International Journal of Multidisciplinary Research and Growth Evaluation.

From Ambiguity to AI-Ready Requirements: A Business Analysis Framework for Structuring Business Problems for Intelligent Process Automation

Joseph Idanesi Alieme [1*], Aumbur Sule [2], Valentina Ochuko Obukadata [3]
[1] Academy of Applied Sciences WSZiA in Opole, Poland
[2] Access Bank Plc, Nigeria
[3] University of New Haven, Connecticut United States, USA

* Corresponding Author: **Joseph Idanesi Alieme**

Article Info

ISSN (online): 2582-7138
Volume: 05
Issue: 06
November-December 2024
Received: 28-10-2024
Accepted: 26-11-2024
Published: 24-12-2024
Page No: 2088-2117

Abstract
This paper argues that intelligent process automation fails at specification more often than at implementation. Requirements engineering offers a mature account of ambiguity as a defect to be removed and a mature account of the quality characteristics of a well-formed requirement, but neither transfers cleanly to processes whose automation depends on judgement and on probabilistic components. This paper develops the Ambiguity-to-Specification (A2S) pipeline, a five-gate business analysis framework; problem framing, decision decomposition, data grounding, behavioural specification, and oversight and acceptance, that converts an ambiguously stated business problem into a specification an intelligent automation can be built and governed against. The framework rests on three arguments: that the unit of specification shifts from system behaviour to the decision; that ambiguity is not uniformly a defect but a resource whose resolution should be routed to the gate at which it is cheapest and most informative, leaving a residue that must be specified rather than resolved; and that probabilistic components break the correctness contract on which classical quality criteria rest. A second contribution extends the requirement quality characteristics of ISO/IEC/IEEE 29148 with six AI-readiness characteristics; decidability, data-groundedness, error-tolerance specification, escalation specification, observability and drift sensitivity, each with a test question, an evidencing artefact and a corresponding requirement smell. The framework is illustrated on a hypothetical mid-sized services organisation performing document-heavy casework, and eight propositions are stated for empirical evaluation. The paper is conceptual: no primary data were collected and the framework has not been validated.

DOI: https://doi.org/10.54660/.IJMRGE.2024.5.6.2088-2117

Keywords: business analysis, requirements engineering, ambiguity, intelligent process automation, decision modelling, requirements quality, AI governance

1. Introduction

It is argued here that organisations attempting intelligent process automation rarely discover that the technology does not work. They discover, usually late, that the business problem was never stated in a form a machine could act on. The distance between the way a business states a desired outcome and the level of precision an automated system requires to produce it is what this paper calls the specification gap. This paper is about the method by which a business analyst closes it.

A frequently cited industry observation places early RPA project failure at thirty to fifty per cent (EY, 2016); it should be read as practitioner experience rather than as a measured rate, and the academic literature has recirculated it without independent verification. A Deloitte survey of 479 executives across 35 countries reported that "seventy-four per cent of survey respondents

are already implementing RPA, and 50 per cent are already implementing OCR", naming process fragmentation, lack of clear vision, IT readiness and workforce resistance as barriers to scaling (Telford *et al*., 2022). Both are vendor-adjacent instruments of unclear sampling, cited as indications of practitioner concern rather than as measurements; what they suggest is that the difficulty sits upstream of the tooling. A further caution is conceptual: success and failure of an information system are enacted judgements that different parties can hold simultaneously about the same system, so a headline failure rate is analytically empty however collected (Cecez-Kecmanovic, Kautz & Abrahall, 2014).

Earlier work in this programme argued two preconditions: that decision logic and management reporting must be standardised before they can be automated, since a specification cannot be written against a process that varies without being documented as varying; and that organisational readiness for AI is not a technological property and must be sequenced rather than assumed. Neither precondition has been empirically tested, and both are advanced here as argued positions rather than established findings. One question remains. Granted a standardised process in a ready organisation, how does an analyst turn "the backlog is unmanageable" into something that can be built, tested and governed against?

The contribution is twofold. The Ambiguity-to-Specification (A2S) pipeline treats specification as a staged transformation across five gates rather than a single elicitation event, and ambiguity as something to be *routed* rather than uniformly eliminated. A proposed extension to the requirement quality characteristics of ISO/IEC/IEEE 29148 (International Organization for Standardization, 2018) adds six AI-readiness quality characteristics, with test questions, evidencing artefacts and corresponding smells. Table 2, mapping ambiguity types to the gate at which each is most cheaply resolved and to its consequence if unresolved, is the signature artefact. Section 2 names the gap in the research literature; Section 3 surveys how the same specification problem presents itself in applied and practitioner writing across several domains, and states explicitly what evidential weight that survey does and does not carry; Section 4 states what kind of contribution this is; Section 5 develops the framework; Section 6 illustrates it; Sections 7 to 9 give implications, limitations and conclusions.

2. Background and Related Literature

2.1. Requirements Engineering Foundations

Pohl's (2010) three-dimensional framework: content, agreement, documentation, supplies the most useful decomposition here, because the dimensions come apart in practice and much of what appears as a specification defect is an unresolved agreement defect. The process accounts of Sommerville and Sawyer (1997) and Kotonya and Sommerville (1998) remain reference descriptions of good practice, and Wiegers and Beatty (2013) the most widely used practitioner treatment; Robertson, Robertson and Reed (2024) supply the Volere shell, notable here because it forces the analyst to record a fit criterion alongside each requirement. Nuseibeh and Easterbrook (2000) framed the field's enduring difficulty as the coordination of partial descriptions held by parties with different stakes and vocabularies. Two traditions matter disproportionately to automation: goal-oriented requirements engineering (van Lamsweerde, 2009), which refines an objective into satisfiable sub-goals with assigned responsibility; and business analysis practice as codified by the International Institute of Business Analysis (2015) and by Paul, Cadle, Eva, Rollason and Hunsley (2020).

Two further traditions supply structures the pipeline reuses. Cockburn's (2001) goal-levelled use cases, with a main success scenario and enumerated extensions, established that a behavioural specification is incomplete until departures from the nominal path are stated, the ancestor of the escalation characteristic of Section 5.7. Agile practice moved the other way, trading traceability and non-functional coverage for responsiveness (Inayat *et al*., 2015), and small organisations manage requirements through informal, culture-carried arrangements rather than documented process (Aranda, Easterbrook & Wilson, 2007).

2.2. Evidence on Elicitation

Pacheco, García and Reyes (2018) reviewed the period 1993–2015 and analysed 140 studies, classifying elicitation techniques by maturity; Dieste and Juristo (2011) aggregated the experimental evidence and found interviews to perform comparatively well on the measures the primary studies used; Zowghi and Coulin (2005) give the standard taxonomy. Practice is less orderly: Palomares *et al*. (2021), interviewing 24 practitioners across 12 companies, found group interaction techniques; meetings and workshops, dominant, and techniques typically combined rather than selected on evidential grounds. Bano and Zowghi (2015) found the user-involvement evidence heterogeneous and methodologically uneven; their review is best read as showing that the repeated claim of a straightforward positive relationship between involvement and success is less securely established than its repetition implies.

The comparative tradition is older. Goguen and Linde (1993) distinguished interviews, introspection, protocol analysis and discourse analysis by their sensitivity to what informants can articulate, and Porter, Votta and Basili (1995) supplied the replicated-experiment template for comparing detection methods that Section 7.6 adopts.

2.3. Requirements Quality as A Measured Construct

Requirement quality is itself a studied construct. Montgomery, Fucci, Bouraffa, Scholz and Maalej (2022) filtered 6,905 retrieved articles to 105 relevant primary studies and identified 12 quality attributes and 111 sub-types, ambiguity, completeness, consistency and correctness being the most prominent; none of them concerning the availability of an input when a decision is taken, the direction of a tolerated error, or the behaviour of a system that is unsure.

Frattini, Montgomery, Fischbach, Mendez, Fucci and Unterkalmsteiner (2023) argue that the field lacks a shared theory and propose one organising quality factors, the activities they affect and the impacts that follow. The move that matters is relational: a factor is defective not in itself but in virtue of what it does to a downstream activity; the position Femmer and Vogelsang (2019) compress as requirements quality is quality in use. In intelligent process automation, these activities include model training, threshold setting, verification over a population, and oversight design. Which factors matter is also local: Frattini (2024) reports a case study identifying 17 factors and 11 interaction effects within one company.

Operational models exist: the Quality User Story framework defines 13 quality criteria implemented in the AQUSA tool

and evaluated on 1,023 user stories from 18 software companies (Lucassen *et al*., 2016), and the lineage runs back to the Automated Requirements Measurement indicators of Wilson, Rosenberg and Hyatt (1997), the ancestor of the requirements-smells approach on which Section 5.7 builds.

2.4. Controlled Natural Language and Requirements Templates

A long-running response to ambiguity constrains the syntax in which requirements may be written: the Easy Approach to Requirements Syntax supplies five patterns ubiquitous, event-driven, state-driven, optional-feature and unwanted-behaviour that remove structural ambiguity without formal notation (Mavin *et al*., 2009; Mavin & Wilkinson, 2019), and the SOPHIST sentence templates codified in the IREB body of knowledge prescribe the grammatical skeleton of a requirement sentence (Rupp & Pohl, 2015).

The apparatus is also checkable: conformance to such a boilerplate can be cast into pattern matchers and checked automatically even where the glossary is incomplete (Arora, Sabetzadeh, Briand & Zimmer, 2015), and the glossary itself can be extracted and clustered automatically (Arora *et al*., 2017).

That apparatus is necessary and structurally insufficient, because a template constrains the *form* of a statement about behaviour and is silent on its *content*. "When the system receives an application, the system shall assess eligibility" is a well-formed event-driven statement and an unusable specification: it names no alternatives, no inputs, no error profile, and no behaviour under uncertainty. It would pass an automated conformance check and a glossary check alike. The characteristics proposed in Section 5.7 are therefore content characteristics rather than form characteristics, and the automated checks proposed in Section 7.4 are checks of relations between artefacts rather than of sentence shape.

2.5. Ambiguity

Berry and Kamsties (2004) supply the canonical typology of lexical, syntactic, semantic and pragmatic ambiguity, alongside vagueness, generality and outright language error, and the more consequential observation that the practical problem is not that requirements are ambiguous but that their ambiguity goes *undetected*: each reader resolves it silently and consistently with their own frame, and divergence surfaces downstream. Kamsties (2005) distinguishes linguistic ambiguity from ambiguity arising in the application and system domains, where a term is linguistically clear but underdetermined relative to the domain of interpretation. Ferrari, Spoletini and Gnesi (2016) complicate this productively: drawing on 34 customer–analyst interviews, they develop a four-part framework of unclarity, multiple understanding, incorrect disambiguation and correct disambiguation, and argue that ambiguity in elicitation can be a *productive resource* for surfacing tacit knowledge, since noticing that a term is used in two ways is the moment at which unarticulated domain knowledge becomes articulable. Gervasi, Ferrari, Zowghi and Spoletini (2019) propose a unifying framework spanning written requirements and spoken elicitation, since ambiguity occurs in documents but is not exhausted by them. It also transfers beyond software: Massey, Rutledge, Antón and Swire (2014) report that participants using their taxonomy of regulatory ambiguity identified on average 33.47 ambiguities in 104 lines of legal text in 50 minutes, with 97.5% coverage of those found; casework criteria often originate in such language.

2.6. Automated Ambiguity Detection, Inconsistency and Their Limits

Ferrari and Esuli (2019) demonstrate an NLP approach to cross-domain ambiguity, where a term carries different meanings in different technical communities, and Ferrari *et al*. (2018) report an industrial application of NLP defect patterns validated on 1,866 expert-annotated requirements from a railway signalling manufacturer. The tool lineage is longer than either (Lami, Fusani & Trentanni, 2019), and two of its design principles are adopted here: separate flagging from explanation and leave the judgement to a human (Kiyavitskaya, Zeni, Mich & Berry, 2008); and target complete recall over a narrow, decidable vocabulary rather than precision over an open-ended one (Tjong & Berry, 2013).

Detection now reaches past ambiguity itself. Anaphoric ambiguity is a measurable class whose reader disagreement a trained classifier can predict (Yang, de Roeck, Gervasi, Willis & Nuseibeh, 2011), and a masked language model evaluated over 40 specifications highlighted missing terminology (Luitel, Hassani & Sabetzadeh, 2024). Incompleteness detection is the closer analogue of an AI-ready check, since the defects proposed in Section 5.7 are predominantly absences.

The field-level picture comes from Zhao *et al*. (2021), whose systematic mapping covered 404 studies over 36 years across 170 venues and found that 67.08% were solution proposals validated only by laboratory experiment or example, that approximately 7% had been assessed industrially, and that of 130 identified tools only 17 (13.08%) remained available. Automated detection is therefore useful triage at volume, but not a substitute for deciding which ambiguities matter, to whom, and where they should be resolved.

Three refinements bear on the routing principle of Section 5.1. Chantree, Nuseibeh, de Roeck and Willis (2006) distinguish *innocuous* ambiguity, where readers converge on one reading, from *nocuous* ambiguity, where they diverge, and argue that only the latter is worth acting on. Ezzini, Abualhaija, Arora, Sabetzadeh and Briand (2022) report that the best solution for anaphoric ambiguity detection achieves an average precision of approximately 60% at a recall of 100%, against approximately 98% success for interpretation once ambiguity is established, so automation belongs as an imprecise net feeding human adjudication. And ambiguity is not the only surviving defect: Gervasi and Zowghi (2005) detect inconsistency between natural-language requirements by partial formalisation with default reasoning, which matters because rules elicited from different stakeholders' conflict.

Whether tools outperform careful reading is doubtful: Dalpiaz, van der Schalk, Brinkkemper, Aydemir and Lucassen (2019) found manual inspection delivered significantly higher recall than tool support, and Berry, Gacitua, Sawyer and Tjong (2012) conclude that a transparent tool doing an identifiable part of such a task may be better than an intelligent one failing unpredictably at the whole. Detection is also not confined to documents: Spoletini, Ferrari, Bano, Zowghi and Gnesi (2018) found 68% of the ambiguities discovered in their study came from

structured post-hoc review of elicitation interviews rather than from the interviews themselves, where 32% were caught. The community's own assessment is that capability has outrun adoption (Dalpiaz *et al*., 2018).

2.7. Cognitive Bias, Tacit Knowledge And Productive Ambiguity

If ambiguity can be productive, the reason lies in what elicitation is up against. Polanyi's (2009) claim that we can know more than we can tell names the difficulty; Gervasi *et al*. (2013) decompose tacit knowledge into facets, arguing that it creates risk precisely where professionals from different backgrounds lack common ground; Sutcliffe and Sawyer (2013) find domain-knowledge modelling weakly supported by existing technique; and Berry (1995) argues that the analyst's ignorance of the domain is an asset, because it forces the tacit to be stated. Nonaka and Takeuchi's (1995) tacit-to-explicit conversion model explains why documentation alone cannot discharge this: externalisation is interactional, not retrieval.

Requirements work also has framing effects internal to it. Mohanani, Ralph and Shreeve (2014) report experimental evidence of *requirements fixation*: the form in which requirements are presented anchors the design thinking that follows, which is the strongest warrant for the ordering principle of Section 5.1. Fixation is one instance of a wider pattern (Mohanani, Salman, Turhan, Rodríguez & Ralph, 2020), and misleading information embedded in a requirements document has been shown to shift professional effort estimates in a randomised field experiment (Jørgensen & Grimstad, 2011): a number written into a draft requirement is not inert. And some vagueness is not a drafting failure at all. Endicott (2000) argues that vagueness in legal rule systems is ineliminable and sometimes functional, since a rule precise enough to remove every borderline case will misclassify cases the rule-maker wanted decided otherwise.

2.8. Prioritisation, Traceability and The Provenance of a Requirement

Two adjacent literatures supply constructs the pipeline relies on. Prioritisation research is large and fragmented: Achimugu, Selamat, Ibrahim and Mahrin (2014) reviewed 73 primary studies covering 1996 to 2013 and identified 49 distinct techniques with little cumulative comparison among them, so criterion weighting is elicited and recorded rather than derived. Ranking was originally framed as a pairwise cost-versus-value trade-off yielding a ratio scale (Karlsson & Ryan, 1997), and Berander and Andrews (2005) give the canonical taxonomy of what is traded; but the scarcity is of criteria rather than technique (Riegel & Doerr, 2015).

Traceability matters more, because the observability characteristic proposed in Section 5.7 is a traceability requirement in a new setting. Gotel and Finkelstein (1994) argued that the traceability problem is largely a pre-specification one of lost rationale and provenance, and Mäder and Egyed (2015) report controlled-experiment evidence that traceability improves the speed and correctness of maintenance tasks. Ramesh and Jarke (2001) add that practice divides into low-end and high-end reference models differing in what is traced and why, and Cleland-Huang, Gotel, Huffman Hayes, Mäder and Zisman (2014) make purposed traceability trace with a named interrogator a first-order challenge; automatic recovery of links is less secure (Borg, Runeson & Ardö, 2014). In intelligent automation the rationale for a threshold, the population over which an error rate was agreed and the model version in force are provenance facts, and an automation that does not emit them cannot be audited against its own specification.

2.9. Non-Functional and Quality Requirements

The classical treatment is the NFR Framework of Chung, Nixon, Yu and Mylopoulos (2000), which represents quality goals as softgoals and treats them as *satisficed* rather than satisfied; its insight is that quality goals conflict and that the analyst's task is to make the trade-off explicit.

The category is contested. Glinz (2007) documents inconsistent use of the term and proposes a faceted classification by kind, representation, satisfaction and role; Mairiza, Zowghi and Nurmuliani (2010) document the absence of an agreed taxonomy and the domain-dependence of which attributes matter; and Eckhardt, Vogelsang and Méndez Fernández (2016) analysed 530 non-functional requirements from 11 industrial specifications and found most describe system behaviour. Ameller, Ayala, Cabot and Franch (2012), interviewing 13 software architects, found such requirements often decided by architects rather than elicited, which is how an error profile comes to be set by whoever implements the model. If so, the distinction that does real work is not functional versus non-functional but *determinate* versus *rate-valued*. For machine-learned components that becomes the specification problem entire: classical NFR techniques do not transfer cleanly, because accuracy, fairness and explainability interact and conflict (Horkoff, 2019), and practitioners rarely measure such attributes at all (Habibullah *et al*., 2023). The position here is stronger: where components are right at a rate, the properties conventionally filed as non-functional *are* the requirements, because they exhaust what remains to specify once the decision has been modelled.

2.10. State of Practice and Its Consequences

The NaPiRE initiative (Wagner *et al*., 2019) assembled a theory and a global family of surveys of how requirements engineering is actually practised, and Méndez Fernández *et al*. (2017) provide the strongest available support for the proposition that requirements problems have downstream project consequences; Hofmann and Lehner (2001) supply the earlier observation that requirements practice functions as a success factor in software projects. Requirements defects are therefore consequential, common, and recognised as such by the practitioners who produce them.

2.11. Automation and Intelligent Automation

Robotic process automation is best characterised, following van der Aalst, Bichler and Heinzl (2018), as "outside-in" automation operating at the user-interface level and leaving underlying systems unchanged; the citation is an editorial, and its value lies in that precision rather than in evidence. This deterministic character is what makes RPA specifiable in conventional terms: a routine either replays or it does not. Syed *et al*. (2020) survey contemporary themes and challenges, Wewerka and Reichert (2021) supply a systematic mapping and classification framework, Syed and Wynn (2024) consolidate the state of the art across lifecycle, selection, implementation and open problems, and Ivančić, Suša Vugec and Bosilj Vukšić (2019), reviewing 27 papers from 2016–2019, document how thin the early academic base was. Plattfaut *et al*. (2022) identify critical success factors;

Herm *et al*. (2023) offer a three-phase, eleven-stage framework built from 35 cases, 8 expert interviews and 5 validation workshops; Huang and Vasarhelyi (2019) show, in auditing, how such frameworks adapt where the output carries evidential weight.

Two strands locate the difficulty where this paper locates it. Viability is assessable at activity granularity against explicit criteria rather than by intuition at process level (Wellmann, Stierle, Dunzer & Matzner, 2020), and analysis and design rather than coding are where such projects are won or lost (Jimenez-Ramirez, Reijers, Barba & Del Valle, 2019). On the concept's boundary, Chakraborti *et al*. (2020) anchor intelligent process automation as an extension of rule replay towards learning, unstructured input and dialogue; Engel, Ebel and Leimeister (2022) define cognitive automation distinctly; and Engel, Elshan, Ebel and Leimeister (2024) argue such use cases need their own assessment model. The setting assumed throughout is AI augmentation of process-aware systems (Dumas *et al*., 2023). One analytics result underwrites Gate 4: prediction and decision are formally distinct problems, so an accurate model does not by itself yield a better operational decision (Bertsimas & Kallus, 2020).

2.12. Requirements Engineering for Machine Learning and Ai Systems

The literature closest to this paper's contribution is requirements engineering for machine-learned and AI-enabled systems, referred to below as RE4AI. Much of what the A2S pipeline needs is established there, and what is claimed here is a rearrangement of that programme's content rather than a replacement of it.

Its founding observations are three. Vogelsang and Borg (2019), interviewing four data scientists, supplied the most-cited framing: professionals must understand machine-learning performance measures, must elicit quality requirements such as explainability and freedom from discrimination, and must accommodate a shift from coding to training. Rahimi, Guo, Kokaly and Chechik (2019) decompose a domain concept into specifications of the real-world concept, the dataset and the learned model data as a specification artefact. Belani, Vuković and Car (2019) name the difficulties as data requirements, uncertainty and evolving context, and Ahmad, Bano, Abdelrazek, Arora and Grundy (2021) survey where notations break down.

Both mappings report immaturity. Villamizar, Escovedo and Kalinowski (2021) identified 35 studies reporting a lack of validated techniques and a fragmented understanding of non-functional requirements for machine learning; Ahmad, Abdelrazek, Arora, Bano and Grundy (2023) identified 43 and found current applications not adequately adaptable; and Giray (2021) finds requirements, testing and maintenance among the least well-served areas. This paper inherits those assessments.

What remains open differs in four respects, each of framing rather than content. The unit: that literature specifies a component and its envelope, whereas the analyst specifies a decision inside a process, of which the component may be one contributor among rules, humans and integrations. Ambiguity: the mappings report poorly adapted techniques without treating ambiguity as an object with a type, a cost and a point of resolution. Ordering: nothing states what must be settled before what. Terminal state: the implicit goal there is completeness, whereas the position here is that such a specification contains an explicit residue.

2.13. Software Engineering for Machine Learning and Production Practice

A parallel literature examines what happens to machine-learned components once built. Sculley *et al*. (2015) name the system-level debt invisible at model level boundary erosion, entanglement, hidden feedback loops, undeclared consumers, data dependencies and configuration issues. Amershi *et al*. (2019), in a case study with software teams at Microsoft, describe a nine-stage workflow, and identify data discovery, management and versioning as where such development differs most sharply from other software engineering. Martínez-Fernández *et al*. (2022), surveying 248 studies published between January 2010 and March 2020, found data-related issues to be the most recurrent challenges, and Nahar, Zhou, Lewis and Kästner (2022), interviewing 45 practitioners from 28 organisations, locate the hardest problems at three collaboration points: requirements and planning, training-data handling, and product–model integration, rather than in algorithms. Paleyes, Urma and Lawrence (2022) organise recurring deployment failures by lifecycle stage, Kreuzberger, Kühl and Hirschl (2023) define MLOps and its principles, components and roles, and Ashmore, Calinescu and Paterson (2021) state assurance desiderata per lifecycle stage, which map onto the artefacts of Gates 3 to 5. Steidl, Felderer and Ramler (2023) characterise the continuous-development pipeline for such systems as distinct from conventional continuous integration, cataloguing the triggers that cause it to execute, which is what a drift-sensitivity clause binds to.

2.14. Explainability, Fairness and Human Oversight

The oversight literature has a technical strand and a human-factors strand. Technically, interpretability lacked an agreed definition and evaluation protocol as the field scaled: Doshi-Velez and Kim (2017), in a preprint never formally peer-reviewed, note the absence of consensus on what interpretable machine learning is or how it should be measured. The space of methods is itself large enough to require formal classification (Guidotti *et al*., 2018): explainability names a family, not a technique. Miller (2019) argues that what counts as a good explanation is an empirical question already addressed by philosophy, psychology and cognitive science, and Rudin (2019) that in high-stakes settings an inherently interpretable model is the stronger control. Together these establish that "the system shall be explainable" is not specifiable: such a requirement must name a recipient, a task and a criterion of adequacy.

Fairness enters for the same reason. It admits mutually incompatible formal definitions, and bias enters at many points in a pipeline (Mehrabi, Morstatter, Saxena, Lerman & Galstyan, 2021), so a requirement not naming a definition defers a choice made silently downstream. Barocas and Selbst (2016) account for how ordinary modelling choices produce discriminatory outcomes, and Selbst, boyd, Friedler, Venkatasubramanian and Vertesi (2019) show that technical interventions fail through abstraction traps, of which the framing trap is the one this framework avoids. Practitioner needs diverge from what research supplies (Holstein *et al*.,

2019), and public-sector practitioners report accountability needs that are organisational rather than statistical (Veale, Van Kleek & Binns, 2018).

The human-factors strand is older and, here, more consequential, because it converts Green's (2022) critique of nominal oversight from a policy argument into an empirically grounded prediction. Bainbridge (1983) observed that automating the tractable part of a task leaves the human the residual hard part while eroding the practice on which competence depends. Parasuraman and Riley (1997) decompose reliance into four separately diagnosable modes: use, misuse, disuse, and abuse, so a control addressing only over-reliance is incomplete. Skitka, Mosier and Burdick (1999) supply the experimental evidence that automation bias produces both omission and commission errors, and Lee and See (2004) identify the objective as *calibrated* trust, reliance matched to actual capability.

The direction of the bias cannot be assumed: Dietvorst, Simmons and Massey (2015) find people abandon a statistically superior algorithm faster than an equally erring human forecaster after one observed error, Logg, Minson and Moore (2019) find conditions under which algorithmic judgement is preferred, and Burton, Stein and Jensen (2020) argue from 61 peer-reviewed articles spanning 1950–2018 that the contradiction reflects divergent rationalities. An oversight requirement naming a human and stopping there therefore specifies a role whose failure modes are predictable and nothing by which they could be observed though the controls needed are not novel in kind, the supervisory guidance of the Board of Governors of the Federal Reserve System and the Office of the Comptroller of the Currency (2011) having long built a regime around validation independence, effective challenge and lifecycle documentation.

2.15. Documentation Artefacts as Partial Specifications

A final strand supplies artefacts that function as specifications after the fact: model cards reporting intended use, explicitly out-of-scope use and disaggregated evaluation (Mitchell *et al*., 2019); FactSheets, which transplant the supplier's declaration of conformity from other regulated industries into AI services (Arnold *et al*., 2019); datasheets documenting provenance, composition and recommended uses (Gebru *et al*., 2021); and the standardised data quality measures of ISO/IEC 5259-2:2024 (International Organization for Standardization & International Electrotechnical Commission, 2024). Hutchinson *et al*. (2021) go further: dataset accountability requires a documented, reviewable development *process* rather than a description attached once the dataset exists.

Whether such artefacts function as controls is contested. Checklists succeed or fail on organisational conditions rather than content (Madaio, Stark, Wortman Vaughan & Wallach, 2020); principle-level guidance is experienced as tensions requiring situated judgement (Sanderson *et al*., 2023); and the algorithmic auditing ecosystem has no accreditation or requirement that findings be acted on (Costanza-Chock, Raji & Buolamwini, 2022). Each artefact documents a component after it exists; none states what the business needed it to do, at what error profile or under whose accountability. They are the natural *evidencing* artefacts for several characteristics proposed in Section 5.7, but the specification must be written elsewhere.

2.16. The Gap

Three discontinuities separate this literature from the specification of intelligent process automation. First, the unit of specification changes: conventional requirements specify system behaviour, whereas automating a judgement-bearing process requires specifying a *decision*, its alternatives, its criteria, the inputs available at the moment it is taken, its tolerated error profile, and its behaviour when the system is not confident. The quality characteristics of ISO/IEC/IEEE 29148 (International Organization for Standardization, 2018) address the first and are silent on the rest. Second, ambiguity is not uniformly a defect: the classical account (Berry & Kamsties, 2004; Kamsties, 2005) is one of defect removal, but Ferrari, Spoletini and Gnesi (2016) show elicitation ambiguity can be productive, so the goal is not earliest possible elimination but routing each ambiguity to where its resolution is cheapest and most informative, and identifying the residue that must be *specified* rather than resolved. Third, probabilistic components break the correctness contract: a rule either fires or it does not, whereas a model is right at a rate, so a requirement that does not state its tolerated error behaviour is not implementable however unambiguous its prose. Existing delivery frameworks (Herm *et al*., 2023; Plattfaut *et al*., 2022) handle process selection and change management but supply no such apparatus. The gap is the absence of a method that takes an ambiguously stated business problem as input, treats ambiguity as routable, and terminates in a specification adequate to a probabilistic, judgement-bearing automation. The requirements-engineering-for-AI literature reviewed in Section 2.12 establishes much of the *content* such a specification must carry; what it does not supply, and what is proposed here, is an ordering of that content by analytical dependency together with a treatment of the ambiguity it routes.

A fourth discontinuity is organisational. The literature above is produced by and for software engineering, whereas specification is normally attempted within a functional business area audit, compliance, data governance, or service management, whose written record sits in a different literature.

3. Decision Specification in Applied Settings

This section surveys applied and practitioner literature of varying peer-review provenance, and, following its expansion, that literature now constitutes a substantial share of this paper's reference list. Many of the contributions cited below appear in venues without an established peer-review record; several are conceptual proposals rather than reports of executed studies; and none has been independently verified for this paper. They are offered for one purpose: to illustrate the *generality* of the specification problem named in Section 2.16, that the difficulty of converting a business intention into something an automated component can be built and governed against recurs across functional areas whose written record sits outside software engineering. They are not evidence for the framework of Section 5 and did not contribute to its construction; this section was written against a completed framework. No statistic, percentage, sample size, effect size or numeric finding anywhere in this paper rests on any source cited here, and this section reports none: every quantitative statement in the paper comes from the peer-reviewed research, standards documents and explicitly

caveated practitioner surveys reviewed in Sections 1 and 2, and the pipeline, its five gates, the six AI-readiness quality characteristics and the eight propositions rest on that base alone. Each source is cited only for the claim its title asserts, and where a title is vague the claim attached is correspondingly modest. A reader discounting this section entirely should find Sections 2, 5 and 7 unaffected, save for the generality argument it alone carries.

The subsections are organised by functional domain. Each state what the decision being automated actually is, what the applied writing reports about specifying it, and which gate the domain's characteristic failure appears to sit at, closing by connecting that evidence to a gate or to one of the six quality characteristics of Section 5.7. The mapping is an ordering device, not a finding.

3.1. Automated and Model-Assisted Decisions in Audit, Fraud Analytics And Financial Controls

The decision automated here is an exception decision: given a transaction, a claim, an account or a control test, decide whether the item is ordinary or warrants attention, and if so whose. It has a named owner, a recurring population and a countable outcome, which makes it the applied setting closest to the casework decision of Section 6. Applied writing here treats the modelling question as largely settled and the specification question as open. Atakpa (2021) describes predictive analytics as the means by which a regulated detection obligation is discharged in commercial banking; Atakpa and Abetoh (2022) review machine learning as an established basis for automated decisioning in a regulated financial process; and Atakpa and Fobellah (2023) model exception identification in operational data as something to be automated rather than inspected item by item. Adesuyi, Akomolafe, Olaogun, Ndukwe and Sakyi (2024) reduce a transaction-level decision to a model-generated risk score, and Oluwo, Dada and Isiekwu (2024) describe machine learning applied to insurance underwriting, a judgement-heavy decision of long standing. Mbonu, Aliliele, Iwuanyanwu and Uzoka (2022) make a comparable move at one remove, treating a defined control-testing process as itself a candidate for AI-enabled automation, while Akomolafe, Agu and Bello (2023a) treat the prior question of where assurance effort should be directed as a decision that can be modelled rather than negotiated.

Financial control functions approach this apparatus from the direction of risk. Ominyi and Anichukwueze (2023) treat model risk management as the practice through which accountability for an algorithmic decision is operationalised, and Olaogun, Adesuyi, Akomolafe, Ndukwe and Sakyi (2023) describe a changed regulatory obligation flowing through into redesign of a payment product as a drift-sensitivity event. Kumuyi, Uzoka, Akeju and Ozowara (2024b) treat an automated identity-verification decision as one that must satisfy privacy constraints by design; Sanni, Atima and Attah (2022) argue that competing modelling methods carry biases affecting the conclusions drawn from them; and Komi (2023) argues that a model accurate in aggregate may distribute error unevenly across groups, so an accuracy figure is uninformative about *whose* error it describes.

What recurs is a score, a flag or a ranking whose consequence is left unstated: the confidence at which an alert is worked, the confidence at which it is closed unexamined, and the relative cost of the two error directions belong to operating practice rather than to any written requirement. That is Gate 4 exactly. A detection model produces a prediction; the decision is the routing rule consuming it (Bertsimas & Kallus, 2020), and a domain specifying the former and not the latter produces requirements failing the error-tolerance specification characteristic of Section 5.7 and, where the flag is worked under time pressure, the escalation specification characteristic with it.

3.2. Specification in Regulated and Compliance-Bound Processes

The decision here is a conformance decision: whether a case, transaction or filing satisfies an obligation drafted by a body outside the organisation. Where a process is compliance-bound the criteria originate outside the organisation and were not drafted with automation in view. Aliliele, Mbonu and Iwuanyanwu (2024b) treat a regulatory regime as constraining the architecture on which analytics runs, and Atakpa and Abolaji (2022) derive a data architecture from privacy obligations rather than the reverse, the sequencing principle underlying Gate 3. Ominyi and Anichukwueze (2022a) reconcile obligations across jurisdictions and Mbonu, Aliliele, Uzoka and Oluoha (2019) review comparative regulation generality in the sense of Table 2, while Ominyi (2021) embeds privacy requirements when a structure is designed rather than retrofitting them.

The wider applied record here is largely a literature of translation. Eyetsemitan, Ambali, Oyeleye and Fadayomi (2021) treat externally stated tax and regulatory requirements as systematically translatable into executable organisational workflows, the domain's name for what this paper calls specification. Ominyi and Anichukwueze (2020) describe an external clearance process as imposing a defined sequence of evidence and decision steps, and Ominyi and Anichukwueze (2021) report that compliance regimes differ across markets and impose differing operational obligations, the generality row of Table 2, arriving from outside rather than within. Ominyi and Anichukwueze (2022b) observe that disclosure obligations determine what an organisation must measure and report, a data grounding constraint in regulatory rather than architectural language. Where the obligation concerns automated systems, purpose-built governance structures are proposed for such systems in regulated industries (Ominyi & Anichukwueze, 2024), and readiness for a specific AI regulatory regime is treated as a distinct assessable question (Ominyi *et al*., 2024b). Eze, Akinleye and Anene (2024b) come closest to a behavioural specification: a decision model constructed to make an explicit trade-off between speed and evidence quality, which is a directional error profile written in a regulator's vocabulary rather than a data scientist.

The characteristic failure sits at Gate 1. Regulatory language carries the vagueness Endicott (2000) argues is sometimes functional and the ambiguity Massey *et al*. (2014) taxonomise; an organisation transcribing it into a decision model unexamined has resolved nothing and has silently committed to a reading it will be unable to defend. The gate's product, a decision-bearing problem statement with a contested-terms register, is the appropriate response, and what survives it is what the decidability characteristic requires to be named rather than assumed.

3.3. Data Governance, Information Security and Access-Decision Workflows

The decision here is an access or classification decision: whether a principal may act on a resource, and what class of protection a data item attracts. It is high-volume, low-visibility, and consequential mainly in its failures. Data governance supplies the clearest instance of a decision whose inputs are governance artefacts. Aliliele, Mbonu and Iwuanyanwu (2024a) classify enterprise data by sensitivity and trace each class to the obligation governing it, the data grounding record with an external provenance link added, and Mbonu, Aliliele, Iwuanyanwu and Oluoha (2018) model legal and ethical exposures explicitly. Ladapo, Jooda, Dosunmu and Abolaji (2023b) treat privacy requirements as translating into access-control policy and Badmus, Jooda and Anunagba (2024) express them as design requirements on an access model. Adebayo, Adegbite and Ahmed (2022) model the attack surfaces such systems introduce, and Annan (2024) notes that demands for explanation can conflict with proprietary protection.

The surrounding record documents the machinery such a decision depends on. Mbonu, Aliliele, Iwuanyanwu and Uzoka (2020) treat access control as an enterprise-wide design problem once systems span multiple environments, and Ladapo, Jooda, Dosunmu and Abolaji (2019) describe centralised identity management as the mechanism through which enterprise access is administered, the register against which an automated access decision would be evaluated. Ogbole *et al*. (2021) set out a control architecture as part of a governance model rather than an implementation detail, and Okoruwa *et al*. (2020) take privacy requirements as a starting constraint on system engineering. Annan (2022) describes distinct governance models for personal data across jurisdictions, so the class an item falls into is jurisdiction-dependent rather than intrinsic. Aliliele, Mbonu and Iwuanyanwu (2023a) describe continuous automated monitoring as the means by which a control weakness in an enterprise platform is detected, and Okoruwa *et al*. (2024) observe that the risk environment an organisation governs is itself changing over time.

That last observation locates the characteristic failure. Gate 3 supplies the record of every input named, sourced, and accountable, but the binding difficulty is that the classification an input carries is not stable: an obligation is amended, a jurisdiction added, a threat class appears, and a rule correct at specification time becomes silently wrong without any change to the automation. That is the drift sensitivity characteristic of Section 5.7, whose evidencing artefact is a monitoring clause rather than a policy document. A data grounding record here must state not only who is accountable for each input's quality but under what external change its classification ceases to hold, and who would notice.

3.4. Enterprise Systems, Service Management and Operational Decision Support

The decision here is a routing decision inside an operational system: which queue, which owner, which service response, on the evidence the platform holds at that moment. The domain concerns how a decision reaches an operational system. Badmus, Dosunmu and Ozowara (2019a) treat multi-system integration as a design problem requiring explicit specification, and Nnabueze *et al*. (2021) achieve traceability across a multi-party process through a designed visibility framework. Human oversight recurs here as a design pattern rather than a policy requirement: Ladapo, Dosunmu, Jooda and Abolaji (2022) describe human-in-the-loop configurations as an established pattern, and Ladapo, Jooda, Dosunmu and Abolaji (2024) retain a defined review point inside generative automation of data preparation. Nwakamma, Ojukwu and Oyesiji (2024a) treat release gating as automated evaluation that is itself biased, and Nwakamma, Ojukwu and Oyesiji (2024b) name failure modes including the granting of more agency than intended.

The platform on which such a decision runs is not neutral with respect to what can be specified. Badmus, Jooda, Ozowara and Anunagba (2022b) report that recurring integration problems have named, reusable patterns, so an input's availability at the decision moment is partly a property of a pattern chosen elsewhere; Badmus, Dosunmu and Ozowara (2019b) govern platform management in a regulated setting through an explicit framework; and Okonkwo, Agbabiaka, Mayo and Okeke (2024f) automate operational workflows on a general-purpose service management platform rather than bespoke software, which fixes in advance what the automation can observe and emit. Ladapo, Jooda, Dosunmu and Abolaji (2023a) survey such a platform as the vehicle through which standardised service processes are enacted. Around it sits a decision-support layer, adoption being treated as a measurable indicator alongside delivery and portfolio performance (Edivri & Oteri, 2024) and rollouts as programmes with their own performance frameworks (Oteri & Edivri, 2024). When the component is learned rather than configured, Oyesiji, Nwakamma, and Ojukwu (2024) argue that the data used to adapt a model requires deliberate, context-aware curation. Gate 3 restated for a fine-tuning pipeline.

The characteristic failure is at Gate 5, and is one of evidence rather than design. A service management platform records that a ticket was closed; it does not record what the automation saw, which model version acted, at what confidence, or what the reviewer changed. Conformance to a rate cannot be reconstructed from such a record after the lost-rationale problem in operational form (Gotel & Finkelstein, 1994), which is why observability is specified in Section 5.7 as emission rather than reporting.

3.5. Analytics and Predictive Decision Systems in Operational Settings

The decision here is an intervention decision: given a stream of observations from equipment, infrastructure or an operating environment, decide whether to act now, act later, or do nothing. Kumuyi, Uzoka, Akeju and Ozowara (2024a) describe an operational management decision driven by models trained on continuously collected sensor data; Sunday, Omoegun, Essien and Oluokun (2020) treat the move from a reactive to a predictive operating regime as itself a transition to be managed rather than a change of tooling; and Adebayo, Adegbite and Ahmed (2023) apply model-based detection to operational data streams in a high-consequence setting. Komi (2021) observes that forecasting output feeds operational and market decisions with distributional consequences, and Komi and Adeniji (2023) describe an analytic capability tiered down to work under resource constraints; what can be specified depends on what the receiving organisation can operate, which is why Section 5.6 distinguishes acceptance from verification. Komi and Adamolekun (2021) come closest to this paper's concern,

treating interpretability as a requirement of a predictive model used for operational action rather than as a desirable property of the model in itself. Lawal and Oduleye (2021a) make the complementary move, representing a recurring management decision as an explicit decision model rather than an analytical output awaiting interpretation.

The characteristic failure is at Gate 2. A predictive stream is not a decision; it becomes one only when someone states the alternatives among which it selects, the criteria, and the point at which a prediction becomes an action. Where that step is skipped the organisation acquires a monitoring surface and retains the original decision, unchanged and undocumented, the orphan decision smell of Section 5.7 in operational dress. Komi and Adamolekun's (2021) framing suggest the corrective: an interpretability requirement becomes specifiable only once a recipient and a task exist for it; Miller's (2019) point, arriving from the plant floor, presupposes the decidability characteristic.

3.6. Procurement, Supplier Evaluation and Supply Chain Decisions

The decision here is a selection under multiple criteria: which supplier, which bid, which risk treatment, on criteria partly commercial, partly regulatory and partly held tacitly by the buyer. Ahiaeke Patrick, Okonkwo, Mayo and Okeke (2021) model a recurring supplier selection decision and drive it from data rather than unaided judgement; Bello, Akomolafe and Agu (2023) manage procurement risk on data rather than relationship judgement; and Tonoyan, Dada and Ayivi-Donkor (2024a) conduct vendor risk assessment through predictive models rather than static questionnaires. Akinleye and Adeyoyin (2021) propose process automation as a route to both efficiency and transparency in a transactional function, objectives separable and not always compatible. The domain is unusually explicit about trade-offs: Akin-Oluyomi, Atima and Akinleye (2023a) describe procurement criteria as encoding a deliberate trade-off between competing objectives, and Akin-Oluyomi, Atima and Akinleye (2023b) conduct supplier assessment in a regulated sector through a defined compliance framework, so part of the criterion set is fixed externally. Babatope, Akokodaripon, Akinleye and Okoruwa (2023) claim that how information is presented bears on decision effectiveness.

That last is a Gate 5 claim made inside a Gate 2 domain, and the juxtaposition is instructive. Criterion weighting here is elicited rather than derived, as the prioritisation literature would lead one to expect (Achimugu *et al*., 2014; Riegel & Doerr, 2015), so a decision model is the right artefact and its weights a recorded stakeholder settlement rather than a discovered fact. But if presentation bears on the decision, any human confirmation step inside an automated supplier evaluation is itself a designed artefact whose informational preconditions must be specified rather than inherited from whatever the tool emits; the requirement Section 5.6 imposes, and the absence the oversight placeholder smell detects.

3.7. Clinical, Health Service and Public-Service Decision Settings

The decisions here are allocation and eligibility decisions taken under statutory duty: who is seen, what is funded, what is flagged, what is reported to whom. Ilodigwe and Adesemoye (2021) treat data quality and interoperability as governance and architecture problems prerequisite to analytics Gate 3 as an organisational programme rather than a specification artefact, and Ilodigwe and Adesemoye (2024a) propose evidence generated from operational data as an input to organisational decision-making. Atakpa, Abetoh and Akeju (2024) describe AI-based detection deployed against transaction streams in a public-sector payer setting, where the population is a duty rather than a market. Adelanwa, Basnet and Anene (2024) treat outcome measurement in large service organisations as a distinct modelling problem, which matters because a Gate 5 acceptance criterion is an outcome statement, not a model metric. Eze, Anene and Akinleye (2022a) supply the clearest instance of escalation design: an early warning signal linked to a defined and pre-agreed response process, which is what a confidence threshold with a stated consequence looks like when drafted by a regulator rather than an engineer. Afrihyia, Akinse and Ojukwu (2024) report that executive oversight and accountability arrangements for AI systems differ by regulatory context, so the actor and forum of Gate 5 are not constant across jurisdictions.

Ilodigwe and Adesemoye (2019b) supply the caution governing the whole domain: a technically sound innovation fails to deliver benefit where the surrounding delivery capability is absent. The characteristic failure is therefore at Gate 5, and is one of capacity rather than design: the under-resourced reviewer whose failure Green (2022) predicts, performing work the automation generates rather than removes (Grønsund & Aanestad, 2020). An oversight specification stating what the reviewer is shown but not how much time the process allows has specified half an arrangement, and the half it omits is the half that fails.

3.8. Customer-Facing and Marketing Decision Automation

The decision here is a treatment decision at population scale: which customer receives which contact, offer or content, and when. Ogundairo and Ayivi-Donkor (2023a) manage a customer lifecycle against model-generated predictions, and Atima, Sanni and Attah (2022) apply predictive models to a routine targeting decision inside a regulated service business, so even a commercial decision has criteria partly fixed from outside. Wedraogo and Sanni (2024) treat forecast uncertainty as a property to be modelled rather than ignored, the nearest this corpus comes to the confidence-handling requirement of Gate 4. Sanni and Wedraogo (2024) locate losses across a multi-stage process analytically rather than by intuition, and Sanni, Iwuanyanwu, Essien and Wedraogo (2024) make workflow steps verifiable by encoding them in an executable contract and an observability mechanism reached commercially rather than through governance.

The domain also supplies the sharpest statement of the adoption problem. Ogundairo and Ayivi-Donkor (2024a) argue that an AI capability can be technically delivered and still fail on adoption, with causes that are not technical; and Ogundairo and Ayivi-Donkor (2024b) that the arrival of autonomous agents changes how a business function must be architected, a claim about the unit of specification rather than the capability of a component.

The characteristic failure is at Gate 1. The stated problem more conversion, less attrition is a consequence rather than a decision, and the decision consuming the effort is a per-case treatment choice whose error profile nobody has written down in either direction. Where the treatment is delivered by an agentic component the residue grows rather than shrinks, since the failure modes named by Nwakamma *et al*. (2024b),

including the granting of more agency than intended, are behaviours under uncertainty rather than events with triggers. Gate 1's separation of the presented problem from the decision consuming the effort is the whole of the remedy available before decomposition begins.

3.9. Programme and Project Decision Governance

The decision here is a gate decision: whether a programme proceeds, at what scope, under whose accountability, and on what evidence. Amayo, Owulade and Isi (2023a) treat project governance arrangements as a variable organisations deliberately tune, and Amayo, Owulade and Isi (2023b) treat risk management and compliance as paired delivery controls on a regulated programme. Ike *et al*. (2024a) make risk allocation between contracting parties explicit and quantified, and Aminu-Ibrahim, Ogbete and Ambali (2024) model governance and accountability arrangements between partner organisations the actor-and-forum question of Wieringa's (2020) account, arriving from the delivery side rather than the audit side. Adelanwa, Basnet and Anene (2023a) manage performance across a delivery lifecycle rather than at a single checkpoint, the objection Section 7.2 raises against converting the five gates into approval milestones. Eyetsemitan, Ambali, Oyeleye and Fadayomi (2020) require an explicit governance alignment mechanism where business processes are coordinated across multiple stakeholders in a highly regulated environment, and Adeyelu and Dagodzo (2024) model audit outcomes as a function of assessed organisational maturity readiness as a property of the organisation rather than of the artefact under review.

The characteristic failure is at Gate 5, in a form the other domains do not display: not an absent reviewer but an absent criterion. A programme gate asking whether the work is on track has asked a question no evidence answers; one asking whether the specified error profile has been verified over the specified population and window has asked something a third party could evaluate. The distinction Section 5.6 draws between verification and acceptance is exactly this, and it is why acceptance criteria are stated as observables. It is also where the six proposed characteristics are most usable as a checklist rather than as theory: a programme board can ask whether each in-scope decision has a decision model, a data grounding record, a directional error profile, a defined below-threshold behaviour, an emitted evidence trail and a stated drift condition, without adjudicating the underlying analysis.

3.10. What the Survey Does and Does Not Show

Across these domains the specification gap appears not as a property of any domain's technology but of the relation between how a business states what it wants and what a machine requires. The same absences recur: no alternatives named, no inputs established at the decision moment, no direction of error, no behaviour under low confidence, no emitted evidence, no stated condition of expiry in functions sharing neither vocabulary nor tooling. That recurrence is the only thing the survey is offered to illustrate. It is not evidence that the gates are the right gates, that their ordering is correct, or that traversing them would improve any outcome; Section 7.5 returns to the limits of the generality claim and Section 8 to the standing of the literature carrying it.

4. Method

This paper is a conceptual contribution. The framework was developed by structured narrative synthesis of the published literature identified in Section 2, read against the author's professional experience of business analysis practice. No primary data were collected, and the framework has not been empirically validated. Section 8 sets out the resulting limitations, and Section 7.6 proposes a validation agenda.

The synthesis proceeded in three passes: identifying the requirements-engineering constructs that survive translation into a business-process setting; identifying, from the automation, decision-modelling, data-quality and AI-governance literatures, the constructs such a specification must carry but that the requirements literature does not supply; and arranging these into a staged pipeline ordered by the principle that each ambiguity should be resolved where the information needed to resolve it first becomes available at least cost. Section 5.9 states eight propositions suitable for test; no results are reported and no validation is claimed.

The evidence base is of two kinds, kept apart. Section 2 is peer-reviewed research on which the constructs rest and from which every quantitative statement here is taken; Section 3 is applied and practitioner writing, much of it in venues without an established peer-review record, which now forms a substantial share of the reference list and is used only to show that the problem recurs across domains. Nothing in Section 5 or Section 7 depends on it.

5. The Ambiguity-To-Specification (A2s) Pipeline

5.1. Design Rationale

The pipeline is organised by one ordering principle: resolve each ambiguity at the gate where the information needed to resolve it first becomes available, and no earlier. Resolving too early is expensive and often wrong: an analyst who pins down a term in a framing workshop, before the decision has been decomposed and the data inspected, will frequently pin it to the wrong meaning and defend that meaning downstream, undetected ambiguity in the sense of Berry and Kamsties (2004), in the damaging form Ferrari, Spoletini and Gnesi (2016) call incorrect disambiguation. Resolving too late is also expensive, because design decisions will already assume a resolution.

A second principle follows: some ambiguity is irreducible at specification time and must be *specified* rather than resolved. If eligibility genuinely depends on a caseworker's assessment of whether a document is "consistent with the applicant's account", and no decomposition renders that mechanical, the correct specification records the residual judgement, the tolerated error profile and the escalation path rather than manufacturing false precision. Each gate therefore produces both resolved items and a register of residual ambiguity carried forward. Figure 1, rendered as Table 1, presents the pipeline as five gates with feedback paths to their predecessors and that register running across all of them. The gates are not project phases but an ordering of *analytical dependency*.

5.2. Gate 1 Problem Framing

The input is a business complaint expressed as a condition, symptom or wish. The distinguishing move is separating the *presented problem* from the *decision that actually consumes the effort*: a backlog is not a problem an automation can be built against but a consequence of some distribution of work over some set of decisions. The analyst refines the stated condition into goals with assigned responsibility (van Lamsweerde, 2009), establishes which stakeholders hold which partial description (Nuseibeh & Easterbrook, 2000),

and applies the stakeholder analysis and root-cause investigation of business analysis practice (International Institute of Business Analysis, 2015; Paul *et al*., 2020), recording following Pohl's (2010) separation of content from agreement not only what is claimed but who agrees. Gate 1 resolves pragmatic ambiguity of intent and generality, where one stated problem covers situations needing different treatment; it is the wrong place for lexical or domain ambiguity, for which evidence has not been assembled.

Two results give the gate its discipline. Because the presentation form of a requirement anchors the design thinking that follows (Mohanani *et al*., 2014), the artefact here is a register of contested terms rather than a set of draft definitions compiled while the analyst is still ignorant of the domain, which is how Berry's (1995) argument that ignorance is an asset should be read.

5.3. Gate 2 Decision Decomposition

Decision points are first *located* precisely within the process, which requires a process model of adequate fidelity (Object Management Group, 2014). Decision logic is then *separated from control flow*, following Hasić, De Smedt and Vanthienen (2018): the logic belongs in a decision model, the process model showing only where the decision is invoked and what happens to its outcome. Batoulis *et al*. (2015) observe that real-world process models routinely bury detailed decision logic inside control flow as cascades of gateways, and present a semi-automatic method to extract it; until it is lifted out, the logic is effectively unspecifiable. The decision is expressed in DMN (Object Management Group, 2024) over a controlled business vocabulary (Object Management Group, 2019) so each term carries one agreed meaning. Where work is genuinely non-sequential, CMMN (Object Management Group, 2016) is the appropriate notation, and its use is itself a finding: such a decision is a poor candidate for the later gates. Di Ciccio, Marrella and Russo (2015) characterise knowledge-intensive processes as knowledge- and data-centric and as requiring flexibility at design and run time, conditions under which a control-flow-first model misdescribes the work; recognising this early is cheaper than discovering at Gate 4 that no stable population exists over which a rate could be defined.

Compliance criteria are lifted out on the same principle. Hashmi, Governatori, Lam and Wynn (2018), analysing 79 papers published between 2000 and 2015, report compliance approaches distributed across design-time (28%), run-time (32%) and auditing (10%) categories, and Ly, Maggi, Montali, Rinderle-Ma and van der Aalst (2015) supply a framework of monitoring functionalities for run-time approaches. An obligation checkable only after the fact is a different specification object from one evaluable as the decision is taken.

The third task is to *measure determinacy* rather than assert it, determinacy being a property of the process as executed rather than as documented. Process mining (van der Aalst, 2016) provides the evidential route from event data to the process as it actually runs; robotic process mining extends this to the user-interaction level, analysing data collected during execution of user-driven tasks to identify and assess candidate routines for automation, with frequency and determinism as selection criteria (Dumas *et al*., 2022; Leno *et al*., 2021). Bosco *et al*. (2019) discover automatable routines from user interaction logs, and Wanner *et al*. (2019) argue for a quantifiable basis for process selection in place of workshop intuition. The claim here is narrow but consequential: an analyst who says a decision is "mostly straightforward" has produced an opinion, whereas one who reports the observed outcome distribution for recurring input patterns has produced evidence. The gate resolves syntactic ambiguity in rule statements, scope ambiguity about which cases a rule covers, and much domain ambiguity.

Three qualifications keep that claim honest. An unsegmented interface log has no case notion, so routines must be discovered before assessment (Leno *et al*., 2020); comparability is limited by the absence of an agreed log schema (Abb & Rehse, 2024); and deviation from a prescribed model is measurable through alignment, fitness and precision, so determinacy is a property of an extraction (Carmona *et al*., 2018). Predictability is separable: Teinemaa, Dumas, La Rosa and Maggi (2019) benchmark eleven predictive monitoring methods over 24 tasks from nine event logs. Deployed bots also emit logs on which process mining can be turned back (Egger *et al*., 2020).

Two further points of grounding. Rules elicited separately from operations, compliance and quality functions conflict, and conflict is a property of no single sentence; treating the decision table as a partial formalisation exposes contradictions as well as ambiguities (Gervasi & Zowghi, 2005). And where a decision rests on several criteria, their relative weight is a prioritisation question the literature does not settle (Achimugu *et al*., 2014).

5.4. Gate 3 Data Grounding

Gate 3 establishes what data is available *at the moment the decision is taken*, not what exists somewhere in the estate, nor what could be reconstructed afterward. The position taken here is that this gate is where most automation initiatives silently fail: the decision model is sound, the rules are agreed, and the field the rule depends on is populated three days late, or as free text, or truncated in the interface the automation will read.

Three assessments are required. *Availability at the moment of decision* is a temporal and architectural question, not a data-modelling one. *Quality along named dimensions* draws on Wang and Strong's (1996) four-category, fifteen-dimension model intrinsic, contextual, representational and accessibility quality whose claim that data quality is fitness for use rather than accuracy alone is what a specification needs, since an input can be accurate and still unusable for the decision at hand; Pipino, Lee and Wang (2002) supply assessment apparatus, Lee *et al*. (2002) a methodology for information quality assessment, and Nelson, Todd and Wixom (2005) a decomposition into accuracy, completeness, currency and format. *Runtime accessibility* asks whether the automation can obtain the input, at the required latency, through a supported interface.

Three features of that tradition make it usable as specification apparatus rather than vocabulary: it is methodologically mature, with a comparative description of assessment methodologies spanning phases, strategies, dimensions and target systems (Batini, Cappiello, Francalanci & Maurino, 2009); its dimensions are derived rather than conventional, resting on an ontological account of how a system represents a real-world system (Wand & Wang, 1996); and fitness is relative to a task (Strong, Lee & Wang, 1997).

Two cautions follow. Metrics must satisfy stated formal properties to support economically grounded decisions (Heinrich, Hristova, Klier, Schiller & Szubartowicz, 2017),

and a survey identifying 667 data quality tools and evaluating 13 found profiling covered far better than ongoing monitoring (Ehrlinger & Wöß, 2022), so a drift-sensitivity clause requires construction rather than configuration.

Data grounding is also a governance question. Khatri and Brown (2010) treat data governance as decision domains, principles, quality, metadata, access, and lifecycle with explicit accountabilities, and Abraham, Schneider and vom Brocke (2019) synthesise the field into mechanisms, scope and domains. A record naming an input without naming who is accountable for its quality is how an input silently degrades.

A last consideration concerns the reviewer. Disclosing data quality metadata changes decision strategies and outcomes (Chengalur-Smith, Ballou & Pazer, 1999), but whether it is used depends on experience and time pressure (Fisher, Chengalur-Smith & Ballou, 2003), so the Gate 5 requirement must state what the reviewer sees and how long they have.

That this gate carries disproportionate weight is supported, though not established, by the software-engineering-for-machine-learning literature: data-related issues are the most recurrent challenges across 248 studies (Martínez-Fernández *et al*., 2022); training-data handling is a characteristic point of failure (Nahar *et al*., 2022); and data dependencies, undeclared consumers and hidden feedback loops make an input adequate when assessed inadequate later, through a change made by a party who does not know the automation exists (Amershi *et al*., 2019; Sculley *et al*., 2015). Stated positively, collection, labelling and validation are the dominant quality levers in such systems (Whang, Roh, Song & Lee, 2023).

Two findings bear on how a tolerance is written. Northcutt, Athalye and Mueller (2021), in a preprint, report that labelling errors are pervasive even in benchmark test sets and can invert model rankings, so a verification design comparing candidates on a labelled sample measures the labels as much as the components; and where a component is trained, dataset accountability is a process obligation (Hutchinson *et al*., 2021).

Following Rahimi *et al*. (2019), the data grounding record has the same standing as the decision model rather than being an appendix to it; where a component will be trained, the data-management desiderata of Ashmore *et al*. (2021) attach here, ISO/IEC 5259-2:2024 supplies measures in which a tolerance can be stated rather than asserted (International Organization for Standardization & International Electrotechnical Commission, 2024), and the model card of Mitchell *et al*. (2019) becomes the artefact against which a later change of population is checked. None of this establishes that Gate 3 failures dominate abandonment, which is Proposition P5, and only tests that data problems are common, structural rather than incidental, and discovered late unless looked for early.

5.5. Gate 4 Behavioural Specification

Because the components in question are right at a rate rather than simply right (Ng *et al*., 2021), the specification must state four things conventional requirements do not. The first is *which errors matter and in which direction*: error costs in casework are almost never symmetric, and a single accuracy target says nothing useful about either direction, so the analyst elicits the asymmetry from those who bear the consequences and records it directionally. The second is *thresholds and confidence handling*: at what confidence the outcome is acted upon, at what confidence it becomes a recommendation, and at what confidence it is suppressed. The third is *fallback behaviour below threshold and for out-of-scope input*; a requirement silent on what happens when the system is unsure has left the most operationally consequential case to implementation default. The fourth is *how the requirement will be verified*, as ISO/IEC/IEEE 29148 (International Organization for Standardization, 2018) requires of any well-formed requirement, but defined over a sample and a period, since a rate cannot be verified by a single test case; the Volere fit-criterion discipline (Robertson *et al*., 2024) transfers well provided the criterion is expressed as a rate over a defined population.

The gate resolves modal ambiguity, the difference between may, should, and shall, and the ambiguity latent in evaluative terms such as "accurate", "reliable," and "acceptable", resolvable only as a rate with a direction. It exits when the verification design could be executed by someone who did not write the requirement.

Two commitments here are easily left implicit. Since prediction and decision are distinct (Bertsimas & Kallus, 2020), stating predictive performance has not stated the decision rule that consumes it: threshold, routing and fallback are the decision. And since fairness admits mutually incompatible definitions (Mehrabi *et al*., 2021), the specification must name the subgroups over which the rate is reported separately, because a rate never disaggregated cannot be found uneven.

Three strands shape this gate. The satisficing logic of the NFR Framework (Chung *et al*., 2000) implies a recorded trade-off rather than a set of independent targets; if most requirements labelled non-functional in fact describe behaviour (Eckhardt *et al*., 2016; Glinz, 2007), the operative distinction is between requirements true of an execution and requirements true at a rate over a population; and since classical techniques do not transfer intact to machine learning and practitioners struggle to measure such attributes at all (Habibullah *et al*., 2023; Horkoff, 2019), the response taken here is to require that the trade-off be written down by the party bearing the consequence, in a form a third party could test.

5.6. Gate 5 Oversight and Acceptance

Gate 5 designs the human role, the accountability path, the evidence trails and the acceptance criteria. The risk-management frameworks supply the scaffolding a specification must fit: the NIST AI Risk Management Framework, organised around Govern, Map, Measure and Manage (National Institute of Standards and Technology, 2023), ISO/IEC 42001 as a certifiable management-system standard (International Organization for Standardization, 2023a), and ISO/IEC 23894 as risk-management guidance (International Organization for Standardization, 2023b). Regulation (EU) 2024/1689 (European Parliament and Council, 2024), adopted 13 June 2024, published in the Official Journal on 12 July 2024 and entered into force on 1 August 2024, is risk-tiered in structure; the tier assigned to a use determines the weight of obligation attaching to it, and the framework engages with that structure and principle rather than with any compliance calendar. The ethics guidelines of the High-Level Expert Group on Artificial Intelligence (2019) supply the vocabulary of trustworthiness. For allocation, Shneiderman's (2020) three-layer model distributes responsibility across team, organisation and industry; Mäntymäki *et al*. (2022) define organisational AI

governance as a distinct object; Schneider *et al*. (2023) decompose governance into data, model and system layers along who/what/how dimensions; Raji *et al*.'s (2020) SMACTR framework supplies the internal audit apparatus; and Wieringa (2020), synthesising 242 articles (93 core) from 2008–2018, gives five dimensions of algorithmic accountability; actor, forum, relationship, content and criteria, and consequences.

The critical move at this gate is negative. Green (2022) argues that policies requiring human oversight of government algorithms often fail as safeguards and can serve to legitimise flawed systems: the human is present, under-resourced, insufficiently informed and disinclined to override. It follows that specifying a human in the loop is not the same as specifying effective oversight. A requirement stating that "a caseworker reviews the system's output before issue" is unfalsifiable; it is satisfied by a click. An oversight requirement is well formed only if it states what the reviewer sees, what they are asked to judge, how much time the process allows, what proportion of outputs they are expected to change, and how the override rate is monitored; a specified override rate is not a target but an observable whose departure from expectation signals that the arrangement is failing. The gate thereby resolves ambiguity of responsibility, invisible earlier because it concerns relations between actors rather than the content of the decision.

Section 2.14 turns that critique into design obligations. The requirement must state the reviewer's informational preconditions, since an explanation is adequate relative to a recipient and a task (Miller, 2019); it must state the reviewer's time; and the override rate must be a two-sided observable, because the direction of departure cannot be assumed (Burton *et al*., 2020; Dietvorst *et al*., 2015; Logg *et al*., 2019) and collaboration can itself degrade independent judgement (Fügener *et al*., 2021).

The informational precondition deserves care, because the default is to supply whatever the stack emits, an artefact built for engineers debugging a model (Bhatt *et al*., 2020), and because a requirement not selecting within the family of explanation methods has selected nothing (Guidotti *et al*., 2018). Explanation can also conflict with proprietary protection (Annan, 2024).

The design must describe the work oversight creates, not the work it removes. Deploying an algorithm generates new human activity (Grønsund & Aanestad, 2020), and an operating model that has not budgeted for it produces the under-resourced reviewer whose failure Green (2022) predicts. Public-sector practitioners report accountability needs that are organisational and procedural (Veale *et al*., 2018); and since the arrangement encodes a settlement between developers' and experts' claims (van den Broek *et al*., 2021), recording who conceded what makes it auditable.

The evidence trail is the gate's second product, and where the artefacts of Section 2.15 attach: a model card (Mitchell *et al*., 2019), a supplier's declaration (Arnold *et al*., 2019) and a datasheet (Gebru *et al*., 2021) support a claim that a component is used inside its specified envelope, while continuous operation adds MLOps monitoring (Kreuzberger *et al*., 2023) and the failure patterns catalogued by Paleyes *et al*. (2022). The trail must be emitted rather than reconstructed, since reconstruction is where the lost-rationale problem of Gotel and Finkelstein (1994) becomes unrecoverable; what it can carry is bounded (Doshi-Velez & Kim, 2017), and where the stakes justify it an interpretable-model constraint may be written into the specification (Rudin, 2019). Acceptance is then distinct from verification: verification asks whether the specified behaviour occurs at the specified rate, acceptance whether the arrangement as a whole is fit to be operated by the receiving organisation, a step argued to warrant structured validation rather than informal sign-off even in small-business deployments (Eyetsemitan *et al*., 2023b), and one that here includes a human role no component test can assess.

Table 1: The five gates of the Ambiguity-to-Specification (A2S) pipeline. (Rendered content of Figure 1: five gates with feedback paths to their predecessors and a residual-ambiguity register carried across all gates.)

Gate	Input state	Analytical work	Artefact	Exit criterion
1. Problem Framing	Business complaint as condition, symptom or wish	Separate presented problem from the decision consuming the effort; refine goals and assign responsibility; identify stakeholders and contested terms	Decision-bearing problem statement with contested-terms register	A decision point named, with an owner and an outcome expressible against a baseline
2. Decision Decomposition	Decision-bearing problem statement	Locate decision points; lift decision logic out of control flow; ground vocabulary; measure determinacy from event and interaction evidence	Decision model plus determinacy assessment with stated evidential basis	Every in-scope decision modelled; determinacy evidenced; undecomposed criteria listed
3. Data Grounding	Decision model with named inputs	Establish availability at the decision moment; assess quality on named dimensions; test runtime accessibility; record provenance and readiness	Data grounding record per input, with sensitivity to degradation	No input unavailable, unassessed or unreachable; failures returned to Gate 2 or recorded as limitations
4. Behavioural Specification	Decision model with grounded inputs	State directional error tolerance; set thresholds and their consequences; define fallback and out-of-scope behaviour; design verification over a population and window	Behavioural specification per decision point, with verification design	Directional error profile, defined below-threshold behaviour and executable verification design for every decision
5. Oversight and Acceptance	Behavioural specification	Design the human role and its informational preconditions; assign the accountability path; define the evidence trail; state acceptance as observables	Oversight and acceptance specification	Named actor, forum, sufficient evidence trail and third-party-evaluable acceptance criteria

Table 2: Ambiguity types mapped to the gate of cheapest resolution and to the automation consequence of leaving them unresolved.

Ambiguity type	Characterisation	Gate of cheapest resolution, and why	Consequence if unresolved	Terminal state
Lexical	A term carries more than one sense ("case", "claim", "file")	Gate 2: decision table and controlled vocabulary force each term into a single relation	Rules operate over a different population than intended; silent under- or over-coverage	Resolve
Syntactic	Scope of a modifier, conjunction or negation is structurally unclear	Gate 2: decision tables have no syntax in which the ambiguity survives	Rule fires on conditions no stakeholder intended; appears later as inexplicable exceptions	Resolve
Semantic	Structurally clear but admitting more than one reading of what is asserted	Gate 2, residue to Gate 4: decomposition settles most; what remains is evaluative and becomes a rate	Two implementers build two different systems, each defensible against the text	Resolve, then specify residue
Pragmatic / intent	What the requester wants differs from what the statement says	Gate 1: only at framing are stakeholder and stake jointly available	Correct system, wrong problem; the backlog persists after successful delivery	Resolve
Referential	"It", "the record", "the previous decision" unclear which entity is meant	Gate 3: only when inputs are traced to sources does the referent become determinate	Automation reads the wrong field, or the wrong version of the right field	Resolve
Vagueness	The predicate admits borderline cases ("recent", "complete", "sufficient")	Gate 3 for input vagueness, Gate 4 for outcome vagueness: the first resolves against fields and tolerances, the second only as a rate	Threshold set by implementer default; behaviour drifts with no one having decided it should	Resolve where operational; specify where evaluative
Generality	One statement covers several situations requiring different treatment	Gate 1, refined at Gate 2: framing exposes the situations, decomposition separates them	A single automated path applied to heterogeneous cases; error concentrates in the minority case	Resolve
Domain underdetermination	Linguistically clear but underdetermined relative to the application domain	Gate 2 with Gate 3 evidence: model and data record together fix the domain meaning	Expert and implementer agree on the words and disagree on the cases	Resolve
Modal	"May", "should" and "shall" used interchangeably	Gate 4: only behavioural specification separates obligation, permission and expectation	Optional behaviour implemented as mandatory or omitted; acceptance disputes	Resolve
Evaluative	"Accurate", "reliable", "acceptable" applied to an outcome	Gate 4: only here can the term become a rate with a direction over a population	Requirement is unverifiable; acceptance becomes negotiation rather than test	**Specify** as directional rate
Confidence / uncertainty	What the system does when unsure is not addressed at all	Gate 4: threshold and consequence are behavioural properties	The most consequential case is decided by implementation default	**Specify**
Responsibility	Who decides, who answers for it, on what criteria	Gate 5: only oversight design brings actor, forum and criteria into one view	Oversight exists on paper and is a click in practice; no defensible account after an adverse outcome	**Specify**
Irreducible judgement	A criterion resists decomposition because it genuinely requires human assessment	Gate 4 for the error profile, Gate 5 for the human role: nothing earlier can remove it	False precision is manufactured and the automation is built against a fiction	**Specify**

The table is diagnostic. An analyst meeting an ambiguity reads across to establish whether it is being addressed too early, the common failure, in which a framing workshop invents a definition for a term the data would have settled, or too late, and reads the final column to establish whether the correct terminal state is a resolution or a specified residue.

5.7. AI-Readiness Quality Criteria: An Extension of ISO/IEC/IEEE 29148

ISO/IEC/IEEE 29148 (International Organization for Standardization, 2018) characterises a well-formed individual requirement as necessary, appropriate, unambiguous, complete, singular, feasible, verifiable, correct and conforming, with further characteristics defined over a requirement set. These are the substrate of the requirements-smells approach of Femmer *et al*. (2017), which operationalises them as lightweight, automatable indicators of likely defects: a subjective phrase, a comparative without a referent, a loophole detectable cheaply and at volume. They are necessary but not sufficient for intelligent process automation. Six additional characteristics are proposed in Table 3; they are proposed, not derived, having been assembled analytically from the gate structure, and Section 8 records this as a limitation.

The proposal is deliberately conservative. Each characteristic extends an existing construct rather than adding a new dimension of quality, and each inherits the relational form that the harmonised theory of Frattini *et al*. (2023) and the quality-in-use position of Femmer and Vogelsang (2019) give to requirements quality: it is a property determining whether a downstream activity can proceed, not a virtue of the sentence. Where the attributes mapped by Montgomery *et al*. (2022) concern properties of requirement text, these concern the relation between a requirement and the operational conditions under which it will be evaluated.

Positioning each characteristic shows how modest the extension is. Decidability extends singularity and completeness on content rather than form: template and

glossary conformance are automatically checkable (Arora *et al*., 2015, 2017), yet a sentence can pass both while naming no alternatives. Data-groundedness extends feasibility, warranted by the derived account of quality dimensions (Wand & Wang, 1996) and by fitness being relative to a use (Strong *et al*., 1997).

Error-tolerance specification extends unambiguity and verifiability, and its closest relative is the non-functional requirements tradition, which is why it matters, since that tradition lacks an agreed taxonomy (Mairiza *et al*., 2010) and such requirements are often settled by whoever builds rather than whoever bears the consequence (Ameller *et al*., 2012); the rate must also be a metric whose aggregation behaviour is defined (Heinrich *et al*., 2017). Escalation specification extends completeness and has the oldest lineage of the six (Cockburn, 2001), its novelty being the trigger: the system's own confidence.

Observability extends verifiability and relocates traceability to run time, with warrant and caution alike: what is traced and why differs between low-end and high-end practice (Ramesh & Jarke, 2001), purposed traceability is a first-order challenge (Cleland-Huang *et al*., 2014), and automatic link recovery is doubtful at scale (Borg *et al*., 2014). Drift sensitivity time-indexes correctness and finds most support in the engineering literature, which catalogues the triggers that cause a system to be rebuilt (Steidl *et al*., 2023).

Table 3: Proposed AI-readiness quality characteristics extending ISO/IEC/IEEE 29148.

Characteristic	Definition	Test question	Artefact evidencing it	29148 characteristics extended
Decidability	The requirement names the decision, the alternatives among which it selects, and the inputs on which selection depends	Does it name a decision, its alternatives and its inputs or only a behaviour?	Decision model (Gate 2)	Singular; Complete
Data-groundedness	Every input is available, at the stated quality, at the moment the decision is taken and through a supported runtime interface	Is every named input obtainable at the decision moment, at the quality the decision needs?	Data grounding record (Gate 3)	Feasible
Error-tolerance specification	The acceptable error profile is stated, and stated asymmetrically wherever the costs of the two error directions differ	Which error is worse, by how much, and what rate of it is accepted?	Behavioural specification (Gate 4)	Unambiguous; Verifiable
Escalation specification	Behaviour under low confidence, out-of-scope input or unavailable input is defined	What happens when the system is unsure, or the input is missing or out of scope?	Behavioural specification (Gate 4)	Complete
Observability	Conformance can be evidenced from the system's own outputs, without reconstruction from external records	Could an auditor demonstrate conformance from what the system emits?	Evidence trail in the oversight specification (Gate 5)	Verifiable
Drift sensitivity	The condition under which the requirement ceases to hold is stated, with how that condition would be detected	Under what change in inputs, population or environment does this stop being satisfiable, and who would notice?	Sensitivity statement and monitoring clause (Gates 3 and 5)	Correct

Table 4 states the six characteristics against the constructs each extends, and names what must be added for the construct to bear on a probabilistic, decision-bearing automation; its purpose is to make the extension claim auditable rather than impressionistic.

Table 4: The six proposed AI-readiness characteristics positioned against existing requirements-quality constructs.

Proposed characteristic	Existing construct it extends	What the existing construct supplies	What the extension adds
Decidability	Singularity and completeness in ISO/IEC/IEEE 29148 (International Organization for Standardization, 2018); the constrained-syntax patterns of EARS (Mavin *et al*., 2009) and the sentence templates of Rupp and Pohl (2015)	One requirement, one behaviour, stated in a grammatical form that fixes trigger, actor and response	The requirement must name the alternatives among which a selection is made and the inputs on which selection depends; a well-formed template sentence containing an evaluative verb and no alternatives satisfies the construct and fails the extension
Data-groundedness	Feasibility in ISO/IEC/IEEE 29148; fitness for use in the data quality tradition (Wang & Strong, 1996) and its measures for machine learning (International Organization for Standardization & International Electrotechnical Commission, 2024)	That a requirement must be implementable, and that data quality is judged against a use rather than in the abstract	Fitness is assessed at the moment of decision and through the runtime interface the automation will use, and the dataset becomes a specification artefact in its own right (Rahimi *et al*., 2019) rather than an implementation input
Error-tolerance specification	Unambiguity and verifiability in ISO/IEC/IEEE 29148; satisficing and trade-off reasoning in the NFR Framework (Chung *et al*., 2000)	That evaluative terms must be made testable, and that quality goals are traded rather than maximised	The trade-off is stated *directionally* and as a rate over a defined population, and is elicited from the party bearing the consequence rather than inferred by the analyst; a single accuracy figure satisfies verifiability and fails the extension
Escalation specification	Completeness in ISO/IEC/IEEE 29148; the unwanted-behaviour pattern in EARS (Mavin *et al*., 2009)	That the specification must cover the cases that fall outside the nominal	The uncovered case is not an unwanted event but a state of the system's own confidence, so the requirement must define behaviour under

		path	low confidence and unavailable input, which no exception pattern over external events reaches
Observability	Verifiability in ISO/IEC/IEEE 29148; post-specification traceability and the recovery of rationale (Gotel & Finkelstein, 1994; Mäder & Egyed, 2015)	That conformance must be demonstrable, and that links between artefacts support later reasoning about them	Conformance must be evidenced from what the system emits at run time: inputs used, model version, confidence, reviewer action, because the rate being verified cannot be reconstructed after the fact from documents
Drift sensitivity	Correctness in ISO/IEC/IEEE 29148; the activity-based view that a defect is a defect only where it degrades a consuming activity (Femmer & Vogelsang, 2019)	That a requirement must correspond to a real need, assessed against the use made of it	Correctness is time-indexed: the requirement states the change in population, inputs, or environment under which it ceases to hold, and who would detect that change a property the static constructs do not express (Sculley *et al*., 2015)

Each characteristic generates a corresponding smell in the sense of Femmer *et al*. (2017). The orphan decision smell is an evaluative verb ("assess", "determine", "classify") with no named alternatives among which the assessment selects; the ungrounded input smell, information named in business terms that resolves to no field, source or timestamp; the symmetric accuracy smell, a single accuracy figure attached to a decision whose error costs are visibly asymmetric; the escalation void smell, automated behaviour specified with no statement of behaviour under low confidence; the oversight placeholder smell, an oversight clause reducing to a human "reviewing" or "approving" an output with no statement of what is shown, judged or expected to change (Green, 2022); and the timeless requirement smell, a requirement expressed as though its population were stationary. These are indicators, not verdicts.

5.8. Worked contrast

Table 5 shows three requirement statements as they might plausibly be elicited in the illustrative domain of Section 6, and as they read after traversal of the pipeline. They are hypothetical and are not drawn from any organisation.

Table 5: Requirement statements before and after traversal of the A2S pipeline.

#	As originally elicited	As rewritten through the pipeline	Defect closed
R1	"The system shall automatically assess eligibility for straightforward applications."	"For applications of type A received through the online channel, the system shall select one of three outcomes*: eligible, not eligible, or refer* by evaluating eligibility decision model DM-01 against the six inputs listed in data grounding record DGR-01, all of which must be present and current as at the intake timestamp. Applications not of type A, or missing any required input, are routed to *refer* without evaluation. 'Straightforward' is not an operative criterion; the operative criteria are case type and completeness of the six inputs."	Decidability and data-groundedness. The original names no alternatives, no inputs, and uses an evaluative term as though it were a criterion; the rewrite converts generality into an enumerated case type and vagueness into an input-completeness test.
R2	"The system shall verify that submitted documents are valid."	"For each document submitted against document class C1–C4, the system shall return a validity classification with a confidence value. At or above the agreed upper threshold the classification is applied automatically. Between the lower and upper thresholds, the document is queued for caseworker confirmation, with extracted fields and source image displayed side by side. Below the lower threshold the document is treated as unclassified and the case is routed to manual verification. The tolerated error profile is asymmetric: a document wrongly accepted as valid is materially more serious than one wrongly referred, and the acceptance test measures the two directions separately over a rolling monthly sample of closed cases."	Error-tolerance specification and escalation specification. The original states no direction of error, no confidence handling and no fallback; "valid" is an evaluative term resolvable only as a rate.
R3	"A caseworker will review the system's recommendation before the decision is issued."	"Before issue, a caseworker holding the relevant authorisation shall be shown the recommended outcome, the decision-model criteria that produced it, the values of the six inputs used and the confidence value, and shall record either agreement or a substantive reason for departure selected from a defined list. Review is allotted a defined minimum handling time, and the case cannot be issued until a reason code is recorded. The override rate is reported monthly by outcome type to the process owner; a sustained rate outside the expected band is treated as a signal that either the model or the oversight arrangement requires review, and triggers a documented reassessment."	Observability, and the unfalsifiability of nominal oversight. The original is satisfied by a click and generates no evidence; the rewrite specifies what the reviewer sees, what must be recorded, and the observable by which the oversight arrangement is itself monitored (Green, 2022).

5.9. Propositions for empirical evaluation

The framework is untested. The following propositions state it in a form suitable for evaluation by controlled comparison, multi-case field study or longitudinal observation.

P1. Specifications produced through the A2S pipeline exhibit fewer defects attributable to unresolved ambiguity at implementation than those produced by conventional practice.

P2. The gate at which an ambiguity is resolved predicts the cost of its resolution: ambiguities resolved at the gate identified in Table 2 consume less analyst and stakeholder effort than the same types resolved earlier or later.

P3. Registering residual ambiguity as specified rather than resolved reduces late-stage requirement change relative to practices that force premature resolution.
P4. Determinacy estimated from event or user-interaction evidence at Gate 2 predicts realised automation rate more accurately than determinacy estimated from expert judgement alone.
P5. Failures at Gate 3 account for a larger share of abandoned or descoped automation initiatives than failures attributable to model performance.
P6. Requirements satisfying the six AI-readiness characteristics are associated with fewer post-deployment behavioural disputes between business and delivery parties than requirements satisfying only the ISO/IEC/IEEE 29148 characteristics.
P7. The proposed AI-readiness smells can be detected with acceptable inter-rater agreement by analysts who did not write the requirement, and their presence is associated with independently assessed specification inadequacy.
P8. Oversight requirements specifying the reviewer's informational preconditions, recorded reason and monitored override rate are associated with higher observed override rates and more frequent detection of model degradation than those specifying human review alone.

6. Illustration

The following illustration is hypothetical. It uses the illustrative domain of this paper: a mid-sized services organization performing high-volume, document-heavy casework comprising application intake, eligibility assessment, document verification, decision issuance, and reporting to management and to an external regulator. No organisation is described and no data are reported; every quantity mentioned is a placeholder for a value that would be established empirically in a real engagement.
Gate 1. The presenting complaint is that "the assessment backlog is unmanageable and we need AI to fix it". Framing would be expected to establish that the backlog is a consequence, and that three decisions consume the effort: whether an application is complete enough to progress, whether documents are what they purport to be, and whether the applicant meets the eligibility criteria. The operations manager's concern is throughput, the quality lead's is consistency, and the regulatory liaison's is defensibility of the audit trail three partial descriptions of one situation (Nuseibeh & Easterbrook, 2000), and goal refinement (van Lamsweerde, 2009) separates throughput from consistency. The gate exits with *document verification* selected, its owner named, "complete", "verified" and "straightforward" on the contested-terms register, and the outcome stated as handling time per case at a stated quality level against a pre-implementation baseline.
Gate 2. Process modelling would be expected to reveal the pattern Batoulis *et al*. (2015) describe: verification criteria held nowhere as rules but distributed across a sequence of gateways and the tacit practice of experienced caseworkers. Lifting them out would decompose verification into classification, extraction and consistency with the application record; the first two would be amenable to data-driven components of the kind that superseded hand-built optical-character-recognition-plus-rules pipelines once text and layout began to be pre-trained jointly (Xu *et al*., 2020), the third largely rule-based with a residue of judgement. The separation principle (Hasić *et al*., 2018) leaves the process model showing only invocation and routing, and vocabulary grounding gives "verified" one meaning across the three sub-decisions. Determinacy is estimated from historical event data and user-interaction evidence, frequency and determinism serving as candidate-selection criteria (Dumas *et al*., 2022). Suppose classification proves highly determinate for four document classes and much less so for a fifth: that finding scopes the automation.
Gate 3. Data grounding would be expected to surface three findings of a kind that recur. Image quality might differ systematically between submission channels, a representational quality issue in Wang and Strong's (1996) terms rather than an accuracy issue bearing directly on extraction. The application record against which consistency is checked might be updated asynchronously, so that for some cases the record read at verification time would not be the record a caseworker would have seen: a currency problem in the decomposition of Nelson *et al*. (2005). A field on which a consistency rule depends might be captured as free text. Each would be recorded with the decision's sensitivity stated, provenance documented in the manner of Gebru *et al*. (2021), and readiness staged using Lawrence's (2017) preprint vocabulary. An asynchronous-update finding of this kind would have to be resolved, or the sub-decision returned to Gate 2, before behavioural specification began a sequencing consistent with the data-cascade finding of Sambasivan *et al*. (2021).
Gate 4. Behavioural specification for classification produces the text shown as R2 in Table 5. The asymmetry is elicited from those who bear the consequences: wrongly accepting an invalid document is treated as more serious than wrongly referring a valid one, because the first is discovered by the regulator and the second by the caseworker. Two thresholds are set rather than one, producing three regimes: automatic, confirm, and manual; verification is designed over a rolling monthly sample of closed cases with the two error directions measured separately, adapting the verifiability requirement of ISO/IEC/IEEE 29148 (International Organization for Standardization, 2018) to the fact that what is verified is a rate.
Gate 5. Oversight design produces the text shown as R3. The accountability path is stated in Wieringa's (2020) terms: the process owner is the actor, the internal quality forum and the external regulator are the forums, the criteria are the stated error profile and the documented decision model, and the consequence of an adverse finding is a defined reassessment. The evidence trail is designed so that inputs used, model version, confidence value and reviewer reason code are emitted by the system itself, satisfying observability. Governance responsibilities are allocated across data, model and system layers following Schneider *et al*. (2023), within the organisation's management-system obligations (International Organization for Standardization, 2023a) and risk-management practice (International Organization for Standardization, 2023b; National Institute of Standards and Technology, 2023). The override rate becomes an operational observable rather than a compliance formality.

7. Discussion

7.1. Implications for Business Analysis Practice

The analyst's artefact sets changes. A conventional requirements package: stakeholder list, process model, requirement statements with acceptance criteria, is insufficient for an intelligent automation, and the pipeline names four artefacts that must be added: a determinacy

assessment with an evidential basis, a data grounding record traceable input by input, a behavioural specification carrying a directional error profile and a verification design, and an oversight specification carrying an accountability path and an evidence trail. None appears in the standard business analysis repertoire (International Institute of Business Analysis, 2015; Paul *et al*., 2020).

Technique changes too. That group interaction techniques dominate practice (Palomares *et al*., 2021) is not a criticism: workshops are effective for Gates 1 and 5, which concern agreement in Pohl's (2010) sense. But Gates 2 and 3 concern content, and workshop consensus is a poor instrument for establishing content that event data can establish directly. The analyst therefore needs a working competence with process and interaction evidence (Dumas *et al*., 2022; Leno *et al*., 2021; van der Aalst, 2016) not conventionally part of the role.

The third implication is dispositional: the pipeline asks analysts to state a residual uncertainty rather than manufacture a rule nobody applies, and the instinct to deliver certainty works against AI-ready specification.

7.2. Implications for Method

The pipeline is a specification method, not a delivery lifecycle. Gates 1 and 2 have a natural affinity with up-front analysis because they establish scope; Gates 3, 4 and 5 have a natural affinity with iterative work because thresholds, fallbacks and oversight arrangements are refined against observed behaviour. That suggests a hybrid arrangement, but which lifecycle carries the method best, and under what conditions, is treated in the author's ongoing work rather than settled here. Two points hold regardless: the gates are dependency constraints rather than stage gates with formal sign-off, since a governance process that converts them into approval milestones reproduces the pathology the framework exists to avoid; and the residual-ambiguity register, which carries state between gates and iterations, is the natural attachment point for change control.

7.3. Relationship to Requirements Engineering for Ai

The claim made here is narrower than the framework's scope suggests. Three of its positions are adopted from the programme of Section 2.12 rather than argued for: that requirements for machine-learned components differ in kind from those for deterministic software (Vogelsang & Borg, 2019); that a dataset is a specification artefact with the same standing as a model specification (Rahimi *et al*., 2019); and that uncertainty and evolving context are first-class requirements concerns for such systems rather than implementation contingencies (Belani *et al*., 2019). None of the three is defended here, and if any of them fails the framework fails with it.

The differences lie in what is done with them. The unit differs: that literature specifies a component and its envelope, whereas the analyst specifies a decision inside a process, which is why Gate 2 exists, and why an undecomposable case is out of scope rather than a harder instance. The treatment of ambiguity differs: the mapping studies report that techniques are poorly adapted (Ahmad *et al*., 2023; Villamizar *et al*., 2021) without treating ambiguity as an object with a type, a cost and a point of resolution. And the terminal state differs: proposals such as the perspective-based technique of Villamizar *et al*. (2024) aim at more complete elicitation, whereas the position here is that some concerns must be carried forward as specified residue. That literature has nonetheless produced evaluated techniques where this paper offers an untested framework, so a reasonable programme would insert such techniques into the gates rather than treat them as competing.

Three points deserve more generous statement. The survey of where notations break down (Ahmad *et al*., 2021) does much of the diagnostic work Table 2 does, on a different axis; the finding that requirements, testing and maintenance are among the least well-served knowledge areas (Giray, 2021) argues for more work of this kind; and that programme has produced evaluated techniques where this paper offers an unevaluated ordering, so the agenda implied by Section 7.6 is insertion and test.

7.4. Implications for Tooling and For the Analyst's Role

Which of the pipeline's artefacts could be checked automatically? Three things look tractable. Textual smell detection of the kind Femmer *et al*. (2017) operationalise is cheap, and several proposed AI-readiness smells share that surface form: the orphan decision smell is an evaluative verb without enumerated alternatives, the symmetric accuracy smell a single numeric target attached to a decision described asymmetrically elsewhere both pattern-matching problems the field has addressed since the Automated Requirements Measurement indicators of Wilson *et al*. (1997). Conformance checking against a controlled vocabulary is tractable for the same reason, template conformance and glossary extraction both having been evaluated on industrial case material (Arora *et al*., 2015, 2017).

Two design principles from the tool literature transfer directly: separate flagging from explanation, leaving the judgement to a human (Kiyavitskaya *et al*., 2008), and prefer complete recall over a small decidable vocabulary (Tjong & Berry, 2013). Both suit the proposed smells, each defined by an absence. The nearest analogue is detection of missing rather than defective terminology (Luitel *et al*., 2024); and the record of such analysers (Lami *et al*., 2019) is a caution, since the survivors were embedded in existing quality work and adoption has lagged capability (Dalpiaz *et al*., 2018).

Three things are not tractable. Deciding which ambiguities matter is a judgement about readers rather than text, and the innocuous/nocuous distinction of Chantree *et al*. (2006) is where automation must hand over; detection remains harder and less precise than resolution (Ezzini *et al*., 2022); and tools do not yet beat attentive reading on recall (Dalpiaz *et al*., 2019), in a field where approximately 7% of studies were assessed industrially and only 17 of 130 tools remained available (Zhao *et al*., 2021). Nor should automated link recovery be assumed for the evidence trail (Borg *et al*., 2014), which is why observability is specified as emission. The design conclusion is Berry's, that for recall-critical work a transparent tool doing an identifiable sub-task beats an opaque one failing unpredictably at the whole (Berry *et al*., 2012). The implication is to build narrow checks producing candidate lists, that is, inputs with no row in the data grounding record, decisions with no stated below-threshold

behaviour, oversight clauses with no reason code, and to leave adjudication with the analyst, whose role shifts toward adjudicating candidates and maintaining the residual-ambiguity register.

7.5. Cross-Domain Generality and Its Limits

The applied survey in Section 3 asked whether the five gates describe a difficulty peculiar to document-heavy casework or one recurring wherever a business decision meets an automated component; the domains reviewed there line up suggestively rather than evidentially. Gate 1 recurs as an obligation originating outside the organisation, arriving as regulatory language already dense with ambiguity (Massey *et al*., 2014) and requiring divergent obligations to be reconciled first (Mbonu *et al*., 2019; Ominyi & Anichukwueze, 2022a). Gate 2 recurs as the demand that a governance judgement be made explicit rather than left to discretion (Mbonu *et al*., 2018). Gate 3 recurs most strongly: architecture derived from constraint rather than retrofitted to it (Aliliele *et al*., 2024b; Atakpa & Abolaji, 2022), data classified by sensitivity with a trace to the governing obligation (Aliliele *et al*., 2024a), and multi-source integration treated as a specification problem (Badmus *et al*., 2019a). Gate 4 recurs as aggregate accuracy concealing uneven error (Komi, 2023) and method choice bearing on conclusions (Sanni *et al*., 2022). Gate 5 recurs as human review treated as a located design element (Ladapo *et al*., 2022, 2024), release gating whose own evaluation is biased (Nwakamma *et al*., 2024a), and components granted more agency than intended (Nwakamma *et al*., 2024b).

Three limits apply, and the survey's expansion has strengthened rather than weakened them. The sources are cited for the claims their titles assert, a substantial number appear in venues without an established review record, and a correspondence of vocabulary is not one of practice; adding more such sources multiplies instances of the same evidential kind without improving it. The survey came from an author-supplied corpus rather than systematic search, so its silences carry no information and its size carries none either. And generality of a *problem* does not imply generality of a *remedy*: several settings surveyed above are one-off design acts admitting no rate, and for those the gates concerning rates do not apply at all.

7.6. Implications for Research and A Validation Agenda

Three designs would test the propositions at different costs. A controlled comparison, in the tradition of Porter *et al.* (1995) and of the experiments aggregated by Dieste and Juristo (2011), could address P1, P2 and P7 by giving matched analyst groups the same ambiguous problem with and without the pipeline. A multi-case field study could address P4 and P5, which concern the relationship between what was estimated at specification time and what was realised. A longitudinal design following deployed automations would be required for P6 and P8. Three further questions arise from the framework's construction: whether the typology imported from software requirements engineering (Berry & Kamsties, 2004; Kamsties, 2005) is right for business-process specification; whether automated detection of the proposed smells is feasible, extending the NLP-for-RE agenda whose industrial maturity Zhao *et al.* (2021) document as limited; and whether the six characteristics are the right six, which a Delphi study or expert survey could establish.

8. Limitations

The contribution is conceptual and unvalidated. The pipeline was constructed by synthesis of published literature and professional experience; it has not been applied under observation, its gates have not been shown to reduce defects, and the illustration in Section 6 is hypothetical throughout and reports no data.

The six AI-readiness quality characteristics are proposed, not empirically derived. They were assembled from the gate structure rather than induced from a corpus of requirements or elicited from practitioners, and their completeness, separability and independence are open questions; the corresponding smells are proposed by analogy with Femmer *et al*. (2017) without the detection studies that grounded that work.

The pipeline assumes a decomposable decision. Where a process is genuinely non-sequential case work of the kind CMMN (Object Management Group, 2016) exists to model the order of activities set by the unfolding case, the decision not repeated in comparable form, and the caseworker's expertise consisting precisely in knowing which considerations apply Gates 2 and 4 will fit poorly, because there is no stable decision to decompose and no population over which a rate can be defined. Recognising that a process falls outside the boundary is itself a useful result, but the framework offers little to those working inside it.

The ambiguity typology is imported. Berry and Kamsties (2004) and Kamsties (2005) developed their categories for software requirements documents; their transfer to business-process specification is asserted here on grounds of conceptual fit rather than tested, and business-process artefacts may exhibit ambiguity types the software typology does not name.

The governance instruments are recent. The NIST framework (National Institute of Standards and Technology, 2023), ISO/IEC 42001 (International Organization for Standardization, 2023a), ISO/IEC 23894 (International Organization for Standardization, 2023b) and the EU regulation (European Parliament and Council, 2024) are all of recent vintage, and the practical burden Gate 5 imposes by invoking them is not yet understood, particularly for smaller organisations without dedicated compliance capability. Its conscientious application may be disproportionate to the risk of many low-stakes automations.

Much of the evidence assembled in Section 2 was produced in settings other than the one the framework addresses, and the transfer is asserted rather than tested. The automation-bias and trust literatures underpinning Section 2.14 and Gate 5 were developed in aviation, laboratory judgement tasks and forecasting rather than administrative casework; the mechanisms described by Bainbridge (1983), Parasuraman and Riley (1997), Skitka *et al*. (1999) and Lee and See (2004) are general in form, but no magnitude is transferred here, and the direction of reliance bias is itself contested (Burton *et al*., 2020; Dietvorst *et al*., 2015; Logg *et al*., 2019). The requirements-engineering-for-AI base against which the paper positions itself is, on its own account, immature, its two mapping studies reporting that techniques are largely unvalidated (Ahmad *et al*., 2023; Villamizar *et al*., 2021); and requirements-quality research likewise lacks a shared theory by the assessment of those building one (Frattini *et al*., 2023), so, the constructs Table 4 claims to extend are themselves in motion.

A substantial portion of the evidence base is of a different

kind, and its size warrants a plain statement. Approaching a third of the reference list consists of the applied and practitioner contributions surveyed in Section 3, a substantial number in venues without an established peer-review record, none independently verified for this paper. They are cited for illustration of the specification problem's generality, and the share of the reference list they occupy should not be mistaken for evidential weight; a reference list is not a measure. No theoretical or quantitative claim in this paper depends on any of them: the pipeline, its ordering principle, the five gates, the six AI-readiness quality characteristics, the smells and the eight propositions were derived from the peer-reviewed literature of Section 2, and every number in the text comes from a peer-reviewed study, a standards document or a practitioner survey caveated in Section 1. Section 3 was written against the completed framework rather than contributing to it, and a reader who removed it entirely would lose the generality argument and nothing else.

Three sources are preprints and are cited as such: the data readiness levels of Lawrence (2017), the interpretability evaluation taxonomy of Doshi-Velez and Kim (2017) and the analysis of label errors in benchmark test sets by Northcutt, Athalye and Mueller (2021) were not formally peer-reviewed at the versions cited. One source, cited once in Section 5.6 for the conceptual claim that structured validation is a specifiable step in small-organisation technology deployment, is drawn from the author's own working corpus and was screened for topical relevance without independent verification of its venue or content (Eyetsemitan *et al*., 2023b); no statistic is drawn from it, and that single claim should be treated as weaker than the others in the same passage.

A caution runs against the paper's own premise: the framework assumes unresolved ambiguity is costly, but de Bruijn and Dekkers (2010) found only one examined failure stemmed from ambiguous requirements. That a probabilistic component cannot cope as those parties did is an argument, not a finding.

Finally, parts of the evidence base are thin. The user-involvement literature is heterogeneous (Bano & Zowghi, 2015), the practitioner surveys cited in Section 1 are vendor-adjacent instruments of unclear sampling, and the most-cited failure observation in the field is a consultancy report (EY, 2016) recirculated without independent verification; and success and failure are in any case enacted judgements rather than properties of a system (Cecez-Kecmanovic *et al*., 2014). The tooling proposals of Section 7.4 rest on a body of work whose own summary is that approximately 7% of studies were assessed industrially and that most tools have not survived (Zhao *et al*., 2021), and the one direct comparison cited found manual inspection outperforming tool support on recall (Dalpiaz *et al*., 2019). Nothing in this paper should be read as a claim that the artefacts it proposes can currently be produced automatically.

9. Conclusion

Requirements engineering knows a great deal about removing ambiguity from statements about system behaviour. Intelligent process automation requires something different: a specification of a decision, grounded in the data available when that decision is taken, bounded by a stated and directional error profile, equipped with a defined behaviour where the system is unsure, and attached to an oversight arrangement that could fail visibly rather than silently.

This paper has proposed the Ambiguity-to-Specification pipeline as the method that produces such a specification: five gates ordered by analytical dependency rather than calendar time, governed by the principle that each ambiguity should be resolved where the information needed to resolve it first becomes available, with the residue specified rather than resolved. It has proposed six AI-readiness quality characteristics extending ISO/IEC/IEEE 29148, and eight propositions by which the whole may be evaluated. The claim is not that the framework has been shown to work, but that the specification gap in intelligent process automation is a business analysis problem before it is a technical one, and that the extensions proposed here name what an adequate apparatus would have to contain. Whether they are the right extensions is an empirical question.

References

1. Abb L, Rehse JR. Process-related user interaction logs: State of the art, reference model, and object-centric implementation. Inf Syst. 2024;124:102386. doi:10.1016/j.is.2024.102386
2. Abraham R, Schneider J, vom Brocke J. Data governance: A conceptual framework, structured review, and research agenda. Int J Inf Manage. 2019;49:424-38. doi:10.1016/j.ijinfomgt.2019.07.008
3. Achimugu P, Selamat A, Ibrahim R, Mahrin MN. A systematic literature review of software requirements prioritization research. Inf Softw Technol. 2014;56(6):568-85. doi:10.1016/j.infsof.2014.02.001
4. Adebayo A, Adegbite MP, Ahmed MO. Adversarial machine learning in critical infrastructure: A conceptual framework for threat modeling AI enabled OT systems. World J Innov Mod Technol. 2022;6(1):184-234. doi:10.56201/wjimt.v6.no1.2022.pg184.234
5. Adebayo A, Adegbite MP, Ahmed MO. AI augmented threat detection in industrial control systems: A systematic review of machine learning approaches for ICS anomaly detection. Int J Eng Mod Technol. 2023;9(3):287-340. doi:10.56201/ijcsmt.v9.no3.2023.pg287.340
6. Adelanwa A, Basnet A, Anene UN. Data driven digital transformation models for lifecycle performance management in infrastructure delivery. Int J Adv Multidiscip Res Stud. 2023;3(6):2646-62. doi:10.62225/2583049X.2023.3.6.5968
7. Adelanwa A, Basnet A, Anene UN. Performance intelligence models for optimization and outcome measurement in large scale public services. Shodhshauryam Int Sci Ref Res J. 2024;6(1).
8. Adesuyi MO, Akomolafe O, Olaogun BO, Ndukwe VU, Sakyi JK. AI-driven risk scoring model for global cross-border trade payment transactions. Int J Adv Multidiscip Res Stud. 2024;4(1):1569-81. doi:10.62225/2583049X.2024.4.1.5281
9. Adeyelu OO, Dagodzo D. A maturity model for predicting airport safety audit outcomes in resource-constrained regulatory environments. Int J Sci Res Civ Eng. 2024;8(4):132-70. doi:10.32628/IJSRCE248423
10. Afrihyia E, Akinse SG, Ojukwu PU. Comparative governance of AI-driven healthcare management: Executive oversight, regulatory structures, and accountability in the United States and developing countries. Int J Adv Multidiscip Res Stud. 2024;4(6):3087-102. doi:10.62225/2583049X.2024.4.6.5920

11. Ahiaeke Patrick MC, Okonkwo CS, Mayo W, Okeke OT. Model for data driven vendor evaluation and bid selection using geospatial intelligence. Shodhshauryam Int Sci Ref Res J. 2021;4(4):426-43. doi:10.32628/SHISRRJ214461
12. Ahmad K, Abdelrazek M, Arora C, Bano M, Grundy J. Requirements engineering for artificial intelligence systems: A systematic mapping study. Inf Softw Technol. 2023;158:107176. doi:10.1016/j.infsof.2023.107176
13. Ahmad K, Bano M, Abdelrazek M, Arora C, Grundy J. What's up with requirements engineering for artificial intelligence systems? In: Proceedings of the 29th IEEE International Requirements Engineering Conference (RE 2021); 2021. p. 1-12. doi:10.1109/RE51729.2021.00008
14. Akinleye OK, Adeyoyin O. Process automation framework for enhancing procurement efficiency and transparency. Shodhshauryam Int Sci Ref Res J. 2021;4(4):356-87.
15. Akin-Oluyomi OT, Atima ME, Akinleye OK. Green procurement strategies for balancing cost efficiency with long-term environmental responsibility. Int J Adv Multidiscip Res Stud. 2023;3(6):2183-93.
16. Akin-Oluyomi OT, Atima ME, Akinleye OK. Regulatory compliance and supplier risk assessment frameworks in international pharmaceutical procurement. Int J Adv Multidiscip Res Stud. 2023;3(6):2194-204.
17. Akomolafe O, Agu MU, Bello A. A conceptual model for implementing risk-based auditing in strategic financial management. Int J Adv Multidiscip Res Stud. 2023;3(6):2274-86. doi:10.62225/2583049X.2023.3.6.5359
18. Aliliele C, Mbonu IS, Iwuanyanwu U. A conceptual framework for continuous cloud misconfiguration monitoring and enterprise risk mitigation strategies. Int J Sci Res Comput Sci Eng Inf Technol. 2023;9(10):373-94. doi:10.32628/CSEIT2361071
19. Aliliele C, Mbonu IS, Iwuanyanwu U. A conceptual framework for enterprise data sensitivity classification and regulatory traceability mechanisms. Int J Adv Multidiscip Res Stud. 2024;4(6):3103-24. doi:10.62225/2583049X.2024.4.6.5991
20. Aliliele C, Mbonu IS, Iwuanyanwu U. Advances in HIPAA compliant data architecture and secure analytics frameworks for community healthcare organizations. Shodhshauryam Int Sci Ref Res J. 2024;7(2):277-324. doi:10.32628/SHISRRJ2472163
21. Amayo EB, Owulade OA, Isi LR. Optimizing project governance in multinational infrastructure projects: Insights from General Electric's global operations. Int J Multidiscip Res Growth Eval. 2023;4(1):975-83. doi:10.54660/.IJMRGE.2023.4.1.975-983
22. Amayo EB, Owulade OA, Isi LR. Risk management strategies and compliance frameworks in healthcare infrastructure projects: Case studies from West Africa. Int J Manag Organ Res. 2023;2(1):161-8. doi:10.54660/IJMOR.2023.2.1.161-168
23. Ameller D, Ayala C, Cabot J, Franch X. How do software architects consider non-functional requirements: An exploratory study. In: 2012 20th IEEE International Requirements Engineering Conference (RE); 2012. p. 41-50. doi:10.1109/RE.2012.6345838
24. Amershi S, Begel A, Bird C, DeLine R, Gall HC, Kamar E, *et al*. Software engineering for machine learning: A case study. In: 2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP); 2019. p. 291-300. doi:10.1109/ICSE-SEIP.2019.00042
25. Aminu-Ibrahim AY, Ogbete JC, Ambali KB. Governance and accountability models for public private partnerships in healthcare infrastructure development. Int J Adv Multidiscip Res Stud. 2024;4(6):2943-60. doi:10.62225/2583049X.2024.4.6.5699
26. Annan AO. Data privacy governance models for cross-border digital platforms. Int J Multidiscip Res Growth Eval. 2022;3(6):949-64.
27. Annan AO. Algorithmic accountability and trade secret protection in artificial intelligence. Shodhshauryam Int Sci Ref Res J. 2024;7(5):315-47.
28. Aranda J, Easterbrook S, Wilson G. Requirements in the wild: How small companies do it. In: 15th IEEE International Requirements Engineering Conference (RE 2007); 2007. p. 39-48. doi:10.1109/RE.2007.54
29. Arnold M, Bellamy RKE, Hind M, Houde S, Mehta S, Mojsilović A, *et al*. FactSheets: Increasing trust in AI services through supplier's declarations of conformity. IBM J Res Dev. 2019;63(4/5):6:1-6:13. doi:10.1147/JRD.2019.2942288
30. Arora C, Sabetzadeh M, Briand L, Zimmer F. Automated checking of conformance to requirements templates using natural language processing. IEEE Trans Softw Eng. 2015;41(10):944-68. doi:10.1109/TSE.2015.2428709
31. Arora C, Sabetzadeh M, Briand LC, Zimmer F. Automated extraction and clustering of requirements glossary terms. IEEE Trans Softw Eng. 2017;43(10):918-45. doi:10.1109/TSE.2016.2635134
32. Ashmore R, Calinescu R, Paterson C. Assuring the machine learning lifecycle: Desiderata, methods, and challenges. ACM Comput Surv. 2021;54(5):1-39. doi:10.1145/3453444
33. Atakpa MI. A review of predictive analytics models for anti-money laundering detection in commercial banking systems. Int J Sci Res Comput Sci Eng Inf Technol. 2021;7(2):778-804. doi:10.32628/CSEIT2064813
34. Atakpa MI, Abetoh NF. A systematic review of machine learning advances in financial fraud detection for banking systems. Int J Sci Res Comput Sci Eng Inf Technol. 2022;8(2):771-801. doi:10.32628/CSEIT23906220
35. Atakpa MI, Abetoh NF, Akeju B. Advances in artificial intelligence for healthcare payment fraud detection: A review of NHS applications. Int J Sci Res Comput Sci Eng Inf Technol. 2024;10(4):1161-94. doi:10.32628/CSEIT26123240
36. Atakpa MI, Abolaji TO. A privacy-preserving data architecture model for regulated industry analytics under GDPR and HIPAA compliance. Int J Sci Res Comput Sci Eng Inf Technol. 2022;8(3):781-808. doi:10.32628/CSEIT22558
37. Atakpa MI, Fobellah AN. Anomaly detection in financial time-series data: A conceptual model for healthcare and banking applications. Int J Sci Res Comput Sci Eng Inf Technol. 2023;9(2):967-97. doi:10.32628/CSEIT2342441
38. Atima ME, Sanni JO, Attah A. Predictive audience

segmentation models resolving targeting inefficiencies in regulated professional service enterprises. Shodhshauryam Int Sci Ref Res J. 2022;5(1):271-303. doi:10.32628/SHISRRJ247134
39. Babatope OM, Akokodaripon DA, Akinleye OK, Okoruwa PO. The role of data visualization in enhancing strategic procurement decision-making effectiveness. Int J Adv Multidiscip Res Stud. 2023;3(6):2205-12.
40. Badmus O, Dosunmu AA, Ozowara DE. A conceptual model for ETL design and data integration in Salesforce-centric enterprise architectures. Iconic Res Eng J. 2019;3(5).
41. Badmus O, Dosunmu AA, Ozowara DE. A governance framework for Salesforce platform management in regulated healthcare environments. Iconic Res Eng J. 2019;3(6).
42. Badmus O, Jooda D, Anunagba CO. A privacy-by-design framework for role-based security architecture in Salesforce environments handling sensitive personal data. Int J Adv Multidiscip Res Stud. 2024;4(6):3244-54. doi:10.62225/2583049X.2024.4.6.6243
43. Badmus O, Jooda D, Ozowara DE, Anunagba CO. A systematic review of MuleSoft ESB integration patterns in multi-cloud Salesforce architectures. Gyanshauryam Int Sci Ref Res J. 2022;5(3):462-81.
44. Bainbridge L. Ironies of automation. Automatica. 1983;19(6):775-9. doi:10.1016/0005-1098(83)90046-8
45. Bano M, Zowghi D. A systematic review on the relationship between user involvement and system success. Inf Softw Technol. 2015;58:148-69. doi:10.1016/j.infsof.2014.06.011
46. Barocas S, Selbst AD. Big data's disparate impact. Calif Law Rev. 2016;104(3):671-732. doi:10.15779/Z38BG31
47. Batini C, Cappiello C, Francalanci C, Maurino A. Methodologies for data quality assessment and improvement. ACM Comput Surv. 2009;41(3):1-52. doi:10.1145/1541880.1541883
48. Batoulis K, Meyer A, Bazhenova E, Decker G, Weske M. Extracting decision logic from process models. In: Advanced Information Systems Engineering (CAiSE 2015). LNCS Vol. 9097. Cham: Springer; 2015. p. 349-66. doi:10.1007/978-3-319-19069-3_22
49. Belani H, Vuković M, Car Ž. Requirements engineering challenges in building AI-based complex systems. In: 2019 IEEE 27th International Requirements Engineering Conference Workshops (REW); 2019. p. 252-5. doi:10.1109/REW.2019.00051
50. Bello A, Akomolafe O, Agu MU. A conceptual framework for data-driven procurement risk management in multinational enterprises. Int J Adv Multidiscip Res Stud. 2023;3(6):2307-16. doi:10.62225/2583049X.2023.3.6.5363
51. Berander P, Andrews A. Requirements prioritization. In: Aurum A, Wohlin C, editors. Engineering and managing software requirements. Springer; 2005. p. 69-94. doi:10.1007/3-540-28244-0_4
52. Berry DM. The importance of ignorance in requirements engineering. J Syst Softw. 1995;28(1):179-84. doi:10.1016/0164-1212(94)00054-Q
53. Berry DM, Kamsties E. Ambiguity in requirements specification. In: Perspectives on Software Requirements. Boston: Springer; 2004. p. 7-44. doi:10.1007/978-1-4615-0465-8_2
54. Berry D, Gacitua R, Sawyer P, Tjong SF. The case for dumb requirements engineering tools. In: Requirements Engineering: Foundation for Software Quality (REFSQ 2012). LNCS Vol. 7195. Springer; 2012. p. 211-7. doi:10.1007/978-3-642-28714-5_18
55. Bertsimas D, Kallus N. From predictive to prescriptive analytics. Manage Sci. 2020;66(3):1025-44. doi:10.1287/mnsc.2018.3253
56. Bhatt U, Xiang A, Sharma S, Weller A, Taly A, Jia Y, *et al*. Explainable machine learning in deployment. In: Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency; 2020. p. 648-57. doi:10.1145/3351095.3375624
57. Board of Governors of the Federal Reserve System, Office of the Comptroller of the Currency. Supervisory guidance on model risk management (SR Letter 11-7). Board of Governors of the Federal Reserve System; 2011. Available from: https://www.federalreserve.gov/boarddocs/srletters/2011/sr1107.htm
58. Borg M, Runeson P, Ardö A. Recovering from a decade: A systematic mapping of information retrieval approaches to software traceability. Empir Softw Eng. 2014;19(6):1565-616. doi:10.1007/s10664-013-9255-y
59. Bosco A, Augusto A, Dumas M, La Rosa M, Fortino G. Discovering automatable routines from user interaction logs. In: Business Process Management Forum (BPM 2019). LNBIP Vol. 360. Cham: Springer; 2019. p. 144-62. doi:10.1007/978-3-030-26643-1_9
60. Burton JW, Stein MK, Jensen TB. A systematic review of algorithm aversion in augmented decision making. J Behav Decis Mak. 2020;33(2):220-39. doi:10.1002/bdm.2155
61. Carmona J, van Dongen B, Solti A, Weidlich M. Conformance checking: Relating processes and models. Springer; 2018. doi:10.1007/978-3-319-99414-7
62. Cecez-Kecmanovic D, Kautz K, Abrahall R. Reframing success and failure of information systems: A performative perspective. MIS Q. 2014;38(2):561-88. doi:10.25300/misq/2014/38.2.11
63. Chakraborti T, Isahagian V, Khalaf R, Khazaeni Y, Muthusamy V, Rizk Y, *et al*. From robotic process automation to intelligent process automation: Emerging trends. In: Business process management: Blockchain and robotic process automation forum. LNBIP. Springer; 2020. p. 215-28. doi:10.1007/978-3-030-58779-6_15
64. Chantree F, Nuseibeh B, de Roeck AN, Willis A. Identifying nocuous ambiguities in natural language requirements. In: Proceedings of the 14th IEEE International Requirements Engineering Conference (RE'06); 2006. p. 56-65. doi:10.1109/RE.2006.31
65. Chengalur-Smith IN, Ballou DP, Pazer HL. The impact of data quality information on decision making: An exploratory analysis. IEEE Trans Knowl Data Eng. 1999;11(6):853-64. doi:10.1109/69.824597
66. Chung L, Nixon BA, Yu E, Mylopoulos J. Non-functional requirements in software engineering. International Series in Software Engineering, Vol. 5. Springer; 2000. doi:10.1007/978-1-4615-5269-7
67. Cleland-Huang J, Gotel OCZ, Huffman Hayes J, Mäder P, Zisman A. Software traceability: Trends and future directions. In: Future of Software Engineering Proceedings (FOSE 2014); 2014. p. 55-69. doi:10.1145/2593882.2593891

68. Cockburn A. Writing effective use cases. Addison-Wesley; 2001.
69. Costanza-Chock S, Raji ID, Buolamwini J. Who audits the auditors? Recommendations from a field scan of the algorithmic auditing ecosystem. In: 2022 ACM Conference on Fairness, Accountability, and Transparency; 2022. p. 1571-83. doi:10.1145/3531146.3533213
70. Dalpiaz F, Ferrari A, Franch X, Palomares C. Natural language processing for requirements engineering: The best is yet to come. IEEE Softw. 2018;35(5):115-9. doi:10.1109/MS.2018.3571242
71. Dalpiaz F, van der Schalk I, Brinkkemper S, Aydemir FB, Lucassen G. Detecting terminological ambiguity in user stories: Tool and experimentation. Inf Softw Technol. 2019;110:3-16. doi:10.1016/j.infsof.2018.12.007
72. de Bruijn F, Dekkers HL. Ambiguity in natural language software requirements: A case study. In: Requirements Engineering: Foundation for Software Quality (REFSQ 2010). LNCS Vol. 6182. Springer; 2010. p. 233-47. doi:10.1007/978-3-642-14192-8_21
73. Di Ciccio C, Marrella A, Russo A. Knowledge-intensive processes: Characteristics, requirements and analysis of contemporary approaches. J Data Semant. 2015;4(1):29-57. doi:10.1007/s13740-014-0038-4
74. Dieste O, Juristo N. Systematic review and aggregation of empirical studies on elicitation techniques. IEEE Trans Softw Eng. 2011;37(2):283-304. doi:10.1109/TSE.2010.33
75. Dietvorst BJ, Simmons JP, Massey C. Algorithm aversion: People erroneously avoid algorithms after seeing them err. J Exp Psychol Gen. 2015;144(1):114-26. doi:10.1037/xge0000033
76. Doshi-Velez F, Kim B. Towards a rigorous science of interpretable machine learning. arXiv:1702.08608 [Preprint]. 2017. Available from: https://arxiv.org/abs/1702.08608
77. Dumas M, Fournier F, Limonad L, Marrella A, Montali M, Rehse JR, *et al*. AI-augmented business process management systems: A research manifesto. ACM Trans Manage Inf Syst. 2023;14(1):1-19. doi:10.1145/3576047
78. Dumas M, La Rosa M, Leno V, Polyvyanyy A, Maggi FM. Robotic process mining. In: van der Aalst WMP, Carmona J, editors. Process mining handbook. LNBIP Vol. 448. Cham: Springer; 2022. p. 468-91. doi:10.1007/978-3-031-08848-3_16
79. Eckhardt J, Vogelsang A, Méndez Fernández D. Are "non-functional" requirements really non-functional? An investigation of non-functional requirements in practice. In: Proceedings of the 38th International Conference on Software Engineering (ICSE '16); 2016. p. 832-42. doi:10.1145/2884781.2884788
80. Edivri J, Oteri O. A conceptual KPI-driven decision and optimization framework for IT service delivery, portfolio performance, and adoption. Gyanshauryam Int Sci Ref Res J. 2024;7(1):167-87. doi:10.32628/GISRRJ246643
81. Egger A, ter Hofstede AHM, Kratsch W, Leemans SJJ, Röglinger M, Wynn MT. Bot log mining: Using logs from robotic process automation for process mining. In: Conceptual modeling. LNCS. Springer; 2020. p. 51-61. doi:10.1007/978-3-030-62522-1_4
82. Ehrlinger L, Wöß W. A survey of data quality measurement and monitoring tools. Front Big Data. 2022;5:850611. doi:10.3389/fdata.2022.850611
83. Endicott TAO. Vagueness in law. Oxford University Press; 2000. doi:10.1093/acprof:oso/9780198268406.001.0001
84. Engel C, Ebel P, Leimeister JM. Cognitive automation. Electron Mark. 2022;32(1):339-50. doi:10.1007/s12525-021-00519-7
85. Engel C, Elshan E, Ebel P, Leimeister JM. Stairway to heaven or highway to hell: A model for assessing cognitive automation use cases. J Inf Technol. 2024;39(1):94-122. doi:10.1177/02683962231185599
86. European Parliament and Council. Regulation (EU) 2024/1689 of 13 June 2024 laying down harmonised rules on artificial intelligence (Artificial Intelligence Act). Off J Eur Union. 2024 Jul 12. Available from: http://data.europa.eu/eli/reg/2024/1689/oj
87. EY. Get ready for robots: Why planning makes the difference between success and disappointment. EYGM Limited; 2016.
88. Eyetsemitan RA, Ambali KB, Oyeleye AO, Fadayomi O. Multi-stakeholder governance alignment in joint venture operations: A conceptual framework for coordinating business processes in highly regulated environments. Iconic Res Eng J. 2020;4(4):418-41. doi:10.64388/IREV4I4-1716955
89. Eyetsemitan RA, Ambali KB, Oyeleye AO, Fadayomi O. Translating tax and regulatory requirements into SME compliance workflows: A conceptual framework for implementing IAS 12, VAT, PAYE, and withholding tax. Iconic Res Eng J. 2021;4(11):621-41. doi:10.64388/IREV4I11-1716956
90. Eyetsemitan RA, Ambali KB, Oyeleye AO, Fadayomi O. User acceptance testing in small business technology deployment: A structured validation framework for lean operational environments. Int J Multidiscip Res Growth Eval. 2023;4(6):1512-31. doi:10.54660/IJMRGE.2023.4.6.1512-1531
91. Eze FI, Akinleye OK, Anene UN. Development of a risk-based accelerated approval decision model balancing speed, safety, and evidence quality in pharmaceutical regulatory science. Res J Pure Sci Technol. 2024;7(6):120-61. doi:10.56201/rjpst.v7.no6.2024.pg120.161
92. Eze FI, Anene UN, Akinleye OK. Creation of a drug shortage early warning and regulatory response model for essential medicines in global health systems. Res J Pure Sci Technol. 2022;5(1):57-95. doi:10.56201/rjpst.v5.no1.2022.pg57.95
93. Ezzini S, Abualhaija S, Arora C, Sabetzadeh M, Briand LC. Automated handling of anaphoric ambiguity in requirements: A multi-solution study. In: Proceedings of the 44th International Conference on Software Engineering (ICSE '22); 2022. doi:10.1145/3510003.3510157
94. Femmer H, Vogelsang A. Requirements quality is quality in use. IEEE Softw. 2019;36(3):83-91. doi:10.1109/MS.2018.110161823
95. Femmer H, Méndez Fernández D, Wagner S, Eder S. Rapid quality assurance with requirements smells. J Syst Softw. 2017;123:190-213. doi:10.1016/j.jss.2016.02.047
96. Ferrari A, Esuli A. An NLP approach for cross-domain ambiguity detection in requirements engineering. Autom

Softw Eng. 2019;26(3):559-98. doi:10.1007/s10515-019-00261-7
97. Ferrari A, Gori G, Rosadini B, Trotta I, Bacherini S, Fantechi A, *et al*. Detecting requirements defects with NLP patterns: An industrial experience in the railway domain. Empir Softw Eng. 2018;23(6):3684-733. doi:10.1007/s10664-018-9596-7
98. Ferrari A, Spoletini P, Gnesi S. Ambiguity and tacit knowledge in requirements elicitation interviews. Requir Eng. 2016;21(3):333-55. doi:10.1007/s00766-016-0249-3
99. Fisher CW, Chengalur-Smith I, Ballou DP. The impact of experience and time on the use of data quality information in decision making. Inf Syst Res. 2003;14(2):170-88. doi:10.1287/isre.14.2.170.16017
100. Frattini J. Identifying relevant factors of requirements quality: An industrial case study. In: Requirements Engineering: Foundation for Software Quality (REFSQ 2024). LNCS Vol. 14588. Springer; 2024. p. 20-36. doi:10.1007/978-3-031-57327-9_2
101. Frattini J, Montgomery L, Fischbach J, Mendez D, Fucci D, Unterkalmsteiner M. Requirements quality research: A harmonized theory, evaluation, and roadmap. Requir Eng. 2023;28(4):507-20. doi:10.1007/s00766-023-00405-y
102. Fügener A, Grahl J, Gupta A, Ketter W. Will humans-in-the-loop become borgs? Merits and pitfalls of working with AI. MIS Q. 2021;45(3):1527-56. doi:10.25300/misq/2021/16553
103. Gebru T, Morgenstern J, Vecchione B, Vaughan JW, Wallach H, Daumé III H, *et al*. Datasheets for datasets. Commun ACM. 2021;64(12):86-92. doi:10.1145/3458723
104. Gervasi V, Zowghi D. Reasoning about inconsistencies in natural language requirements. ACM Trans Softw Eng Methodol. 2005;14(3):277-330. doi:10.1145/1072997.1072999
105. Gervasi V, Ferrari A, Zowghi D, Spoletini P. Ambiguity in requirements engineering: Towards a unifying framework. In: From Software Engineering to Formal Methods and Tools, and Back. LNCS Vol. 11865. Springer; 2019. p. 191-210. doi:10.1007/978-3-030-30985-5_12
106. Gervasi V, Gacitua R, Rouncefield M, Sawyer P, Kof L, Ma L, *et al*. Unpacking tacit knowledge for requirements engineering. In: Maalej W, Thurimella AK, editors. Managing requirements knowledge. Springer; 2013. p. 23-47. doi:10.1007/978-3-642-34419-0_2
107. Giray G. A software engineering perspective on engineering machine learning systems: State of the art and challenges. J Syst Softw. 2021;180:111031. doi:10.1016/j.jss.2021.111031
108. Glinz M. On non-functional requirements. In: 15th IEEE International Requirements Engineering Conference (RE 2007); 2007. p. 21-6. doi:10.1109/RE.2007.45
109. Goguen JA, Linde C. Techniques for requirements elicitation. In: Proceedings of the IEEE International Symposium on Requirements Engineering; 1993. p. 152-64. doi:10.1109/ISRE.1993.324822
110. Gotel OCZ, Finkelstein ACW. An analysis of the requirements traceability problem. In: Proceedings of the IEEE International Conference on Requirements Engineering (ICRE '94); 1994. p. 94-101. doi:10.1109/ICRE.1994.292398
111. Green B. The flaws of policies requiring human oversight of government algorithms. Comput Law Secur Rev. 2022;45:105681. doi:10.1016/j.clsr.2022.105681
112. Grønsund T, Aanestad M. Augmenting the algorithm: Emerging human-in-the-loop work configurations. J Strateg Inf Syst. 2020;29(2):101614. doi:10.1016/j.jsis.2020.101614
113. Guidotti R, Monreale A, Ruggieri S, Turini F, Giannotti F, Pedreschi D. A survey of methods for explaining black box models. ACM Comput Surv. 2018;51(5):1-42. doi:10.1145/3236009
114. Habibullah KM, Gay G, Horkoff J. Non-functional requirements for machine learning: Understanding current use and challenges among practitioners. Requir Eng. 2023;28(2):283-316. doi:10.1007/s00766-022-00395-3
115. Hashmi M, Governatori G, Lam HP, Wynn MT. Are we done with business process compliance: State of the art and challenges ahead. Knowl Inf Syst. 2018;57(1):79-133. doi:10.1007/s10115-017-1142-1
116. Hasić F, De Smedt J, Vanthienen J. Augmenting processes with decision intelligence: Principles for integrated modelling. Decis Support Syst. 2018;107:1-12. doi:10.1016/j.dss.2017.12.008
117. Heinrich B, Hristova D, Klier M, Schiller A, Szubartowicz M. Requirements for data quality metrics. J Data Inf Qual. 2017;9(2):1-32. doi:10.1145/3148238
118. Herm LV, Janiesch C, Helm A, Imgrund F, Hofmann A, Winkelmann A. A framework for implementing robotic process automation projects. Inf Syst e-Bus Manag. 2023;21(1):1-35. doi:10.1007/s10257-022-00553-8
119. High-Level Expert Group on Artificial Intelligence. Ethics Guidelines for Trustworthy AI. Brussels: European Commission; 2019. Available from: https://digital-strategy.ec.europa.eu/en/library/ethics-guidelines-trustworthy-ai
120. Hofmann HF, Lehner F. Requirements engineering as a success factor in software projects. IEEE Softw. 2001;18(4):58-66. doi:10.1109/MS.2001.936219
121. Holstein K, Wortman Vaughan J, Daumé H III, Dudík M, Wallach H. Improving fairness in machine learning systems: What do industry practitioners need? In: Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems; 2019. p. 1-16. doi:10.1145/3290605.3300830
122. Horkoff J. Non-functional requirements for machine learning: Challenges and new directions. In: 2019 IEEE 27th International Requirements Engineering Conference (RE); 2019. p. 386-91. doi:10.1109/RE.2019.00050
123. Huang F, Vasarhelyi MA. Applying robotic process automation (RPA) in auditing: A framework. Int J Account Inf Syst. 2019;35:100433. doi:10.1016/j.accinf.2019.100433
124. Hutchinson B, Smart A, Hanna A, Denton E, Greer C, Kjartansson O, *et al*. Towards accountability for machine learning datasets: Practices from software engineering and infrastructure. In: Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency; 2021. p. 560-75. doi:10.1145/3442188.3445918
125. Ike PN, Aifuwa SE, Nnabueze SB, Olatunde-Thorpe J, Ogbuefi E, Oshoba TO, *et al*. Quantitative risk architecture for public-private partnerships: A multi-

layered model for allocating public and private risk. Int J Adv Multidiscip Res Stud. 2024;4(6):2669-82. doi:10.62225/2583049X.2024.4.6.5021
126. Ilodigwe L, Adesemoye AC. From molecules to health systems: A conceptual model explaining why scientific innovation fails to improve patient outcomes without effective healthcare delivery infrastructure. Int J Health Pharm Res. 2019;5(2):56-79. doi:10.56201/ijhpr.vol.5.no2.2019.pg56.79
127. Ilodigwe L, Adesemoye AC. The data backbone of health system transformation: A governance and architecture framework for interoperability, data quality, and advanced analytics at national scale. Int J Health Pharm Res. 2021;6(2):52-76. doi:10.56201/ijhpr.vol.6.no2.2021.pg52.76
128. Ilodigwe L, Adesemoye AC. Advances in real-world evidence, predictive analytics, and data-driven decision making for improving health system performance and patient outcomes. Int J Health Pharm Res. 2024;9(5):174-97. doi:10.56201/ijhpr.v9.no5.2024.pg174.197
129. Inayat I, Salim SS, Marczak S, Daneva M, Shamshirband S. A systematic literature review on agile requirements engineering practices and challenges. Comput Hum Behav. 2015;51(Part B):915-29. doi:10.1016/j.chb.2014.10.046
130. International Institute of Business Analysis. A Guide to the Business Analysis Body of Knowledge (BABOK Guide). Version 3. Toronto: International Institute of Business Analysis; 2015.
131. International Organization for Standardization. ISO/IEC/IEEE 29148:2018 Systems and software engineering — Life cycle processes — Requirements engineering. 2nd ed. Geneva: ISO; 2018. Available from: https://www.iso.org/standard/72089.html
132. International Organization for Standardization. ISO/IEC 42001:2023 Information technology — Artificial intelligence — Management system. Geneva: ISO; 2023. Available from: https://www.iso.org/standard/42001
133. International Organization for Standardization. ISO/IEC 23894:2023 Information technology — Artificial intelligence — Guidance on risk management. Geneva: ISO; 2023. Available from: https://www.iso.org/standard/77304.html
134. International Organization for Standardization, International Electrotechnical Commission. Artificial intelligence — Data quality for analytics and machine learning (ML) — Part 2: Data quality measures (ISO/IEC 5259-2:2024). 2024. Available from: https://www.iso.org/standard/81860.html
135. Ivančić L, Suša Vugec D, Bosilj Vukšić V. Robotic process automation: Systematic literature review. In: Business Process Management: Blockchain and Central and Eastern Europe Forum (BPM 2019). LNBIP Vol. 361. Cham: Springer; 2019. p. 280-95. doi:10.1007/978-3-030-30429-4_19
136. Jimenez-Ramirez A, Reijers HA, Barba I, Del Valle C. A method to improve the early stages of the robotic process automation lifecycle. In: Advanced information systems engineering (CAiSE). LNCS. Springer; 2019. p. 446-61. doi:10.1007/978-3-030-21290-2_28
137. Jørgensen M, Grimstad S. The impact of irrelevant and misleading information on software development effort estimates: A randomized controlled field experiment. IEEE Trans Softw Eng. 2011;37(5):695-707. doi:10.1109/TSE.2010.78
138. Kamsties E. Understanding ambiguity in requirements engineering. In: Aurum A, Wohlin C, editors. Engineering and Managing Software Requirements. Berlin: Springer; 2005. p. 245-66. doi:10.1007/3-540-28244-0_11
139. Karlsson J, Ryan K. A cost–value approach for prioritizing requirements. IEEE Softw. 1997;14(5):67-74. doi:10.1109/52.605933
140. Khatri V, Brown CV. Designing data governance. Commun ACM. 2010;53(1):148-52. doi:10.1145/1629175.1629210
141. Kiyavitskaya N, Zeni N, Mich L, Berry DM. Requirements for tools for ambiguity identification and measurement in natural language requirements specifications. Requir Eng. 2008;13(3):207-39. doi:10.1007/s00766-008-0063-7
142. Komi NM. Hybrid deep learning for wind power forecasting: From grid stability to market equity for small generators. Int J Multidiscip Res Growth Eval. 2021;2(6):989-1014. doi:10.54660/.IJMRGE.2021.2.6.989-1014
143. Komi NM. When accurate forecasts are not fair forecasts: Deep learning load prediction and equity across districts. Shodhshauryam Int Sci Ref Res J. 2023;6(6):476-546. doi:10.32628/SHISRRJ247102
144. Komi NM, Adamolekun A. Interpretable machine learning for early failure prediction in distributed renewable energy assets. Int J Multidiscip Res Growth Eval. 2021;2(6):1015-38. doi:10.54660/.IJMRGE.2021.2.6.1015-1038
145. Komi NM, Adeniji IO. A tiered digital twin that brings predictive diagnostics to resource-constrained renewable energy sites. Int J Eng Mod Technol. 2023;9(3):287-347. doi:10.56201/ijemt.v9.no3.2023.pg287.347
146. Kotonya G, Sommerville I. Requirements Engineering: Processes and Techniques. Chichester: John Wiley & Sons; 1998.
147. Kreuzberger D, Kühl N, Hirschl S. Machine learning operations (MLOps): Overview, definition, and architecture. IEEE Access. 2023;11:31866-79. doi:10.1109/ACCESS.2023.3262138
148. Kumuyi O, Uzoka E, Akeju B, Ozowara DE. Architecture for machine learning-enabled predictive energy management using IoT sensor networks. Int J Adv Multidiscip Res Stud. 2024;4(6):3138-49. doi:10.62225/2583049X.2024.4.6.6004
149. Kumuyi O, Uzoka E, Akeju B, Ozowara DE. Framework for privacy-focused digital identity verification supporting financial inclusion in Africa. Int J Adv Multidiscip Res Stud. 2024;4(6):3150-62. doi:10.62225/2583049X.2024.4.6.6005
150. Ladapo OO, Dosunmu AA, Jooda D, Abolaji TO. Human-in-the-loop machine learning: A state of the art. J Front Multidiscip Res. 2022;3(1):656-69. doi:10.54660/.JFMR.2022.3.1.656-669
151. Ladapo OO, Jooda D, Dosunmu AA, Abolaji TO. Implementation of Active Directory for efficient management of enterprise networks. Iconic Res Eng J. 2019;3(4):608-27. doi:10.64388/IREV3I4-1717206
152. Ladapo OO, Jooda D, Dosunmu AA, Abolaji TO. A comprehensive survey on ServiceNow for IT service management. Int J Multidiscip Res Growth Eval.

2023;4(6):1532-45. doi:10.54660/.IJMRGE.2023.4.6.1532-1545
153. Ladapo OO, Jooda D, Dosunmu AA, Abolaji TO. Systematic literature review on security access control policies and techniques based on privacy requirements in a BYOD environment. Int J Adv Multidiscip Res Stud. 2023;3(6):2890-904. doi:10.62225/2583049X.2023.3.6.6206
154. Ladapo OO, Jooda D, Dosunmu AA, Abolaji TO. Keeping humans in the loop: Human-centered automated annotation with generative AI. Int J Multidiscip Futuristic Dev. 2024;5(1):81-95. doi:10.54660/IJMFD.2024.5.1.81-95
155. Lami G, Fusani M, Trentanni G. QuARS: A pioneer tool for NL requirement analysis. In: From Software Engineering to Formal Methods and Tools, and Back. LNCS Vol. 11865. Springer; 2019. p. 211-9. doi:10.1007/978-3-030-30985-5_13
156. Lawal OA, Oduleye TE. A conceptual decision model for capital allocation using financial analytics. Gyanshauryam Int Sci Ref Res J. 2021;4(2):269-95.
157. Lawrence ND. Data readiness levels. arXiv:1705.02245. 2017. Available from: https://arxiv.org/abs/1705.02245
158. Lee JD, See KA. Trust in automation: Designing for appropriate reliance. Hum Factors. 2004;46(1):50-80. doi:10.1518/hfes.46.1.50_30392
159. Lee YW, Strong DM, Kahn BK, Wang RY. AIMQ: A methodology for information quality assessment. Inf Manage. 2002;40(2):133-46. doi:10.1016/S0378-7206(02)00043-5
160. Leno V, Augusto A, Dumas M, La Rosa M, Maggi FM, Polyvyanyy A. Identifying candidate routines for robotic process automation from unsegmented UI logs. In: 2020 2nd International Conference on Process Mining (ICPM); 2020. p. 153-60. doi:10.1109/icpm49681.2020.00031
161. Leno V, Polyvyanyy A, Dumas M, La Rosa M, Maggi FM. Robotic process mining: Vision and challenges. Bus Inf Syst Eng. 2021;63(3):301-14. doi:10.1007/s12599-020-00641-4
162. Logg JM, Minson JA, Moore DA. Algorithm appreciation: People prefer algorithmic to human judgment. Organ Behav Hum Decis Process. 2019;151:90-103. doi:10.1016/j.obhdp.2018.12.005
163. Lucassen G, Dalpiaz F, van der Werf JMEM, Brinkkemper S. Improving agile requirements: The Quality User Story framework and tool. Requir Eng. 2016;21(3):383-403. doi:10.1007/s00766-016-0250-x
164. Luitel D, Hassani S, Sabetzadeh M. Improving requirements completeness: Automated assistance through large language models. Requir Eng. 2024;29(1):73-95. doi:10.1007/s00766-024-00416-3
165. Ly LT, Maggi FM, Montali M, Rinderle-Ma S, van der Aalst WMP. Compliance monitoring in business processes: Functionalities, application, and tool-support. Inf Syst. 2015;54:209-34. doi:10.1016/j.is.2015.02.007
166. Madaio MA, Stark L, Wortman Vaughan J, Wallach H. Co-designing checklists to understand organizational challenges and opportunities around fairness in AI. In: Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems; 2020. p. 1-14. doi:10.1145/3313831.3376445
167. Mäder P, Egyed A. Do developers benefit from requirements traceability when evolving and maintaining a software system? Empir Softw Eng. 2015;20(2):413-41. doi:10.1007/s10664-014-9314-z
168. Mairiza D, Zowghi D, Nurmuliani N. An investigation into the notion of non-functional requirements. In: Proceedings of the 2010 ACM Symposium on Applied Computing (SAC '10); 2010. p. 311-7. doi:10.1145/1774088.1774153
169. Mäntymäki M, Minkkinen M, Birkstedt T, Viljanen M. Defining organizational AI governance. AI Ethics. 2022;2(4):603-9. doi:10.1007/s43681-022-00143-x
170. Martínez-Fernández S, Bogner J, Franch X, Oriol M, Siebert J, Trendowicz A, *et al*. Software engineering for AI-based systems: A survey. ACM Trans Softw Eng Methodol. 2022;31(2):37e. doi:10.1145/3487043
171. Massey AK, Rutledge R, Antón AI, Swire PP. Identifying and classifying ambiguity for regulatory requirements. In: Proceedings of the 22nd IEEE International Requirements Engineering Conference (RE 2014); 2014. p. 83-92. doi:10.1109/RE.2014.6912250
172. Mavin A, Wilkinson P. Ten years of EARS. IEEE Softw. 2019;36(5):10-4. doi:10.1109/MS.2019.2921164
173. Mavin A, Wilkinson P, Harwood A, Novak M. Easy Approach to Requirements Syntax (EARS). In: 2009 17th IEEE International Requirements Engineering Conference; 2009. p. 317-22. doi:10.1109/RE.2009.9
174. Mbonu IS, Aliliele C, Iwuanyanwu U, Oluoha OM. A conceptual framework for legal and ethical risk modeling in enterprise data protection governance systems. Iconic Res Eng J. 2018;2(2):207-26. doi:10.64388/IREV2I2-1714911
175. Mbonu IS, Aliliele C, Iwuanyanwu U, Uzoka E. A review of identity and access management integration strategies in hybrid and multi cloud environments. Int J Multidiscip Res Growth Eval. 2020;1(5):795-810. doi:10.54660/.IJMRGE.2020.1.5.795-810
176. Mbonu IS, Aliliele C, Iwuanyanwu U, Uzoka E. A conceptual framework for AI enabled IT general controls and SOX audit automation processes. Gyanshauryam Int Sci Ref Res J. 2022;5(5):384-414. doi:10.32628/GISRRJ2256239
177. Mbonu IS, Aliliele C, Iwuanyanwu U, Oluoha OM. A review of comparative data protection regulations and secure cloud implementation strategies across jurisdictions. Iconic Res Eng J. 2019;2(9):482-501. doi:10.64388/IREV2I9-1714912
178. Mehrabi N, Morstatter F, Saxena N, Lerman K, Galstyan A. A survey on bias and fairness in machine learning. ACM Comput Surv. 2021;54(6):1-35. doi:10.1145/3457607
179. Méndez Fernández D, Wagner S, Kalinowski M, Felderer M, Mafra P, Vetrò A, *et al*. Naming the pain in requirements engineering. Empir Softw Eng. 2017;22(5):2298-338. doi:10.1007/s10664-016-9451-7
180. Miller T. Explanation in artificial intelligence: Insights from the social sciences. Artif Intell. 2019;267:1-38. doi:10.1016/j.artint.2018.07.007
181. Mitchell M, Wu S, Zaldivar A, Barnes P, Vasserman L, Hutchinson B, *et al*. Model cards for model reporting. In: Proceedings of the Conference on Fairness, Accountability, and Transparency; 2019. p. 220-9. doi:10.1145/3287560.3287596
182. Mohanani R, Ralph P, Shreeve B. Requirements fixation. In: Proceedings of the 36th International

Conference on Software Engineering (ICSE '14); 2014. p. 895-906. doi:10.1145/2568225.2568235
183. Mohanani R, Salman I, Turhan B, Rodríguez P, Ralph P. Cognitive biases in software engineering: A systematic mapping study. IEEE Trans Softw Eng. 2020;46(12):1318-39. doi:10.1109/TSE.2018.2877759
184. Montgomery L, Fucci D, Bouraffa A, Scholz L, Maalej W. Empirical research on requirements quality: A systematic mapping study. Requir Eng. 2022;27(2):183-209. doi:10.1007/s00766-021-00367-z
185. Nahar N, Zhou S, Lewis GA, Kästner C. Collaboration challenges in building ML-enabled systems: Communication, documentation, engineering, and process. In: Proceedings of the 44th International Conference on Software Engineering (ICSE '22); 2022. p. 413-25. doi:10.1145/3510003.3510209
186. National Institute of Standards and Technology. Artificial Intelligence Risk Management Framework (AI RMF 1.0) (NIST AI 100-1). Gaithersburg, MD: NIST; 2023. doi:10.6028/NIST.AI.100-1
187. Nelson RR, Todd PA, Wixom BH. Antecedents of information and system quality: An empirical examination within the context of data warehousing. J Manage Inf Syst. 2005;21(4):199-236. doi:10.1080/07421222.2005.11045823
188. Ng KKH, Chen CH, Lee CKM, Jiao J, Yang ZX. A systematic literature review on intelligent automation: Aligning concepts from theory, practice, and future perspectives. Adv Eng Inform. 2021;47:101246. doi:10.1016/j.aei.2021.101246
189. Nnabueze SB, Ike PN, Olatunde-Thorpe J, Aifuwa SE, Oshoba TO, Ogbuefi E, *et al*. End-to-end visibility frameworks improving transparency, compliance, and traceability across complex global supply chain operations. Int J Multidiscip Futuristic Dev. 2021;2(2):50-60. doi:10.54660/IJMFD.2021.2.2.50-60
190. Nonaka I, Takeuchi H. The knowledge-creating company: How Japanese companies create the dynamics of innovation. Oxford University Press; 1995.
191. Northcutt CG, Athalye A, Mueller J. Pervasive label errors in test sets destabilize machine learning benchmarks. arXiv:2103.14749 [Preprint]. 2021. Available from: https://arxiv.org/abs/2103.14749
192. Nuseibeh B, Easterbrook S. Requirements engineering: A roadmap. In: Proceedings of the Conference on the Future of Software Engineering (ICSE '00); 2000. p. 35-46. doi:10.1145/336512.336523
193. Nwakamma S, Ojukwu J, Oyesiji SO. LLM-as-a-judge for automated model governance: A review of methods, biases, and release-gating practices. World J Innov Mod Technol. 2024;8(6):185-216. doi:10.56201/wjimt.v8.no6.2024.pg185.216
194. Nwakamma S, Ojukwu J, Oyesiji SO. Securing agentic AI enterprise workflows against prompt injection, tool poisoning, memory manipulation, and excessive agency threats. World J Innov Mod Technol. 2024;8(6):217-48. doi:10.56201/wjimt.v8.no6.2024.pg217.248
195. Object Management Group. Business Process Model and Notation (BPMN), Version 2.0.2 (OMG Document formal/13-12-09). 2014. Available from: https://www.omg.org/spec/BPMN/2.0.2/
196. Object Management Group. Case Management Model and Notation (CMMN), Version 1.1 (OMG Document formal/16-12-01). 2016. Available from: https://www.omg.org/spec/CMMN/
197. Object Management Group. Semantics of Business Vocabulary and Business Rules (SBVR), Version 1.5 (OMG Document formal/19-10-02). 2019. Available from: https://www.omg.org/spec/SBVR/
198. Object Management Group. Decision Model and Notation (DMN), Version 1.5 (OMG Document formal/24-01-01). 2024. Available from: https://www.omg.org/spec/DMN/1.5/
199. Ogbole JI, Okoruwa PO, Fadayomi O, Abolaji TO, Edivri J, Akeju B. Conceptual model for identity-centric zero trust architecture in enterprise security governance. Int J Sci Res Comput Sci Eng Inf Technol. 2021;7(5):393-415. doi:10.32628/IJSRCSEIT217562
200. Ogundairo KM, Ayivi-Donkor SS. Machine learning approaches to customer acquisition and retention: A framework for predictive lifecycle management. IIARD Int J Econ Bus Manage. 2023;9(10):210-46. doi:10.56201/ijebm.v9.no10.2023.pg210.246
201. Ogundairo KM, Ayivi-Donkor SS. Bridging the "adoption gap": Why supply chain AI fails without product marketing principles. Int J Mark Commun Stud. 2024;8(5):169-205. doi:10.56201/ijmcs.v8.no5.2024.pg169.205
202. Ogundairo KM, Ayivi-Donkor SS. The vanishing interface: Why the coming era of autonomous agents demands a paradigm shift in product marketing architecture. IIARD Int J Econ Bus Manage. 2024;10(11):350-85. doi:10.56201/ijebm.v10.no11.2024.pg350.385
203. Okonkwo CS, Agbabiaka J, Mayo W, Okeke OT. Supply chain automation framework using service management platforms. Shodhshauryam Int Sci Ref Res J. 2024;7(2):157-77. doi:10.32628/SHISRRJ2472157
204. Okoruwa PO, Fadayomi O, Abolaji TO, Edivri J, Ogbole JI, Akeju B. Enterprise cybersecurity trends and threat evolution, advances and emerging research directions. Glob Multidiscip Perspect J. 2024;1(6):194-204. doi:10.54660/GMPJ.2024.1.6.194-204
205. Okoruwa PO, Fadayomi O, Akeju B, Edivri J, Ogbole JI, Abolaji TO. Conceptual model for privacy-centric security engineering in digital and cloud computing systems. Int J Sci Res Comput Sci Eng Inf Technol. 2020;7(5):371-89.
206. Olaogun BO, Adesuyi MO, Akomolafe O, Ndukwe VU, Sakyi JK. Regulatory compliance adaptation model for cross-border payment product redesign. Int J Adv Multidiscip Res Stud. 2023;3(1):1651-62. doi:10.62225/2583049X.2023.3.1.5279
207. Oluwo K, Dada T, Isiekwu CP. Transforming insurance underwriting with machine learning: A review and application cases. Int J Adv Multidiscip Res Stud. 2024;4(6):3008-19.
208. Ominyi M. Toward a conceptual framework for embedding privacy by design in corporate restructuring and transaction structuring. J Account Financ Manage. 2021;7(5):152-73. doi:10.56201/jafm.vol.7.no5.2021.p152.173
209. Ominyi M, Anichukwueze CC. A review of regulatory clearance frameworks for cross-border mergers and acquisitions in African financial markets. J Account Financ Manage. 2020;6(4):73-99. doi:10.56201/jafm.vol.6.no4.2020.pg73.99
210. Ominyi M, Anichukwueze CC. A systematic review of

anti-money laundering and anti-bribery compliance regimes across emerging and developed markets. Int J Econ Financ Manage. 2021;6(2):89-118. doi:10.56201/ijefm.v6.no2.2021.pg89.118
211. Ominyi M, Anichukwueze CC. A conceptual framework for harmonizing data protection compliance across GDPR, NDPR, and United States privacy regimes. Int J Soc Sci Manage Res. 2022;8(5):148-74. doi:10.56201/ijssmr.v8.no5.2022.pg148.174
212. Ominyi M, Anichukwueze CC. Reviewing greenhouse gas disclosure and ESG compliance obligations for energy and infrastructure enterprises under evolving regulatory regimes. Int J Econ Financ Manage. 2022;7(6):94-124. doi:10.56201/ijefm.v7.no6.2022.pg94.124
213. Ominyi M, Anichukwueze CC. A review of algorithmic accountability and model risk management approaches in financial services. J Account Financ Manage. 2023;9(12):218-38. doi:10.56201/jafm.v9.no12.2023.pg218.238
214. Ominyi M, Anichukwueze CC. Conceptualizing responsible AI governance structures for automated decision systems in regulated industries. Int J Soc Sci Manage Res. 2024;10(11):403-30. doi:10.56201/ijssmr.v10.no11.2024.pg.403.430
215. Ominyi M, Anichukwueze CC, Uzougbo NS. A systematic review of organizational readiness for the EU Artificial Intelligence Act in multinational enterprises. World J Innov Mod Technol. 2024;8(6):185-203. doi:10.56201/wjimt.v8.no6.2024.pg185.203
216. Oteri O, Edivri J. Applied performance optimization frameworks for managing high-demand enterprise technology programs and system rollouts. Gyanshauryam Int Sci Ref Res J. 2024;7(1):188-210.
217. Oyesiji SO, Nwakamma S, Ojukwu J. A conceptual framework for context-aware data curation in fine-tuning pipelines. Int J Eng Mod Technol. 2024;10(11):197-228. doi:10.56201/ijemt.v10.no11.2024.pg197.228
218. Pacheco C, García I, Reyes M. Requirements elicitation techniques: A systematic literature review based on the maturity of the techniques. IET Softw. 2018;12(4):365-78. doi:10.1049/iet-sen.2017.0144
219. Paleyes A, Urma RG, Lawrence ND. Challenges in deploying machine learning: A survey of case studies. ACM Comput Surv. 2022;55(6):1-29. doi:10.1145/3533378
220. Palomares C, Franch X, Quer C, Chatzipetrou P, López L, Gorschek T. The state-of-practice in requirements elicitation: An extended interview study at 12 companies. Requir Eng. 2021;26(2):273-99. doi:10.1007/s00766-020-00345-x
221. Parasuraman R, Riley V. Humans and automation: Use, misuse, disuse, abuse. Hum Factors. 1997;39(2):230-53. doi:10.1518/001872097778543886
222. Paul D, Cadle J, Eva M, Rollason C, Hunsley J. Business Analysis. 4th ed. Swindon: BCS, The Chartered Institute for IT; 2020.
223. Pipino LL, Lee YW, Wang RY. Data quality assessment. Commun ACM. 2002;45(4):211-8. doi:10.1145/505248.506010
224. Plattfaut R, Borghoff V, Godefroid M, Koch J, Trampler M, Coners A. The critical success factors for robotic process automation. Comput Ind. 2022;138:103646. doi:10.1016/j.compind.2022.103646
225. Pohl K. Requirements Engineering: Fundamentals, Principles, and Techniques. Berlin: Springer; 2010.
226. Polanyi M. The tacit dimension. Foreword by A. Sen. University of Chicago Press; 2009.
227. Porter AA, Votta LG, Basili VR. Comparing detection methods for software requirements inspections: A replicated experiment. IEEE Trans Softw Eng. 1995;21(6):563-75. doi:10.1109/32.391380
228. Rahimi M, Guo JLC, Kokaly S, Chechik M. Toward requirements specification for machine-learned components. In: 2019 IEEE 27th International Requirements Engineering Conference Workshops (REW); 2019. p. 241-4. doi:10.1109/REW.2019.00049
229. Raji ID, Smart A, White RN, Mitchell M, Gebru T, Hutchinson B, *et al*. Closing the AI accountability gap: Defining an end-to-end framework for internal algorithmic auditing. In: Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency; 2020. p. 33-44. doi:10.1145/3351095.3372873
230. Ramesh B, Jarke M. Toward reference models for requirements traceability. IEEE Trans Softw Eng. 2001;27(1):58-93. doi:10.1109/32.895989
231. Riegel N, Doerr J. A systematic literature review of requirements prioritization criteria. In: Fricker SA, Schneider K, editors. Requirements engineering: Foundation for software quality (REFSQ 2015). LNCS. Springer; 2015. p. 300-17. doi:10.1007/978-3-319-16101-3_22
232. Robertson J, Robertson S, Reed A. Mastering the Requirements Process. 4th ed. Boston: Addison-Wesley Professional; 2024.
233. Rudin C. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. Nat Mach Intell. 2019;1(5):206-15. doi:10.1038/s42256-019-0048-x
234. Rupp C, Pohl K. Requirements engineering fundamentals: A study guide for the Certified Professional for Requirements Engineering exam — Foundation level IREB compliant. 2nd ed. Rocky Nook; 2015.
235. Sambasivan N, Kapania S, Highfill H, Akrong D, Paritosh P, Aroyo L. "Everyone wants to do the model work, not the data work": Data cascades in high-stakes AI. In: Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems; 2021. p. 1-15. doi:10.1145/3411764.3445518
236. Sanderson C, Douglas D, Lu Q, Schleiger E, Whittle J, Lacey J, *et al*. AI ethics principles in practice: Perspectives of designers and developers. IEEE Trans Technol Soc. 2023;4(2):171-87. doi:10.1109/TTS.2023.3257303
237. Sanni JO, Atima ME, Attah A. Systematic review of attribution modeling methods resolving bias in multi-touch journeys. Shodhshauryam Int Sci Ref Res J. 2022;5(1):304-34. doi:10.32628/SHISRRJ247135
238. Sanni JO, Iwuanyanwu UA, Essien MA, Wedraogo L. Integrating blockchain-enabled smart contracts for transparent and verifiable marketing workflows. Int J Adv Multidiscip Res Stud. 2024;4(6):2891-906. doi:10.62225/2583049X.2024.4.6.5646
239. Sanni JO, Wedraogo L. Data-centric funnel optimization frameworks resolving revenue leakage in high-value services. Int J Adv Multidiscip Res Stud.

2024;4(6):2859-74. doi:10.62225/2583049X.2024.4.6.5641
240. Schneider J, Abraham R, Meske C, vom Brocke J. Artificial intelligence governance for businesses. Inf Syst Manag. 2023;40(3):229-49. doi:10.1080/10580530.2022.2085825
241. Sculley D, Holt G, Golovin D, Davydov E, Phillips T, Ebner D, *et al*. Hidden technical debt in machine learning systems. In: Advances in Neural Information Processing Systems 28 (NIPS 2015). Curran Associates; 2015.
242. Selbst AD, boyd d, Friedler SA, Venkatasubramanian S, Vertesi J. Fairness and abstraction in sociotechnical systems. In: Proceedings of the Conference on Fairness, Accountability, and Transparency; 2019. p. 59-68. doi:10.1145/3287560.3287598
243. Shneiderman B. Bridging the gap between ethics and practice: Guidelines for reliable, safe, and trustworthy human-centered AI systems. ACM Trans Interact Intell Syst. 2020;10(4):1-31. doi:10.1145/3419764
244. Skitka LJ, Mosier KL, Burdick M. Does automation bias decision-making? Int J Hum Comput Stud. 1999;51(5):991-1006. doi:10.1006/ijhc.1999.0252
245. Sommerville I, Sawyer P. Requirements Engineering: A Good Practice Guide. Chichester: John Wiley & Sons; 1997.
246. Spoletini P, Ferrari A, Bano M, Zowghi D, Gnesi S. Interview review: An empirical study on detecting ambiguities in requirements elicitation interviews. In: Requirements Engineering: Foundation for Software Quality (REFSQ 2018). LNCS Vol. 10753. Springer; 2018. p. 101-18. doi:10.1007/978-3-319-77243-1_7
247. Steidl M, Felderer M, Ramler R. The pipeline for the continuous development of artificial intelligence models: Current state of research and practice. J Syst Softw. 2023;199:111615. doi:10.1016/j.jss.2023.111615
248. Strong DM, Lee YW, Wang RY. Data quality in context. Commun ACM. 1997;40(5):103-10. doi:10.1145/253769.253804
249. Sunday EA, Omoegun GO, Essien MA, Oluokun OA. Transitioning from reactive to predictive maintenance in mechanical systems. Int J Sci Res Comput Sci Eng Inf Technol. 2020;6(6):425-47.
250. Sutcliffe A, Sawyer P. Requirements elicitation: Towards the unknown unknowns. In: 2013 21st IEEE International Requirements Engineering Conference (RE); 2013. p. 92-104. doi:10.1109/RE.2013.6636709
251. Syed R, Wynn MT. Robotic process automation: A review of the state-of-the-art. In: Handbook on business process management and digital transformation. Edward Elgar; 2024. p. 333-62. doi:10.4337/9781802206098.00024
252. Syed R, Suriadi S, Adams M, Bandara W, Leemans SJJ, Ouyang C, *et al*. Robotic process automation: Contemporary themes and challenges. Comput Ind. 2020;115:103162. doi:10.1016/j.compind.2019.103162
253. Teinemaa I, Dumas M, La Rosa M, Maggi FM. Outcome-oriented predictive process monitoring: Review and benchmark. ACM Trans Knowl Discov Data. 2019;13(2):1-57. doi:10.1145/3301300
254. Telford T, Thopalli K, Schaefer G, Polner A, Urbaniak T, Wright D. Automation with intelligence (2022 survey). Deloitte Insights; 2022.
255. Tjong SF, Berry DM. The design of SREE — A prototype potential ambiguity finder for requirements specifications and lessons learned. In: Requirements Engineering: Foundation for Software Quality (REFSQ 2013). LNCS Vol. 7830. Springer; 2013. p. 80-95. doi:10.1007/978-3-642-37422-7_6
256. Tonoyan A, Dada O, Ayivi-Donkor SS. Advances in supply chain resilience: Predictive models for vendor risk assessment and procurement cost optimization. Int J Soc Sci Manage Res. 2024;10(11):525-51. doi:10.56201/ijssmr.v10.no11.2024.pg.525.551
257. van den Broek E, Sergeeva A, Huysman M. When the machine meets the expert: An ethnography of developing AI for hiring. MIS Q. 2021;45(3):1557-80. doi:10.25300/misq/2021/16559
258. van der Aalst WMP. Process mining: Data science in action. 2nd ed. Berlin: Springer; 2016. doi:10.1007/978-3-662-49851-4
259. van der Aalst WMP, Bichler M, Heinzl A. Robotic process automation. Bus Inf Syst Eng. 2018;60(4):269-72. doi:10.1007/s12599-018-0542-4
260. van Lamsweerde A. Requirements Engineering: From System Goals to UML Models to Software Specifications. Chichester: John Wiley & Sons; 2009.
261. Veale M, Van Kleek M, Binns R. Fairness and accountability design needs for algorithmic support in high-stakes public sector decision-making. In: Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems; 2018. p. 1-14. doi:10.1145/3173574.3174014
262. Villamizar H, Escovedo T, Kalinowski M. Requirements engineering for machine learning: A systematic mapping study. In: 2021 47th Euromicro Conference on Software Engineering and Advanced Applications (SEAA); 2021. p. 29-36. doi:10.1109/SEAA53835.2021.00013
263. Villamizar H, Kalinowski M, Lopes H, Méndez D. Identifying concerns when specifying machine learning-enabled systems: A perspective-based approach. J Syst Softw. 2024;213:112053. doi:10.1016/j.jss.2024.112053
264. Vogelsang A, Borg M. Requirements engineering for machine learning: Perspectives from data scientists. In: 2019 IEEE 27th International Requirements Engineering Conference Workshops (REW); 2019. p. 245-51. doi:10.1109/REW.2019.00050
265. Wagner S, Méndez Fernández D, Felderer M, Vetrò A, Kalinowski M, Wieringa R, *et al*. Status quo in requirements engineering: A theory and a global family of surveys. ACM Trans Softw Eng Methodol. 2019;28(2):1-48. doi:10.1145/3306607
266. Wand Y, Wang RY. Anchoring data quality dimensions in ontological foundations. Commun ACM. 1996;39(11):86-95. doi:10.1145/240455.240479
267. Wang RY, Strong DM. Beyond accuracy: What data quality means to data consumers. J Manage Inf Syst. 1996;12(4):5-33. doi:10.1080/07421222.1996.11518099
268. Wanner J, Hofmann A, Fischer M, Imgrund F, Janiesch C, Geyer-Klingeberg J. Process selection in RPA projects: Towards a quantifiable method of decision making. In: Proceedings of the 40th International Conference on Information Systems (ICIS 2019); 2019. Munich, Germany.
269. Wedraogo L, Sanni JO. Machine learning models addressing uncertainty in cross-channel campaign performance forecasting accuracy. Int J Adv Multidiscip

Res Stud. 2024;4(6):2875-90. doi:10.62225/2583049X.2024.4.6.5648
270. Wellmann C, Stierle M, Dunzer S, Matzner M. A framework to evaluate the viability of robotic process automation for business process activities. In: Business process management: Blockchain and robotic process automation forum. LNBIP. Springer; 2020. p. 200-14. doi:10.1007/978-3-030-58779-6_14
271. Wewerka J, Reichert M. Robotic process automation: A systematic mapping study and classification framework. Enterp Inf Syst. 2021;17(2):1986862. doi:10.1080/17517575.2021.1986862
272. Whang SE, Roh Y, Song H, Lee JG. Data collection and quality challenges in deep learning: A data-centric AI perspective. VLDB J. 2023;32(4):791-813. doi:10.1007/s00778-022-00775-9
273. Wiegers K, Beatty J. Software Requirements. 3rd ed. Redmond, WA: Microsoft Press; 2013.
274. Wieringa M. What to account for when accounting for algorithms: A systematic literature review on algorithmic accountability. In: Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency; 2020. p. 1-18. doi:10.1145/3351095.3372833
275. Wilson WM, Rosenberg LH, Hyatt LE. Automated analysis of requirement specifications. In: Proceedings of the 19th International Conference on Software Engineering (ICSE '97); 1997. p. 161-71. doi:10.1145/253228.253258
276. Xu Y, Li M, Cui L, Huang S, Wei F, Zhou M. LayoutLM: Pre-training of text and layout for document image understanding. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining; 2020. p. 1192-200. doi:10.1145/3394486.3403172
277. Yang H, de Roeck A, Gervasi V, Willis A, Nuseibeh B. Analysing anaphoric ambiguity in natural language requirements. Requir Eng. 2011;16(3):163-89. doi:10.1007/s00766-011-0119-y
278. Zhao L, Alhoshan W, Ferrari A, Letsholo KJ, Ajagbe MA, Chioasca EV, *et al*. Natural language processing for requirements engineering: A systematic mapping study. ACM Comput Surv. 2021;54(3):1-41. doi:10.1145/3444689
279. Zowghi D, Coulin C. Requirements elicitation: A survey of techniques, approaches, and tools. In: Aurum A, Wohlin C, editors. Engineering and Managing Software Requirements. Berlin: Springer; 2005. p. 19-46. doi:10.1007/3-540-28244-0_2