、その法令に基づく要求、関連する業務プロセス、データ項目、AI 利用条件、テスト、監査証跡を確認できる。AI モデルを更新する場合は、そのモデルが関係する要求、評価指標、利用業務、利用者、ログ、リスクを確認できる。
ただし、すべてのプロジェクトにおいて、これらの対応関係を一律に網羅する必要はない。プロジェクトの規模や特性、要求の重要度やリスクに応じて管理対象と粒度を定め、まずは主要な要求とその根拠、対応する要件、検証方法などから関連付けることが考えられる。一方、安全性や法令遵守などに関わる重要な要求については、必要なトレーサビリティを確実に確保することが求められる。

なお、特にデータに関しては「7.4 データ供給チェーンの透明性とトレーサビリティ」に記載した関係も整理するとよい。

11.3 DevOps 型の要求管理

AI 時代の要求管理では、DevOps 型のフィードバックが重要になる。従来は、要求定義、設計、開発、テスト、運用が段階的に進むことが多かったが、AI を含むシステムでは、運用後のデータや利用状況から新しい要求が生まれるためである。
そのため、要求管理は、次のような循環で行うことが望ましい。

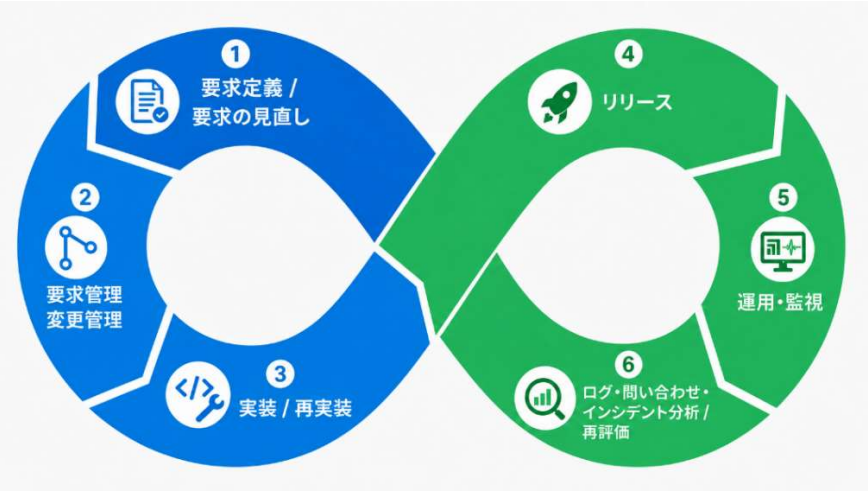


図 21 要求管理の DevOps 型循環プロセス

この考え方により、要求管理は静的な文書管理ではなく、価値を継続的に高める活動になる。AI の出力品質、利用者の反応、問い合わせ内容、モデルの性能劣化、監査結果を継続的に確認し、必要に応じて要求、データ、プロンプト、モデル、業務手順、社内規則を更新する。
DevOps 型の要求管理では、開発チームに加え、業務部門、AI 担当、データ担当、法務、監査、セキュリティ部門も関与する。運用中に発生した問題は、単なる障害対応として扱うのではなく、要求、要件、データ、ルール、運用手順のどこに原因があるのかを確認し、必要に応じて要求定義へ戻す必要がある。

11.4 トレーサビリティマトリクス

トレーサビリティを管理する代表的な方法が、トレーサビリティマトリクスである。これは、要求と設計、実装、テスト、運用、ルールなどの対応関係を表で管理するものである。
簡単な例は次のとおりである。

表 41 トレーサビリティマトリクスの例

要求 ID	要求	発生源	関連要件	関連データ	関連機能	検証方法	運用ログ
R-001	利用者本人を確認する	法令・社内規則	FR-001	利用者 ID、認証情報	ログイン機能	認証テスト	認証ログ
R-002	AI 回答を人間が確認する	リスク評価	FR-012	問い合わせ内容、回答案	回答案生成	レビュー確認	採用率、修正率
R-003	個人情報の目的外利用を防ぐ	法令	NFR-008	顧客情報	アクセス制御	権限テスト	アクセスログ

トレーサビリティマトリクスは、最初から巨大で複雑なものを作る必要はない。重要な要求、リスクの高い要求、法令・標準に基づく要求、AI に関係する要求から始めるとよい。
また、表のみで管理するのではなく、要求管理ツールやモデリングツールと連携させることが望ましい。要求 ID、データ項目 ID、テストケース ID、リスク ID、ルール ID を付与し、変更時に影響範囲を検索できるようにする。

11.5 変更管理と影響分析

5.1 節で示したとおり、AI 時代には、変更管理を前提に要求管理を行うことが重要である。要求、データ、AI モデル、プロンプト、法令、標準、業務プロセスは変化する。変更を禁止することはできない。重要なのは、変更を管理できる状態にすることである。
変更管理では、少なくとも次の流れで整備する。

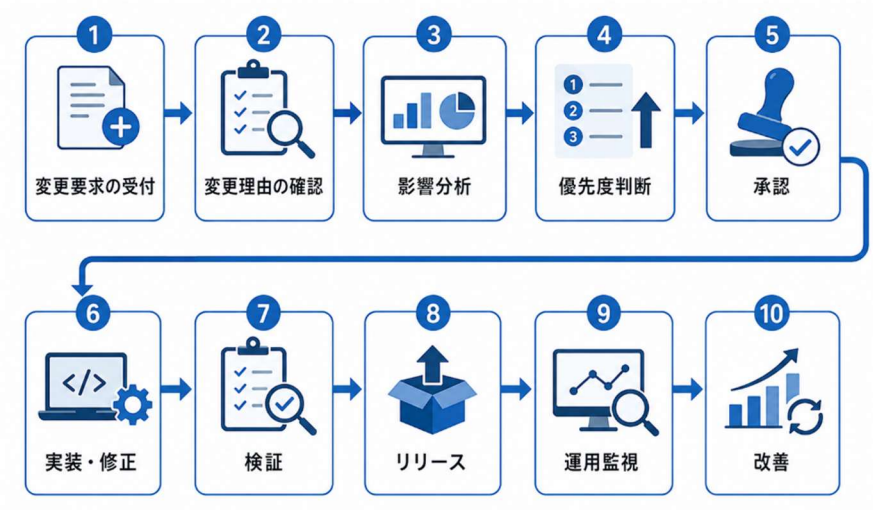


図 22 変更管理の主要プロセス

変更要求は、さまざまなところから発生する。利用者からの要望、業務変更、制度改正、標準改定、AIモデル更新、データ品質問題、セキュリティインシデント、監査指摘、障害、コスト削減要求などである。
変更要求を受け付けた場合には、まず、その変更が既存の目的・ゴールを実現するための要求・要件や実現手段の変更なのか、目的・ゴールそのものの見直しを伴う変更なのかを確認する必要がある。

既存の目的・ゴールが維持される場合には、影響分析を行った上で、要求・要件、機能、データ、AIモデル、業務プロセス等の必要な箇所を変更する。一方、目的・ゴールやその成功基準自体を変更する場合には、通常の要求変更として個別項目を修正するのみでは不十分である。ビジネス又はミッションの分析まで戻り、実現すべき価値、対象となる業務、ビジネス・業務モデル、機能の分担、必要なデータ、評価基準等を再整理した上で、要求・要件を再定義し、新たなベースラインを設定する必要がある。
変更管理で重要なのは、変更を単なる作業依頼として扱わないことである。変更の背景、目的、影響範囲、リスク、優先度、検証方法を確認する必要がある[24]。
例えば、AIモデルを更新する場合、単に新しいモデルに差し替えるのみでは不十分である。次のような確認が必要になる。

- どの要求に関係するモデルか
- どの業務で利用されているか
- どのデータを入力としているか
- 既存の評価指標を満たすか
- 出力の傾向は変わるか
- 人間の確認手順に変更は必要か
- ログや証跡に変更はあるか
- 利用者への説明は必要か
- 法令・標準・契約との整合性は保たれるか

以上を踏まえて、変更のレベルごとにどのような対応をとるべきかを次に示す。

変更のレベル	例	基本的な対応方針
目的・ゴールの変更	事業目的、提供価値、成功基準そのものの変更	ビジネス／ミッション分析まで戻り、影響のある要求・要件などの下位要素も再構成
要求・業務モデルの変更	ステークホルダーからの要求の変更、業務プロセスの変更、役割分担の変更	要求分析・モデリングまで戻って、影響のある要求とその下位要素を再整理
実現手段・要件の変更	AIモデル変更、性能条件変更、データ項目変更	要件仕様化まで戻って、影響のある要件やその下位要素を再整理

[24] BABOK [3]では要求変更を「全体戦略として整合しているか」「ビジネスまたはステークホルダーへの価値に影響するか」「価値を届けるまでの時間やリソースに影響するか」「リスク・機会・制約を変化させるか」といった観点で評価すべきとしている。

また、変更要求を管理する際には、変更が生じた理由を区別することが重要である。例えば、法令、業務、市場、利用環境等の変化に対応する「環境変化による変更」、実運用や利用者から新たな知見を得たことによる「学習による変更」、当初の要求獲得・分析・評価が十分でなかったことによる「初期要求定義の不備の訂正」などがある。これらを同一に扱うのではなく、変更理由を記録し、後者については要求定義プロセス自体の改善につなげることが望ましい。なお、要求・要件自体は妥当であり、実装又は運用がそれを満たしていない場合には、要求を変更するのではなく、不具合又は運用上の問題として是正する必要がある。
これらの点から、変更管理は、AI 時代の品質管理、リスク管理、説明責任の基盤である。

11.6 ReqIF と OSLC による要求情報の交換・連携

変更管理と影響分析を継続的に行うためには、要求情報を特定の文書や特定のツールに閉じ込めず、設計、モデル、コード、テスト、変更要求、障害情報、運用ログ、監査証跡などの関連情報と接続できる形で管理する必要がある。また、複数の企業、部門及びツールの間で、要求情報を交換し、要求と関連情報との関係を維持できるようにすることが重要である。
このとき重要になる標準が、ReqIF と OSLC である。ReqIF は、要求情報をツール間で交換するための標準形式であり、OSLC は、要求、設計、テスト、変更管理などのライフサイクル情報を、ツールをまたいでリンクし、連携させるための考え方である。どちらも要求情報の管理とツール間連携に関係するが、両者は競合するものではなく、目的に応じて使い分けるものと考えるとよい。
ReqIF（Requirements Interchange Format） [29]は、要求管理ツール間で要求情報を交換するための国際標準である。要求 ID、要求文、属性、階層、関係、状態、検証方法などを XML 形式で構造化して交換するために利用できる。特定のツールに依存せずに要求情報をやり取りできるため、発注者と受注者、複数ベンダー、異なるモデリングツール間での連携、特に「ある時点の要求情報をまとめて交換する」場合に有効である。例えば、次のような場面で活用できる。

- 発注者が要求一覧を受注者に渡す
- 受注者が詳細化した要求を発注者に返す
- 複数の開発会社が要求情報を共通形式で共有する
- ツールを移行する際に要求データを移す

要求情報を文書としてではなく、ID、属性、階層、関係を持つデータとして交換できる点に意義がある。

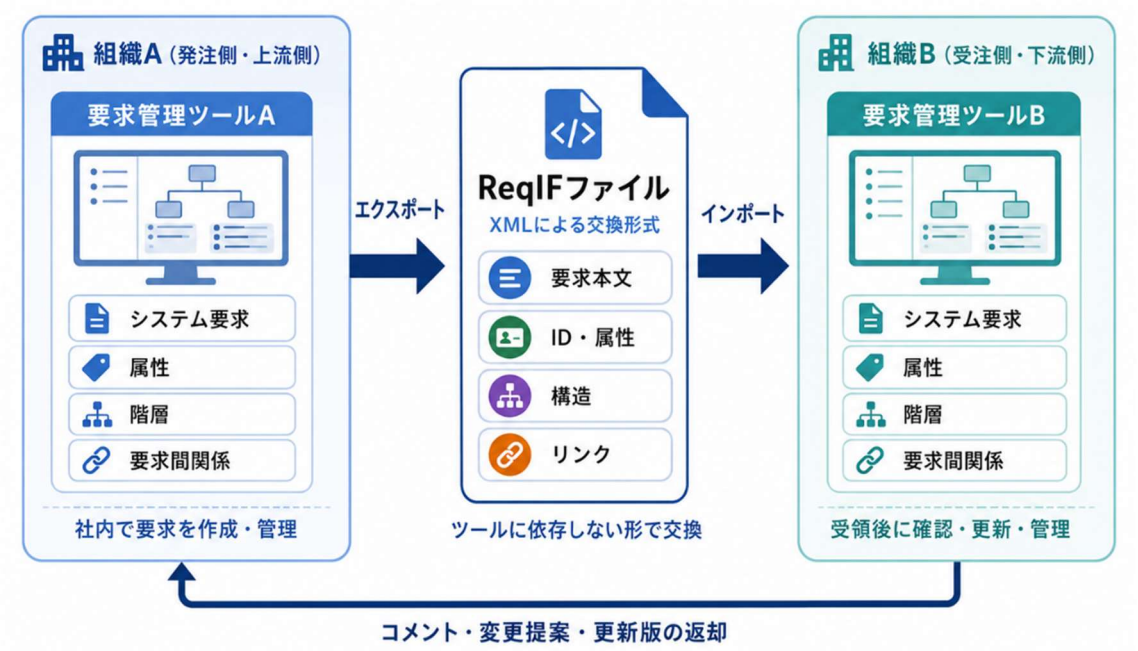


図 23 ReqIF による要求の交換イメージ
Object Management Group, Requirements Interchange Format (ReqIF), Version 1.2, Figure 1.1 [29]**を参考に作成**

一方、OSLC（Open Services for Lifecycle Collaboration） [30]は、要求管理、設計管理、テスト管理、変更管理、構成管理など、ライフサイクル全体に関わる異なるツールを連携させるための標準的な考え方である。要求をファイルとして一括交換することに加え、要求、設計モデル、テストケース、変更要求、障害票などを Web 上のリソースとしてリンクし、ツール間で参照できるようにすることを重視する。

例えば、ある要求が、SysML モデルのどの要素に対応し、どのテストケースで検証され、どの変更要求により修正され、どの障害票と関係しているかを、ツールをまたいで追跡したい場合がある。このような場合、OSLC の考え方は有効である。要求管理ツール、モデリングツール、テスト管理ツール、課題管理ツールが異なっていても、要求と関連情報をリンクできれば、トレーサビリティや影響分析を行いやすくなる。

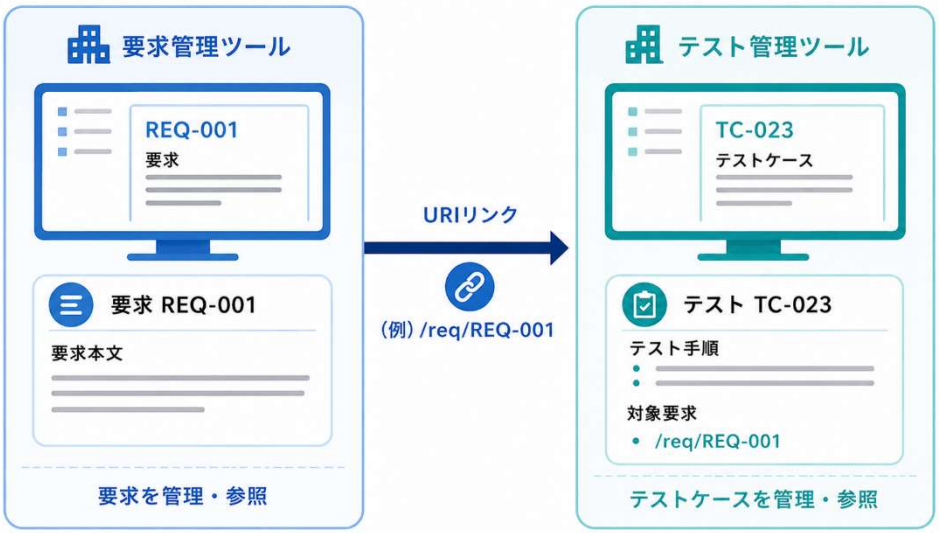


図 24 OSLC の考え方に基づく要求とテストのリンクイメージ

ReqIF と OSLC の違いを整理すると、次のようになる。

表 42 ReqIF と OSLC の違い

観点	ReqIF	OSLC
主な目的	要求情報の交換	ライフサイクル情報の連携
使い方	要求データをファイルとして受け渡す	要求、設計、テスト、変更等をリンクする
向いている場面	発注者・受注者間の要求交換、ツール移行、要求一覧の共有	複数ツール間のトレーサビリティ、影響分析、変更管理
管理対象	要求、属性、階層、関係	要求、設計、テスト、変更、構成、障害等の関連情報
特徴	要求情報をまとまった単位で交換しやすい	ツールをまたいでライフサイクル情報をつなぎやすい
活用効果	要求情報の標準的な受け渡し	継続的なトレーサビリティとライフサイクル管理

企業で ReqIF を活用する場合、すべての要求を最初から高度に管理する必要はない。まずは、重要な要求、法令・標準に基づく要求、AI に関係する要求、高リスク要求について、要求 ID、要求文、発生源、優先度、検証方法、関連要求を持たせることから始めるとよい。これにより、要求情報を要求定義書の文章から切り出し、要求管理ツールやモデリングツールで扱いやすくなる。

OSLC を活用する場合には、要求情報に加え、要求に関連する成果物をリンクすることが重要である。例えば、要求 ID と業務フロー、データモデル、SysML モデル、テストケース、変更要求、障害票、監査項目を関連づける。これにより、ある要求を変更した場合に、どの設計、どのテスト、どの運用手順、どの監査項目に影響するかを確認しやすくなる。

AI 時代には、ReqIF と OSLC の重要性がさらに高まる。AI を活用して要求候補を抽出したり、要求の過不足を確認したり、影響分析を行ったりするためには、要求情報が構造化され、他のライフサイクル情報と接続されている必要がある。要求が単なる Word 文書や Excel 表として孤立していると、AI やツールが要求の意味、関係、変更影響を十分に扱えない。

例えば、AI が「この法令変更はどのシステム機能に影響するか」を分析するためには、法令に基づく要求、関連する業務プロセス、データ項目、設計モデル、テストケース、運用ログがつながっている必要がある。ReqIF により要求情報を標準的に交換し、OSLC により要求と関連成果物をリンクできれば、AI による影響分析、変更候補の提示、テスト対象の抽出、監査証跡の確認がしやすくなる。

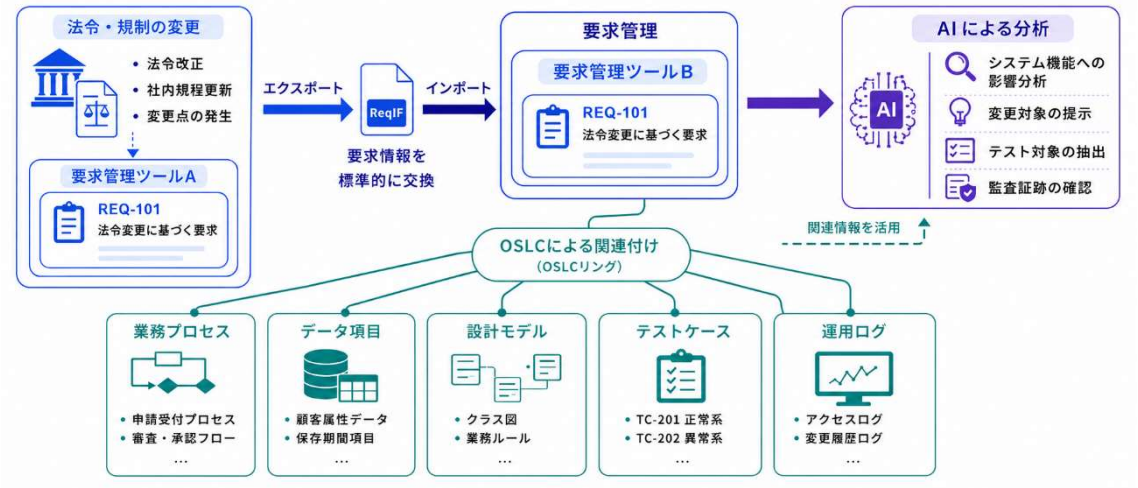


図 25 ReqIF と OSLC と AI による法令変更の影響調査イメージ

ReqIF と OSLC を導入する際には、最初から大規模な統合基盤を作る必要はない。まずは、重要要求に ID を付け、要求管理台帳を整備し、要求と要件、要件とテスト、要件と設計、要件と変更要求を対応づけるところから始める。そのうえで、発注者・受注者間の要求交換には ReqIF を活用し、複数ツール間のリンク管理や継続的なトレーサビリティには OSLC の考え方を取り入れるとよい。

ReqIF と OSLC の活用により、要求定義書を単なる文書として扱うのではなく、要求情報の集合、さらにライフサイクル全体と接続された知識基盤として扱うことができる。これにより、要求管理ツール、モデリングツール、テスト管理ツール、開発管理ツール、運用管理ツールとの接続が容易になる。特に、複数の企業や部門が関わる開発では、要求情報の交換形式とリンク方法を標準化しておくことが、変更管理、責任分担、説明責任を明確にする上で重要である。

第 4 部　ルール・制度・コンプライアンスと要求管理

AI 時代の要求管理では、業務部門や利用者からの要望に加え、法令、規制、標準、ガイドライン、契約、社内規則、マニュアルなどのルールを重要な要求源として扱う必要がある。これらのルールは、システムや AI が守るべき制約であると同時に、企業活動の品質、安全性、説明責任、信頼性を支える基盤でもある。

企業にとって、ルールには大きく次の二つの種類がある。

1. 外部から与えられるルール

   法令、規制、行政ガイドライン、業界標準、認証基準、契約条件などがこれにあたる。これらは、企業が事業を行ううえで守らなければならない外部要件であり、コンプライアンス、監査、説明責任に直結する。

2. 企業自身が定める内部ルール

   社内規則、業務ガイドライン、マニュアル、手順書、チェックリスト、FAQ、教育資料などがこれにあたる。これらは、外部要件を社内業務に落とし込み、従業員が日々の業務で適切に判断し、行動するための実務上の基盤である。

AI 時代には、後者の重要性が特に高まる。なぜなら、AI やシステムが業務を支援するためには、社内規則やマニュアルが、単なる文書ではなく、AI が参照しやすく、人間も理解しやすく、変更管理しやすい形で整備されている必要があるためである。社内規則やマニュアルを要求工学の観点で整理すれば、問い合わせ対応、業務支援、教育、監査、リスク管理、AI 活用が大きく改善する。

第 4 部では、まず外部要件としての法令・規制・標準を整理し、そのうえで、企業が実務で活用しやすい社内規則、ガイドライン、マニュアルを中心に、AI 時代の要求管理との関係を説明する。

第 12 章　外部要件としての法令・規制・標準

12.1 外部要件とは何か

外部要件とは、企業の外部にある制度、規則、基準、契約などから発生する要件である[25]。企業は、事業活動を行ううえで、顧客や市場の要求に加え、法令、規制、業界標準、契約、社会的責任に基づく要件を満たすことが求められる。

代表的な外部要件には、次のようなものがある。

[25] ISO/IEC/IEEE 29148:2018 [1]においても、法令、規制、標準、契約、環境条件等を要求源又は制約として識別し、要求仕様へ反映することが想定されている。

表 43 代表的な外部要件

	例	要件定義で確認すべきこと
法令	個人情報保護、労働、安全、金融、消費者保護等	義務、禁止、手続、保存、説明責任
規制	監督官庁の規則、許認可条件	報告、記録、監査、審査基準
行政ガイドライン	AI、データ、セキュリティ等の指針	推奨事項、リスク対応、説明事項
標準	ISO、JIS、業界標準、技術標準	品質、互換性、データ形式、評価方法
認証基準	セキュリティ、品質、AI、製品安全等	適合性評価、証跡、監査項目
契約	顧客契約、委託契約、SLA	責任分担、納期、品質、データ利用条件
社会的要請	ESG、人権、倫理、安全性	説明責任、透明性、リスク対応

従来、外部要件は、法務部門やコンプライアンス部門が確認し、システム開発では「法令を遵守すること」「契約条件に従うこと」といった抽象的な要件として記載されることが多かった。しかし、このような書き方では、開発者や AI 担当者が何を実装すればよいのか分からない。
AI 時代の要求管理では、具体的な外部要件を、業務、データ、システム、AI、運用、監査に落とし込むことが求められる。

12.2 「法令を遵守すること」では要件にならない

要件定義書に「関連法令を遵守すること」と書かれていることは多い。しかし、この表現のみでは、実装も評価もできない。どの法令の、どの条項や基準に基づき、どの業務で、どのデータを、どのように扱い、どの証跡を残し、どのように確認するのかが分からないためである。
例えば、個人情報の取扱いに関して、単に「個人情報保護法を遵守すること」と書いても、要件としては不十分である。要件として扱うためには、次のように具体化する必要がある。

表 44 抽象的な表現を要件として具体化する観点の例

抽象的な表現	要件として具体化する観点
個人情報を適切に扱う	どのデータが個人情報か
利用目的を明確にする	どの画面・文書で利用目的を表示するか
目的外利用を防ぐ	アクセス制御、利用ログ活用方法、AI 入力制限はどうするか
必要な記録を残す	利用履歴、同意、削除依頼、開示請求など具体的にどれか
安全管理措置を講じる	権限管理、暗号化、監査ログ、教育などの何を実施するか

同じように、「AI 利用に関するガイドラインに従うこと」と書くのみでは、AI システムの要件としては不十分である。利用目的、入力データ、出力、禁止用途、人間の関与、評価方法、ログ、監視、インシデント対応に分解する必要がある。
外部要件を要求定義に取り込む際には、次の問いを確認する。

- どの法令、規制、標準、契約に基づく要件か
- 企業に求められる義務は何か
- 禁止されている行為は何か
- 実施できる条件は何か
- 必要な手続は何か
- 保存すべきデータや証跡は何か
- 誰が責任を持つのか
- どの業務プロセスに反映するのか
- どのシステム機能に反映するのか
- どのように評価・監査するのか
- 変更があった場合、どこに影響するのか

このように、外部要件は、法務部門の確認事項にとどめず、業務要件、データ要件、AI 要件、ルール要件、評価要件、運用要件へ展開する必要がある。

12.3 外部要件を社内要件へ変換する

企業実務で重要なのは、外部要件をそのまま現場に渡すことではなく、社内で実行できる要件に変換することである。
法令や標準は、一般的・抽象的に書かれていることが多い。企業ごとの業務、組織、システム、データ、リスク、顧客対応にそのまま当てはめるには、社内の文脈に合わせた解釈と具体化が必要である。
例えば、外部要件が「必要な安全管理措置を講じること」であった場合、社内要件としては、次のように変換される。

外部要件

必要な安全管理措置を講じること

社内要件

- 顧客データへのアクセス権限は職務に応じて付与すること
- 退職者および異動者の権限は月次で確認すること
- 顧客データを AI に入力する場合は、承認済みの AI 環境に限定すること
- 顧客データの利用ログを保存すること
- 年 1 回、データ取扱いに関する教育を実施すること

この変換により、抽象的な外部要件が、現場で実行可能な業務ルール、システム要件、データ要件、監査要件、教育要件になる。
外部要件から社内要件への変換では、次の対応関係を作ることが重要である。

表 45 外部要件からの社内要件への展開

外部要件種別	社内要件への展開
義務	社内規則、業務手順、システム機能
禁止	禁止行為リスト、アクセス制御、チェック機能
条件	承認フロー、判定基準、例外処理
記録	ログ、台帳、証跡、保存期間
説明責任	説明文、通知、FAQ、問い合わせ対応
評価	チェックリスト、監査項目、テスト
変更	改定管理、影響分析、教育更新

AI 時代には、この変換を AI で支援することも可能である。法令や標準の文書を読み込み、要件候補を抽出し、社内規則やチェックリストの案を作ることができる。ただし、最終的な解釈、リスク判断、社内適用の妥当性は、人間が確認する必要がある。

第 13 章 規則・ガイドライン・マニュアルを要求定義に活かす

13.1 社内規則は「使われる要件」でなければならない

企業において、社内規則、ガイドライン、マニュアルは、日々の業務を支える重要な文書である。ところが、現実には、これらが十分に活用されていないことが多い。
例えば、社内規則は総務部門や法務部門が管理しているが、現場の業務システムや AI 活用とは十分につながっていない。よくある事例は次のとおりである。

- マニュアルは作成されているが、内容が古く、どの業務に適用されるのか分かりにくい
- ガイドラインは存在しているが、具体的に何をすればよいのかが現場に伝わっていない
- FAQ はあるが、正式な規則や手順との対応関係が不明である

このような状態では、社内ルールは「存在している文書」にとどまり、「使われる要件」になっていない。
AI 時代には、社内規則やマニュアルを、AI やシステムが参照しやすい形で整備することが重要になる。社内規則が構造化されていれば、社員からの問い合わせに対して、AI が正確な規則名や該当箇所を示しながら回答できる。マニュアルが業務プロセスやデータ項目と結びついていれば、業務支援 AI や RPA、ワークフローシステムと連携しやすくなる。
社内規則は、単に遵守することのみを目的とした文書ではない。業務を標準化し、AI 活用を進め、教

育を効率化し、監査を容易にし、変更に強い組織を作るための要件群である。

13.2 社内ルールを種類ごとに整理する

社内のルール文書には、さまざまな種類がある。これらを区別せずに扱うと、どの文書が何を決めているのか分かりにくくなる。まず、文書の役割を整理することが重要である。

表 46 社内文書の種類ごとの主な役割と要求定義での位置づけ

	主な役割	要求定義での位置づけ
社内規則	守るべき基本ルールを定める	義務、禁止、権限、責任の根拠
ガイドライン	判断の考え方や推奨事項を示す	判断基準、運用方針、リスク対応
マニュアル	具体的な作業手順を示す	業務プロセス、操作手順、例外処理
チェックリスト	確認事項を示す	評価要件、監査要件、受入条件
FAQ	よくある質問への回答	利用者支援、教育、問い合わせ対応
教育資料	理解促進・定着を支援する	人材育成、運用定着
テンプレート	文書や申請の標準形を示す	入力項目、必要データ、証跡

社内規則は上位ルールであり、マニュアルは現場の行動に近い。ガイドラインは、その間に位置し、判断の考え方や実務上の解釈を示す。チェックリストは、実施状況を確認するための要件である。
この階層を整理すると、社内ルールを要求定義に活用しやすくなる。

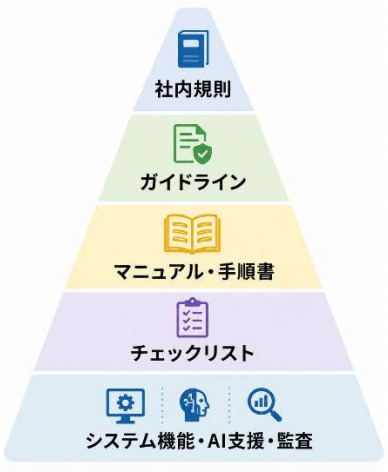


図 26 社内ルールの階層

AI を活用する場合も、この階層が重要である。AI が社員からの問い合わせに回答する場合、上位の規則、実務上のガイドライン、具体的なマニュアル、FAQ のどれに基づいて回答しているのかを明確にする必要がある。

13.2.1 社内規則を要件として読む

社内規則は、組織内の要件を文書化したものである。したがって、社内規則を要件定義に活用するには、規則を「要件」として読み解く必要がある。
社内規則には、次のような要件が含まれている。

表 47 社内規則に含まれる要件の例

規則中の表現	要件種別
しなければならない	義務要件
してはならない	禁止要件
できる	許可条件
承認を得る	承認プロセス要件
記録する	証跡・ログ要件
保存する	データ保存要件
報告する	通知・報告要件
確認する	チェック要件
定期的に見直す	評価・改善要件
責任者を置く	役割・責任要件

例えば、社内規則に「重要情報を外部サービスに入力してはならない」と書かれている場合、これは単なる禁止文ではない。要件定義の観点では、次のように展開できる。

規則

　重要情報を外部サービスに入力してはならない。

要件

- 重要情報の定義を明確にすること
- 外部サービスの範囲を明確に定義すること
- AI サービス利用時に入力禁止情報を明示すること
- 入力画面または利用手順で注意喚起すること
- 利用ログを保存すること
- 違反時の報告・対応手順を定めること
- 社員教育を実施すること

このように、社内規則を要件として読むと、業務手順、システム機能、AI 利用制限、ログ、教育、監査につながる。

13.2.2 ガイドラインは判断基準を明確にする

ガイドラインは、社内規則ほど厳格な義務ではないが、現場が判断するための重要な基準である。特にAI やデータ活用のように、技術変化が早く、すべてを詳細な規則で定めにくい領域では、ガイドラインの役割が大きい。

ガイドラインは、次のような場面で有効である。

- AI を使ってよい業務と使ってはいけない業務を示す
- 個人情報や機密情報の扱い方を示す
- 生成 AI の出力を確認する際の観点を示す
- リスクが高い場合のエスカレーション基準を示す
- 人間による確認が必要な場面を示す
- 顧客説明や社内承認の考え方を示す
- 外部サービス利用時の注意点を示す

ガイドラインを要求定義に活用する場合、抽象的な方針を具体的な判断基準に落とし込むことが重要である。

例えば、「生成 AI の出力は適切に確認すること」というガイドラインは、そのままでは不十分である。要件として使うには、次のように具体化する。

ガイドライン

生成 AI の出力は適切に確認すること。

要件

- 顧客に送付する文書は、担当者が内容を確認してから送信すること
- 法務判断を含む回答は、法務部門または承認者が確認すること
- 数値、日付、固有名詞は原資料と照合すること
- AI が参照した資料を記録すること
- 生成 AI の出力を修正した場合は、修正履歴を保存すること

ガイドラインは、現場の裁量を残しつつ、判断の品質をそろえるためのものである。AI 時代には、ガイドラインを AI が参照できる形にし、判断支援、教育、問い合わせ対応に使えるようにすることが重要である。

13.2.3 マニュアルは業務プロセス要件である

マニュアルは、業務の具体的な進め方を示す文書である。AI 時代の要件定義では、マニュアルを単なる手順書ではなく、業務プロセス要件として扱うとよい。

マニュアルには、次のような要件が情報として含まれている。

表 48 マニュアルに含まれる要件の例

マニュアルに含まれる情報	要件種別

作業手順	業務プロセス要件
担当者	役割・責任
入力情報	データ要件
出力物	成果物要件
判断基準	ルール要件
例外処理	例外要件
承認	ワークフロー要件
記録	証跡・ログ要件
注意事項	リスク・制約
問い合わせ先	運用支援要件

マニュアルが構造化されていれば、業務支援 AI に活用しやすくなる。例えば、社員が「この申請は誰の承認が必要か」と質問したとき、AI がマニュアルの該当箇所を参照し、承認者、必要書類、例外条件を回答できる。手順がワークフローシステムと連携していれば、AI が次に行うべき作業を案内することもできる。

一方、マニュアルが古く、部門ごとにばらばらで、規則や実システムと対応していない場合、AI に読み込ませても正しい支援はできない。AI 活用の前提として、マニュアルの整理、構造化、更新管理が必要である。

マニュアルを要件定義に活用する際には、次の点を確認する。

- 現在の業務実態と一致しているか
- 上位の社内規則やガイドラインと整合しているか
- 入力データと出力物が明確か
- 例外処理が記載されているか
- 判断基準が明確か
- システム機能と対応しているか
- AI が参照してよい情報か
- 変更時に更新される仕組みがあるか

第 14 章　社内ルールの構造化と標準形式

14.1 文書を構造化する意義

社内規則やマニュアルを AI 時代に活用するためには、文書の構造をそろえることが有効である。すべての文書を完全にデータベース化する必要はないが、目的、適用範囲、用語定義、義務、禁止、承認、例外、責任者、記録、関連文書、改定履歴を明確にしておくと、AI が検索し、根拠を示し、手順を案内しやすくなる。

また、文書を構造化することにより、人間にとっても読みやすく、更新しやすくなる。ルールの構造が明確であれば、どこが義務で、どこが推奨で、どこが例外で、どこが手順かを区別しやすくなる。
AI が社内文書を活用するためには、少なくとも次の条件が必要である。

- 正本が明確であること
- 文書 ID、版、所管部門が明確であること
- 目的と適用範囲が明確であること
- 義務、禁止、承認、例外が区別されていること
- 関連する社内規則、ガイドライン、マニュアル、FAQ がつながっていること
- 改定履歴が管理されていること
- AI が参照してよい文書と参照してはいけない文書が区別されていること

このように文書を構造化することで、社内ルールは単なる保管文書ではなく、AI、業務システム、監査、教育に活用できる要件情報になる。
なお、文書の構造化には様々な方法があり、今後も技術の進展に伴って新たなデータ形式や手法が登場する可能性がある。本章では、現時点で利用可能な手段の例として Markdown や XML を紹介するとともに、関連する標準や情報交換形式として Akoma Ntoso や ReqIF を取り上げる。重要なのは、特定の形式を採用することではなく、文書の構造や意味、関連性を明確にし、人間と AI・システムの双方が適切に活用できる状態にすることである。実際の適用にあたっては、利用目的や既存システムとの連携、管理・運用の容易さなどを踏まえて適切な形式を選択する必要がある。

14.2 Markdown による軽量な構造化

社内文書を比較的手軽に構造化する方法の一つとして、Markdown がある。Markdown は、人間が読みやすく、Git 等で版管理しやすいという特徴がある。見出し、箇条書き、表、メタデータを用いることで自然言語文書でありながら一定の構造を持たせることができ、AI による文書の解析や情報の抽出にも活用できる。
Markdown で執筆した生成 AI ガイドラインのサンプルは次のとおりである。

```
---
document_id: AI-GUIDE-001
title:  生成 AI 利用ガイドライン
owner: DX 推進部
version: 1.2
effective_date: 2026-07-01
related_rules:
  - 情報管理規程
  - 個人情報取扱規程
```

---

1. 目的
本ガイドラインは、社員が生成 AI を安全かつ有効に利用するための基本方針を定める。

2. 適用範囲
本ガイドラインは、業務で生成 AI を利用するすべての社員に適用する。

3. 禁止事項
- 顧客の個人情報を未承認の生成 AI サービスに入力してはならない。
- 機密情報を外部 AI サービスに入力してはならない。

4. 確認事項
- 顧客向け文書に AI 出力を利用する場合は、担当者が内容を確認する。
- 法務判断を含む場合は、法務部門に確認する。

5. 改定履歴
版	日付	変更内容	承認者
1.0	2026-04-01	初版作成	DX 推進部長

Markdown の利点は、導入しやすいことである。既存の社内文書を一気に高度な XML へ移行することは難しいが、見出し、メタデータ、表、関連文書、改定履歴を整備することでも、AI による検索や要件抽出の精度は高まる。

14.3 XML による構造化

文書の構造や属性をより厳密に定義する方法の一つとして、XML がある。XML では、義務、禁止、条件、例外、責任者、関連文書などをタグで明示できる。また、記述すべき項目やデータ形式などのルールをあらかじめ定め、それに従って記述されているかを機械的に確認できるため、社内規則やマニュアルを AI やシステムが処理しやすい形で管理する場合に有効である。
XML で執筆した生成 AI 利用規則のサンプルは次のとおりである。

```
<internalRule id="AI-RULE-001" version="1.2">
  <title>生成 AI 利用規則</title>
  <purpose>生成 AI を安全かつ有効に利用するための基本事項を定める。</purpose>
```

```
    <scope>業務で生成 AI を利用するすべての社員</scope>

    <prohibition id="P-001">
      <text>顧客の個人情報を未承認の生成 AI サービスに入力してはならない。</text>
      <relatedRequirement>REQ-AI-012</relatedRequirement>
      <evidence>AI 利用ログ</evidence>
    </prohibition>

    <obligation id="O-001">
      <text>顧客向け文書に AI 出力を利用する場合は、担当者が内容を確認しなければならない。</text>
      <relatedRequirement>REQ-AI-021</relatedRequirement>
      <evidence>レビュー記録</evidence>
    </obligation>
  </internalRule>
```

XML は、人間にとってやや扱いにくいが、文書構造を明確にし、システムや AI との連携に向いている。特に、義務、禁止、条件、例外、証跡、関連要求を明示することで、ルールを要件情報として扱いやすくなる。

14.4 Akoma Ntoso の考え方

法令や規則文書をより標準的に構造化する場合、Akoma Ntoso [31]のような標準的な法令文書モデルを参考にできる。Akoma Ntoso は、法律、議会文書、司法文書などを、条、項、号、定義、参照、メタデータなどの構造で記述するための標準的な考え方を提供している。
企業の社内規則をそのまま Akoma Ntoso に完全準拠させる必要はない。しかし、次の考え方は社内ルールの構造化にも有効である。

- 文書 ID、版、施行日、所管部門などのメタデータを持つ
- 条、項、号などの階層構造を明確にする
- 定義、義務、禁止、例外、参照を区別する
- 他の文書や条項への参照を明示する
- 改定履歴を管理する
- 機械処理可能な構造を持たせる

社内規則やガイドラインを AI で活用する場合、こうした構造を持たせることで、AI が根拠条文、関連規則、例外条件を示しやすくなる。

14.5 ReqIF との接続

社内ルールから抽出した要件をはじめ、組織間で交換する可能性がある要求全般を要求管理ツールで扱う場合には、ReqIF との接続を考えるとよい。例えば、社内規則を Markdown や XML 等で管理している場合、そこから抽出した要件、禁止事項、承認条件、ログ要件、監査要件を ReqIF 形式で要求管理ツールへ連携することができる。
例えば、生成 AI 利用規則から次のような要求を抽出する。

表 49 生成 AI 利用規則から抽出した要求の例

要求 ID	要求文	発生源	検証方法
REQ-AI-012	生成 AI の利用に伴う顧客の個人情報の漏えいを防止したい。	生成 AI 利用規則 P-001	入力制限確認、ログ監査
REQ-AI-013	AI を利用して作成した顧客向け文書について、誤りや不適切な内容が顧客に提供されることを防止したい。	生成 AI 利用規則 P-002	レビュー記録確認

このような要求を ReqIF で管理すれば、要求管理ツール、モデリングツール、テスト管理ツールとの連携が容易になる。
ReqIF を使う意義は、要求を文書から切り出し、属性や関係を持った管理対象として扱える点にある。要求 ID、要求文、発生源、優先度、状態、関連要件、検証方法、変更履歴を標準的な形式で交換できれば、組織やツールをまたいだ要求管理が容易になる。特に、複数のベンダーや部門が関わるプロジェクトでは、ReqIF のような交換形式を考慮することが有効である。

14.6 社内ルールを AI で活用する

社内規則、ガイドライン、マニュアルを構造化すると、AI 活用の幅が広がる。特に、企業内での問い合わせ対応、業務支援、教育、監査、リスク管理に効果がある。
代表的な活用例は次のとおりである。

表 50 社内ルールでの AI 活用例

活用場面	AI による支援内容
社内問い合わせ	規則名、該当条項、手順を回答する
業務支援	次に行う作業、必要書類、承認者を案内する
文書作成	規則に沿った申請書、報告書、説明文を作成する
教育	新入社員や異動者向けにルールを説明する
監査	規則違反の可能性、証跡不足を検出する

リスク管理	高リスク行為や例外処理を抽出する
変更管理	規則改定時の影響範囲を提示する
ナレッジ管理	FAQ、マニュアル、過去事例を整理する

ただし、AI に社内ルールを効果的に活用させるためには、次のような状態が求められる。

1. 正本が明確であること
   どの規則やマニュアルが最新で、正式なものか分からなければ、AI は古い情報や誤った情報に基づいて回答する可能性がある。
2. 用語が整理されていること
   同じ言葉が部門ごとに違う意味で使われていると、AI の回答も不安定になる。
3. 文書間の関係が明確であること
   社内規則、ガイドライン、マニュアル、FAQ の対応関係が分からなければ、AI は根拠を示せない。
4. 例外処理が記載されていること
   現場業務では例外が多い。例外が文書化されていなければ、AI は通常手順のみを回答し、実務と合わない支援を行う可能性がある。
5. 利用制限があること
   AI に回答させてよい範囲、人間に確認すべき範囲、法務や専門部署へエスカレーションすべき範囲を明確にしていなければ、AI が権限や専門性を超えて判断し、不適切な回答や対応を行う可能性がある。

第 5 部　AI 時代の要求管理プロセスの組織への定着

AI 時代の要求管理は、個別プロジェクトの上流工程にとどまるものではない。事業、業務、データ、AI、ソフトウェア、法令・標準、社内規則、運用、監査をつなぎ、変化に対応しながら価値を継続的に生み出すための組織能力である[26]。
そのためには、要求管理を属人的な作業にせず、標準的なプロセス、成果物、役割分担、成熟度の考え方を整備する必要がある。すべてのプロジェクトで大規模な文書を作成する必要はないが、目的、要求、要件、データ、AI、ルール、評価、変更管理を最低限確認できる状態にしておくことが重要である。
AI 時代の要求管理プロセスについては第 2 部の第 7 章で示しているため、第 5 部では、AI 時代の要求管理を実務で進めるための組織への導入方法（第 15 章）と主な成果物（第 16 章）を示す。

第 15 章　組織への導入と成熟度

15.1 要求管理を組織能力にする

AI 時代には、要求管理を一部の担当者の経験や勘に任せることはできない。AI、データ、セキュリティ、法令、標準、業務、運用が複雑に関係するため、要求管理は組織として持つべき能力である。
要求管理を組織能力にするためには、次の要素が求められる。

表 51　要求管理を組織能力にするための要素

	内容
標準プロセス	要求獲得、分析、仕様化、評価、管理の流れを定める
標準テンプレート	要求定義書、要件定義書、要求管理台帳、AI 要求管理等の型を用意する
役割分担	事業、IT、AI、データ、法務、監査の責任を明確にする
レビュー体制	要求の妥当性、実現可能性、リスクを確認する
ツール活用	要求管理、モデリング、変更管理を支援する
教育	要求工学、AI、データ、ルール要求の基礎を学ぶ
改善サイクル	運用後のフィードバックを変更要求に反映する

また、要求管理の成果物は、単なるプロジェクト文書ではなく、組織の知識基盤である。これらには、組織が何を重視しているのか、どの業務をどのように行っているのか、どのデータを使っているのか、どのルールを守る必要があるのか、どのリスクを避けるべきか、どのような品質を求めているのかが集約される。すなわ

[26] IIBA においても、要求定義の一環ともいえるビジネス分析を「組織の文脈で変革を可能にする実践であり、ニーズを定義し、利害関係者に価値をもたらす解決策を提案すること」と定義している。 [34]

ち、要求管理は、業務知識、顧客理解、データ理解、技術知識、法令・標準・社内規則、リスク管理、運用経験を結びつける組織の知識資産を作るための活動でもある。
この知識資産が整備されていない組織では、システム開発や AI 導入のたびに、担当者が一から関係者に確認し、過去の資料を探し、暗黙知を掘り起こす必要がある。その結果、同じ議論が繰り返され、要件の抜け漏れや解釈の違いが発生し、担当者が異動すると知識が失われる。逆に、要求が組織的に管理されていれば、過去の要求、判断理由、業務ルール、データ定義、評価基準、変更履歴を再利用できる。これにより、新しいプロジェクトの立ち上げ、AI 活用、業務改革、M&A 後の統合、法令改正対応、システム更新を迅速に進めることができる。
さらに、要求管理は価値の源泉でもある。企業の競争力は、単にソフトウェアや AI モデルそのものにあるのではなく、何を解くべき課題として捉え、どの業務をどう設計し、どのデータを使い、どの品質や体験を実現するかという要求・要件の設計力に大きく左右される。同じ AI や同じクラウドサービスを利用しても、目的、要求、データ、業務プロセス、評価基準が優れていなければ、十分な価値は生まれない。逆に、要求管理が優れていれば、既存の OSS、商用サービス、SaaS、AI サービスを組み合わせても、組織固有の価値を生み出すことができる。
AI 時代には、要求定義そのものを AI に学習・参照させることも重要になる。過去の要求定義、要求管理台帳、業務フロー、データ定義、社内規則、マニュアル、障害記録、監査指摘、改善履歴が整理されていれば、AI は新しい要求定義のたたき台作成、抜け漏れ確認、類似事例検索、影響分析、リスク抽出を支援できる。
したがって、要求管理を組織能力にするということは、単にテンプレートを作ることではない。組織が持つ業務知識、データ知識、ルール知識、技術知識、運用知識を、再利用可能で、更新可能で、人間と AI がともに活用できる形にすることである。要求管理によって組織の知識基盤を整備することにより、組織は変化に対応する力を高め、新しい価値を継続的に生み出すことができる。

15.2 役割分担

AI 時代の要求管理では、複数部門の協力が不可欠である。良い要求を定義するために確認すべきことは多岐にわたり、IT 部門やベンダーのみで良い要求を定義することは非常に困難なためである。
良い要求定義のために、主な部門・担当ごとに次のような責任を負うことが望ましい。

表 52 AI 時代の要求管理における部門・担当ごとの責任

部門・担当	主な責任
経営層・事業責任者	目的、価値、優先度、投資判断を示す
業務部門	現場要求、業務ルール、例外処理を示す
IT 部門	システム要件、実装可能性、運用性を確認する
AI 担当	AI 利用目的、評価、監視、更新条件を整理する
データ担当	データ定義、品質、出典、利用条件を確認する

法務・コンプライアンス	法令、標準、契約、社内規則との整合を確認する
セキュリティ担当	リスク、アクセス制御、ログ、インシデント対応を確認する
監査担当	証跡、評価方法、説明責任を確認する
調達・契約担当	委託先、SLA、責任分担、契約条件を整理する
利用者・現場代表	使いやすさ、現実性、業務適合性を確認する

重要なのは、各部門が自分の専門領域のみを見るのではなく、要求、データ、AI、ルール、運用がどのようにつながるかを共同で確認することである。

15.3 導入ステップ

要求管理の高度化は、一度にすべてを実現する必要はない。要求管理を組織に定着させる際には、初めからすべての成果物やトレーサビリティを網羅的に整備しようとせず、組織の成熟度やプロジェクトの規模・特性に応じて、小さく始めることが重要である。体制や経験が十分でない段階で管理対象を過度に広げると、文書作成や管理作業自体が目的化し、形骸化するおそれがある。まずは、主要な要求とその根拠、関連する要件や検証方法、変更履歴などを確認できる状態とし、試行と改善を繰り返しながら、段階的に適用範囲と管理水準を高めていくことが現実的である。
具体的には、次に示すようなステップが考えられる。

ステップ 1　現状把握

まず、要求管理がどのように行われているかを確認する。要求定義書の有無、要求管理台帳の有無、変更管理の方法、レビュー体制、モデリングの利用状況、AI やデータ要求の扱いを確認する。

ステップ 2　標準テンプレートの整備

次に、基本的なテンプレートを整備する。最初から多くの成果物を作成するのではなく、主要な要求とその根拠などを一覧で管理できる簡易な台帳から始めることも有効である。必要に応じて、AI、データ、ルールなどに関する管理項目や成果物を追加する。また、社内ルールの構造化についても、第 14 章で示した Markdown などを用いた軽量な方法から始めることが考えられる。

ステップ 3　小規模なプロジェクトから試行する

すべてのプロジェクトに一斉展開するのではなく、まずは小規模でリスクが低く、試行錯誤しやすいプロジェクトから要求管理の導入を進める。その際、実際の業務への適用可能性を確認できる対象を選び、要求管理の効果や現場の負担を検証することが重要である。試行結果を踏まえてプロセスや成果物を改善しながら、重要なプロジェクトを含めて段階的に適用範囲を拡大していく。

ステップ 4　レビューと教育を行う

試行結果をもとに、テンプレートやプロセスを改善する。同時に、事業部門、IT 部門、AI 担当、データ担当、法務・監査担当向けの教育を行う。また、要求定義の経験が十分でない組織では、対話型 AI を活用してヒアリング時の質問や確認事項を提示したり、要求候補の抽出や抜け漏れの確認を支援したりすることも有効である。こうした支援を通じて実務経験を蓄積し、組織としての要求管理能力

を段階的に高めていく。

ステップ 5　ツール活用を進める

試行や教育の段階での AI 活用に加え、必要に応じて要求管理ツール、モデリングツール、ドキュメント管理、AI 支援ツールを導入し、要求、要件、モデル、データ、テスト、ログの対応関係を管理しやすくする。

ステップ 6　組織標準として定着させる

プロジェクトごとの取組で終わらせず、組織標準、調達要件、開発標準、AI 利用ルール、監査項目に反映する。これにより、要求管理を継続的な組織能力にする。

15.4 成熟度モデル

要求管理の成熟度は、次の 5 段階で整理できる。

表 53　要求管理の成熟度

	状態	目指す姿
Lv.1 属人型	担当者の経験に依存し、文書や粒度がばらばら	最低限の文書化を始める
Lv.2 文書化型	要求定義書はあるが、変更管理やトレーサビリティが弱い	要求一覧と変更履歴を整備する
Lv.3 管理型	要求管理台帳、レビュー、変更管理がある	AI・データ・ルール要求を明示する
Lv.4 モデル活用型	業務、データ、システム、評価をモデルで接続する	影響分析や監査に活用する
Lv.5 AI 協働型	AI を使って要求抽出、分析、影響分析、改善を継続的に行う	ライブな要求管理を実現する

この成熟度は、組織全体で一律に上げる必要はない。リスクの高い領域、変更頻度の高い領域、AI 活用の効果が大きい領域から優先的に成熟度を高めることが望ましい。

なお、AI の活用は成熟度の高い組織に限られるものではない。要求管理の経験が少ない組織でも、対話型 AI によるヒアリング支援や要求候補の整理などから活用を始めることができる。Lv.5 の「AI 協働型」は、こうした個別の AI 活用にとどまらず、要求管理プロセス全体に AI を継続的に組み込み、人間と AI が協働して管理・改善を行う段階を意味する。

15.5 調達・委託時の注意点

多くの企業では、システム開発や AI 開発を外部委託する。この場合、要求定義を曖昧なまま委託すると、後工程で大きな手戻りや責任分担の問題が生じる。

調達・委託時には、少なくとも次を明確にする。

- 目的と成果
- 対象範囲
- 役割分担
- 要求定義の責任
- データ提供条件
- AI 利用条件
- 成果物
- 受入条件
- 変更管理手順
- セキュリティ要件
- 法令・標準・社内規則への対応
- 運用後の保守・監視
- 知的財産・成果物の扱い
- ログ・証跡の扱い

AI 開発では、モデル、プロンプト、学習データ、評価データ、ログ、改善履歴が重要になる。これらが誰の管理対象であり、どの範囲で利用可能かを契約や要求定義で明確にしておく必要がある。

第 16 章　成果物一覧

AI 時代の要求管理では、プロジェクトの規模やリスクに応じて、必要な成果物を選択する。すべての成果物を必ず作る必要はない。ただし、目的、要求、要件、データ、AI、ルール、評価、変更管理を確認できる最低限の成果物は必要である。
なお、本書では「～書」や「～台帳」と表現する成果物が多く存在するが、紙面で記載することを前提とするものではなく、デジタルデータで管理することを前提としている。

16.1 要求定義書と要求管理台帳の位置づけ

要求定義書とは、要求並びにその背景、根拠、発生源、優先度、関連要求、検証方法及び関連する図、表、モデル等を、関係者が体系的に確認できる形で整理した文書又はビューである。個々の要求に関する情報に加え、複数の要求に関係する業務フロー、ユースケース、データモデル等を含め、要求定義の全体像を示す役割を持つ。
ただし、要求定義書は、すべてのプロジェクトで独立した文書として作成する必要はない。要求に関する情報が、要求管理台帳、業務フロー、ユースケース、データ定義、受入条件その他の成果物に分かれて管理されていても、要求 ID などの識別子によって相互の関係を追跡でき、関係者が必要な情報を確認できる場合には、これらの成果物全体を要求定義書に相当する成果物として扱うことができる。
要求定義書と要求管理台帳の双方を作成する場合には、同一の情報を重複して管理することを避ける必要がある。要求文、優先度、状態、版及び変更履歴等について正本となる管理対象を定め、他方か

らは参照、リンク又は自動生成によって利用することが望ましい。例えば、個々の要求とその管理属性は要求管理台帳を正本として管理し、要求定義書には、要求管理台帳では表現しにくい説明文、業務フロー、図、モデル、意思決定の経緯等を整理する方法がある。この場合も、要求 ID などを用いて要求管理台帳と要求定義書との対応関係を明確にする必要がある。

16.2 AI 時代の要求管理での代表的な成果物

表 54 AI 時代の要求管理での代表的な成果物

	意義	概要
目的・ミッション整理	何のために行うかを明確にする	事業目的、業務目的、社会的価値、成功指標を整理する
ステークホルダー一覧	関係者と責任を明確にする	利用者、業務部門、IT、AI、データ、法務、監査等を整理する
要求一覧	実現したいことを整理する	要望から抽出した要求を分類し、優先度や根拠を付ける
要求定義書	要求とその背景及び関連情報を体系的に確認できるようにする	要求、根拠、発生源、優先度、関連要求、検証方法並びに関連する図、表、モデル等を整理する。独立した文書を作成せず、個別成果物を要求 ID 等で関連付ける構成としてもよい。
要件定義書	実現すべき事項を合意する	機能、非機能、AI、データ、ルール、運用、評価を記述する
業務フロー	業務の流れを可視化する	作業手順、承認、判断、例外処理、責任者を示す
データ定義	データの意味と品質を明確にする	データ項目、定義、出典、形式、品質、利用条件を整理する
AI 要求管理	AI の使い方と管理方法を明確にする	利用目的、入力、出力、人間の関与、評価、監視、更新を定義する
ルール要求一覧	法令・標準・社内規則との関係を示す	外部要件・社内ルールを業務やシステム要件へ展開する
機能要件一覧	システムが提供すべき機能を示す	機能、処理、入出力、振る舞いなどを整理する
非機能要件一覧	システムが満たすべき品質条件を明確にする	性能、可用性、セキュリティ、保守性、拡張性などを整理する
受入条件	完了判断を明確にする	要件が満たされたことを確認する基準を定める
トレーサビリティ表	変更影響と説明責任を支える	目的、要求、要件、設計、テスト、ログの対応を示す
要求評価・妥	要求の評価・妥当性確認	要求一覧に紐づける形で、要求ごとの評価・妥当性

当性確認結果	の結果を記録する	確認結果、次アクションを記述する
要求管理台帳	要求を継続的に管理する	要求 ID、状態、優先度、変更履歴、検証方法を管理する
変更管理記録	変更の理由と影響を残す	変更要求、影響分析、承認、対応結果を記録する
運用・監視計画	運用後の改善につなげる	監視項目、ログ、評価頻度、責任者、改善手順を整理する
構造化文書	AI やツールで活用しやすくする	Markdown、XML、Akoma Ntoso 型構造、ReqIF 等で管理する

成果物は、多ければよいわけではない。重要なのは、関係者が合意でき、実装に使え、評価でき、変更時に参照できることである。

小規模なプロジェクトでは、目的整理、要求一覧、要求管理台帳、要件定義書、受入条件のみでもよい。AI を利用する場合は、これに AI 要求管理とデータ定義を加える。法令・標準・社内規則に関係する場合は、ルール要求一覧とトレーサビリティ表を加える。

高リスクなプロジェクトでは、業務フロー、データモデル、AI 利用モデル、リスク評価、監査証跡、変更管理記録まで整備することが望ましい。

成果物は、作成して終わりではない。運用後も、業務・制度・利用環境の変化、新たな知見等に応じて、変更理由と影響を明確にした上で必要な見直しを行う。要求定義書・要件定義書、要求管理台帳、データ定義、AI 要求管理、ルール要求一覧は、変更管理の対象として扱う必要がある。

おわりに

これまでの話を総括する形で重要なポイントを繰り返す。
まず、AI 時代の要求管理は、開発前に一度限りで実施する作業ではない。目的を確認し、要求を獲得・分析・仕様化し、モデル化し、評価し、実装へ接続し、運用後のデータをもとに改善する循環的な活動である。
また、要求管理は個人の職人芸ではなく、組織能力として整備すべきものである。標準プロセス、成果物、役割分担、テンプレート、要求管理台帳、変更管理、教育、ツール活用を組み合わせることで、要求管理の品質を高めることができる。
そして、AI 時代において重要なのは、AI に要求管理を任せることではない。AI を活用しながら、人間が目的、価値、リスク、制約、評価を明確にし、要求を継続的に管理・更新できる状態を作ることである。
要求管理を組織能力として高めることが、変化に強い企業と社会を作る基盤となる。
本書が、読者それぞれの立場から要求管理の在り方を見直し、身近な業務やプロジェクトにおける実践の第一歩を踏み出す契機となることを期待する。

参考文献

[1] Systems and software engineering — Life cycle processes — Requirements engineering（ISO/IEC/IEEE 29148:2018）, ISO, 2018.

[2] Systems and software engineering — System life cycle processes（ISO/IEC/IEEE 15288:2023）, ISO, 2023.

[3] IIBA, A Guide to the Business Analysis Body of Knowledge® (BABOK® Guide).

[4] INCOSE, Guide to Writing Requirements, 2023.

[5] システム及びソフトウェア技術－システムライフサイクルプロセス（JIS X 0170:2025）, 日本規格協会, 2025.

[6] A. Trabelsi, I. Benzarti, H. Mili and D. Ameyed, "From Business Problems to AI Solutions: Where Does Transformation Support Fail?," 2026.

[7] A. Chan, T. Polacsek, F. Bouissière , P. D. Vecchia, “A Serious Game for Cross-Functional Alignment in Industrial Goal Modeling,” 2026.

[8] H. Saeeda, M. Kim, E. Knauss, J. Thyssen, J. Ørting, J. L. Korsgaard , N. J. Strøm, “Deriving and Validating Requirements Engineering Principles for Large-Scale Agile Development: An Industrial Longitudinal Study,” 2026.

[9] D. Jin, Z. Jin, Y. Yang, L. Li, Z. Fang , Y. He, “From Chat to Interview: Agentic Requirements Elicitation with an Experience Ontology.,” 2026.

[10] Z. Wu, X. Chen, Z. Jin, M. Hu , D. Jin, “Unlocking the Silent Needs: Business-Logic-Driven Iterative Requirements Auto-completion,” 2026.

[11] R. Tanahashi, H. Washizaki, N. Ubayashi, R. Sugiyama, S. Okuda , K. Toriumi, “Associated Personas: Linking Stakeholders to Traceable Requirements,” 2026.

[12] H. Cheng, M. Kim, F. Khomh, T. Racharak, N. Yoshioka , N. Ubayashi, QUARE: Quality-Aware Requirements Analysis Through Multi-Agent Dialectical Negotiation, 2026.

[13] Y. Huang, M. Hu, X. Chen, Z. Jin , S. Xiao, “Modeling Like Peeling an Onion: Layerwise Analysis-Driven Automatic Behavioral Model Generation,” 2026.

[14] C. Huang, Y. Yao, T. Chen , X. Ma, “From Use Cases to Sequence Diagrams: Schema-Constrained Generation with Large Language Models,” 2026.

[15] B. Guindani, M. Camilli, L. Lestingi , M. M. Bersani, “Agentic Generation of

Structured Clinical Specifications for Digital Healthcare Services," 2026.

[16] T. Hey , J. Franttini, "How Requirements Quality Makes (or Breaks) Traceability Link Recovery," 2026.

[17] M. Pradel, "Testora: Using Natural Language Intent to Detect Behavioral Regressions," 2026.

[18] Z. Tian , J. Chen, "Aligning Requirement for Large Language Model's Code Generation," 2026.

[19] A. Yadavally , T. N. Nguyen, "From Seed to Scope: Reasoning to Identify Change Impact Sets," 2026.

[20] A. Bennaceur, G. K. Rajbahadur, P. Mercy, B. Nuseibeh , F. Alrimawi, "From Failure to Alignment: A Requirements Engineering Framework for Machine Learning Systems," 2026.

[21] D. Alrajeh, V. Nowack, P. Benjamin, K. Thomas, W. Hobson, C. G. Munoz, Catherine Hamilton-Giachritsis, J. Kloess, J. Woodhams, D. Butler, M. Law, R. Morton, B. Costello, A. Burrell, T. Grant, P. Shah, F. L. d. Leon , M. Lee, Data-Dependent Goal modeling for ML-Enabled Law Enforcement Systems, 2026.

[22] H. Wang, C. Poskitt , J. Sun, AgentSpec: Customizable Runtime Enforcement for Safe and Reliable LLM Agents, 2026.

[23] A. Mavridou, H. Taylor, S. Lozito, L. Dennis, M. Fisher , M. Farrell, A Taxonomy of Human-Robot Teamwork Requirements, 2026.

[24] INCOSE, システムズエンジニアリングハンドブック 第 5 版, D. D. Walden, 編, 慶應義塾大学出版会, 2025.

[25] E. C. Honour, "Systems engineering return of investment," 2013.

[26] Smart Words, "How IT-Projects really work," [オンライン]. Available: https://www.smart-words.org/jokes/how-it-projects-really-work.html.

[27] JISA, "要求工学 REBOK," [オンライン]. Available: https://www.jisa.or.jp/it_info/engineering/tabid/1165/Default.aspx.

[28] European Union, "Regulation (EU) 2024/1689 of the European Parliament and of the Council of 13 June 2024 (Artificial Intelligence Act).," [オンライン]. Available: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A32024R1689.

[29] Object Management Group（OMG）, "Requirements Interchange Format (ReqIF), Version 1.2," [オンライン]. Available:

https://www.omg.org/spec/ReqIF/1.2/PDF.

[30] OASIS Open, “OSLC（Open Services for Lifecycle Collaboration）,” [オンライン]. Available: https://open-services.net/.

[31] OASIS, “Akoma Ntoso Version 1.0,” [オンライン]. Available: https://www.oasis-open.org/standard/akn-v1-0/.

[32] INCOSE, “MBSE Initiative Working Group,” [オンライン]. Available: https://www.incose.org/group/mbse-initiative/.

[33] T. Gruber, "definition-of-ontology," [Online]. Available: https://tomgruber.org/writing/definition-of-ontology.pdf.

[34] IIBA, “What is Business Analysis?,” [オンライン]. Available: https://www.iiba.org/professional-development/career-centre/what-is-business-analysis/.

[35] M. Obaidi, J. Droste, H. Deters, M. Herrmann, M. Krahl , K. Schneider, “From Quality Properties to Practice: a Guideline and Workflow for Explainability Requirements,” 2026.

[36] T. Sporsem, S. R. Finserås, L. Adde , I. Strümke, “What Does Explainable AI Mean in Practice? Evaluative Requirements from a Longitudinal Clinical Case Study,” 2026.

[37] A. Indika, R. Kazman , A. Peruma, “Practitioner Views on Mobile App Accessibility: Practices and Challenges,” 2026.

[38] L. M. D. Abodinar, P. Adubi, R. Ayad, B. Hogedez, I. Standen , B. Townsend, “CRAYON: Supporting the Elicitation of Pluralistic Normative Requirements for Autonomous Systems,” 2026.

[39] R. Lüthy , N. Seyff, “Letting the AI Decide What the AI Should Code? on Responsibility and Accountability in Requirements-Driven Software Evolution,” 2026.

[40] M. Ulloa, J. L. Butler, S. Haniyur, C. Miller, B. Amos, A. Sarkar , M.-A. Storey, “Product Manager Practices for Delegating Work to Generative AI: ``Accountability must not be delegated to non-human actors'',” 2026.

[41] H. Saeeda, J. Yu, T. Johansson, E. Knauss , F. Warg, “Annotation Requirements to Annotation Errors: A Causal Modelling Analysis for AI-Enabled Perception Systems Development,” 2026.

[42] D. Mendoza, A. Mavridou, A. Katis , C. Trippel, “Automating Requirements Formalization: Using LLMs and Low-Complexity Distinguishing Traces for

Semantic Validation,” 2026.