



Information is All You Need: Requirements Engineering Quality Reframed

Downloaded from: <https://research.chalmers.se>, 2026-10-10 16:30 UTC

Citation for the original published paper (version of record):

Femmer, H., Frattini, J. (2026). Information is All You Need: Requirements Engineering Quality Reframed. Proceedings of the IEEE International Conference on Requirements Engineering: 518-525. <http://dx.doi.org/10.1109/RE68928.2026.00054>

N.B. When citing this work, cite the original published paper.

© 2026 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, or reuse of any copyrighted component of this work in other works.

Information is all you need: Requirements Engineering Quality Reframed

Henning Femmer

South Westphalia University of Applied Sciences
Hagen, Germany
femmer.henning@fh-swf.de

Julian Frattini

Chalmers and University of Gothenburg
Gothenburg, Sweden
julian.frattini@chalmers.se

Abstract—Why do low-quality requirements specifications sometimes produce functional software while some projects with high-quality specifications fail? A common answer to this anomaly is that *it depends on the context*, which is neither expressive nor actionable. To move beyond this vague appeal to context, this vision proposes a novel holistic theory of requirements engineering (RE) quality. This theory models how information particles, i.e., discrete pieces of domain knowledge, are transferred between roles and artifacts. Since the RE process is ultimately an information transfer, holistic RE quality depends on the properties of information flow, i.e., how effectively and efficiently information is transferred from sources (like stakeholders) to targets (like developers and testers), uniting both artifact- and process-based perspectives on RE quality. In an exemplary simulation of the theory we illustrate why a high-quality specification gets bypassed in an agile context, thereby demonstrating that a simulation can provide actionable insights into calibrating the RE process to optimize the information flow. Beyond organizational applications, we envision that the theory can serve as a coherent theoretical framework for understanding the success or failure of RE processes and artifacts.

Index Terms—requirements engineering, quality, context, theory, information, information flow

I. INTRODUCTION

In the software engineering (SE) process, various studies stress the relevance of requirements engineering (RE) for project success [1]. Against many claims, challenges in RE persist regardless of the adopted process model (e.g., waterfall or agile) [1]. These challenges manifest when root causes such as incomplete or hidden requirements [1] affect subsequent activities that rely on these requirements as input [2], e.g., implementing or testing features. As a consequence, various RE quality models have postulated how (not) to write requirements specifications [3]–[6] to avoid such impacts.

However, the adequacy of these quality models has been drawn into question [7]. Despite plausible arguments, several factors of requirements quality experience contradictory evidence, where some empirical studies claim that factors such as *passive voice* [8] or *consistently numbered steps in use cases* [6] matter, while others disagree [9], [10]. The RE community seems content with the consensus to explain these differences with the ominous influence of context [7]. Deflecting the explanation to context is not helpful, as it does not provide reliable decision support for addressing these differences.

These differences resisting explanation by existing quality theories constitute an anomaly requiring a paradigm shift [11]. To this end, we propose an alternative paradigm to the current artifact-centric perspective to RE quality. Consistent with information theory [12], we frame RE as a means of information transfer: Information in the form of requirements needs to be transferred from its sources (e.g., stakeholders, competitor analyses, reuse databases) to its targets just beyond the border of RE (e.g., designers, developers, and testers). RE quality can then be understood as how effectively and efficiently information flows, regardless of its form (i.e., written or verbal), offering a holistic perspective on quality that goes beyond artifact- and process-centric theories.

Our vision for such an information-centric theory of RE quality outlines an avenue of research capable of resolving the aforementioned anomalies. The resulting theory can both explain them, and be operationalized to mitigate them. In the scope of this vision paper, we make the following contributions:

- A novel paradigm framing RE quality from a holistic perspective, focusing on the flow of information particles.
- A simulation of the application of this paradigm to demonstrate its envisioned use case (available online [13]).

II. THEORETICAL FOUNDATIONS

In this section, we introduce the relevant background underpinning this vision. Our presentation follows the epistemological framework proposed by Kuhn [11], according to which research is governed by predominant paradigms that persist until accumulated anomalies can no longer be reconciled with them, triggering a shift toward a more powerful theory.

Section II-A elaborates the current paradigm of quality in RE, and Section II-B provides evidence for current anomalies resisting explanation. Finally, we introduce related work on information flow in Section II-C.

A. Requirements Quality: Current Paradigms

Today, we understand RE either from a perspective of (physical) artifacts or from a perspective of processes and techniques. Most approaches have their own definition of quality without a higher theoretical foundation.

Most fundamentally, early works [14]–[17] iteratively defined RE quality as relationships between different concepts. Quality is ultimately understood as a relationship between the artifact and the domain (*semantic quality*), the RE language (*syntactic quality*) and the audience interpretation (*pragmatic quality*). Further qualities include the consistency of the audience interpretation (*social quality*) and various types of appropriateness, e.g., the choice of language for the intended audience [17].

Common quality models used in standards—such as ISO/IEC/IEEE 29148:2018 [3] or the IREB CPRE FL [4]—focus on sets of generalized quality factors (such as completeness, unambiguity, etc.), identified and specified by experts in the field. The way these factors are postulated, they usually do not provide falsifiable reasoning about the consequences of not adhering to the standard, and accordingly, are presented without empirical underpinning. To amend this, follow-up research postulated artifact-centric theories that determine requirements quality based on artifacts’ impact on subsequent activities [2], [7], [18], leading to falsifiable quality factors: An artifact without negative impact on subsequent activities cannot be considered defective.

More specific definitions for quality exist for specific artifact types such as user stories [19]–[21], use cases [6], vision videos [22], or UML models [23]. Similarly, authors came up with quality criteria for different techniques, e.g., for interviewing [24], or conducting workshops [25].

B. Persisting Anomalies

The artifact-centric quality theories presented in Section II-A are based on plausible arguments, yet *anomalies*—observable phenomena that resist explanation by these theories [11]—still persist. Particularly, when empirical studies contribute evidence to the theories, contradictions arise.

For example, several prior studies have suggested that textual requirements using passive voice are defective [26] because it “can lead to ambiguous interpretations” [27]. Femmer et al. [8] observed in a controlled experiment that passive voice in textual requirements affects the reader’s ability to relate domain concepts to each other supporting this claim. However, a later replication did not confirm this effect [28]. Even more contradictory, Krisch and Houdek arrived at the opposite conclusion during a case study, i.e., that practitioners consider most occurrences of passive voice harmless because “[i]nformation can be drawn through context” [9].

Similarly, catalogs of alleged requirements quality defects see varying support from empirical studies. For example, the 7 C’s by Phalp et al. [6] contain writing rules with a strong foundation in text comprehension theory, but evidence from an empirical case study shows only partial support for them [10].

A *context factor* is often cited as a kind of *deus ex machina* to explain why quality sometimes does (not) matter [7], [9]: An occurrence of a specific quality factor may be harmful in one context but irrelevant in another. While plausible, this coarse explanation is unhelpful to properly understand or act on the alleged defect, though, as it is limited to claiming that

context affects quality [7], [29], but not *how*. This remains a limitation of existing RE quality theories.

C. Related Work on Information Flow in Companies

Our theory—introduced in Section III—is based on the idea that requirements are information flowing through a socio-economic system. Although we could not find a fitting theory that can be reused in the RE context, researchers in the fields of information systems and economics have studied concepts of information and information flow from different angles.

In times of information-based organizations [30], the organizations’ task is to balance information supply and information demand, i.e., a dynamic system that requires continuous adjustment [31]. In this context, information is described as a specific good: immaterial, useful, valuable, mutable, reproducible, and transportable at the speed of light [31].

Durugbo et al. have summarized various approaches for modeling the flow of information in organizations [12]. Diagrammatic modeling allows organizations to better understand their own information flow on a high level [32] and falls into three categories: pictorial representations [33], graph representations [34], [35], and matrix representations [36]. Central to diagrammatic information flow modeling are often the nodes (i.e., between whom the information flows) rather than the actual information itself.

The alternative mathematical modeling approaches serve two main purposes: flow analysis and organizational analysis [12]. Krovi et al. propose a “parameter-based guiding framework of information flow to manage organizational processes” [37] drawing inspiration from fluid flow with concepts like velocity and viscosity, but remain on a descriptive level. Lin and Cheng [38] formalize organizations as interrelated parts whose behavior is defined by mathematical functions, but focuses rather on the overall relationship flow within an organization. Ben Arie and Pollatscheck [39] mathematically describe information overload in hierarchical organizations, modeling organizations as trees mainly with interactions along the hierarchy.

D. Related Work on Information Flow in RE

Winkler [40] analyzes the information flow (only) between documents, acknowledging that people do not always keep the tangible documents consistent with the intangible knowledge.

Schneider et al. [41] model flow of information in RE using various techniques, such as UML diagrams or data flow diagrams. The paper distinguishes between solid and fluid representations, which closely resembles our tangible and intangible information storage concepts. They also introduce the FLOW notation to explicitly model both types. While their work focuses more on flow and visualization, our work has a focus on the concept of the information itself, its simulation, and what this theoretic lens implies for RE quality.

Despite their relatedness, none of the existing approaches matches our use case. Diagrammatic modeling focuses on the nodes between which information flows whereas our perspective focuses on the flowing information itself. Mathematical

modeling approaches are either non-formal [37] or focus on specific aspects [38], [39]

III. AN INFORMATION-CENTRIC PARADIGM OF RE QUALITY

The anomalies presented in Section II-B resist explanation by the current paradigms of quality in RE presented in Section II-A. To resolve this crisis [11], we propose a novel paradigm of RE quality based on information flow introduced in Section II-C. First, we elaborate a deliberately simplified yet plausible RE scenario in Section III-A to illustrate the concepts then introduced in Section III-B. With these concepts, we resolve the described anomaly in Section III-C.

A. The Anomaly

Assume the following plausible, albeit simplified scenario while observing software developers in a small project team. A skilled requirements engineer has created a high-quality requirements specification that stakeholders have approved. It is comprehensive, complete, and consistent, i.e., it meets all criteria commonly demanded of a requirements specification [3], [4], [6]. Despite this, the developers consistently bypass it and talk directly to the customer instead.

This behavior is inexplicable by existing requirements quality theories: Adhering to requirements writing guidelines [3], [4], [6] should maximize the usability of the requirements specification according to artifact-centric quality theories [18]. Hence, these theories do not rationally explain the divergent behavior. The most common answer for dodging this anomaly is deferring to dependency on the context [29]. But even reference to existing catalogs of context factors [42] do not sufficiently resolve the anomaly.

B. Core Concepts of an Information-Centric RE Paradigm

The reason for this anomaly is rooted in the structure of the existing theories itself. Artifact-centric RE quality theories consider *requirements artifacts* (e.g., requirements specifications, user stories, or backlogs) as first-class citizens. But requirements artifacts are just one means-to-an-end, i.e., one form of how requirements can be stored and transferred, just as RE itself is a means-to-an-end [4]. This end, i.e., the ultimate purpose of RE, is gathering requirements from stakeholders and other sources, and providing them to the SE process where a solution to these requirements is developed. In the following, we frame this purpose of RE from an information-centric perspective to arrive at a holistic paradigm of RE quality.

a) *Information particles and flow*: Requirements are information, particularly information about the problem-space [43], [44]. A single requirement, i.e., a single, atomic piece of information, can be considered an **information particle** [37]. For example, a need expressed by a customer about a certain feature (e.g., the feature request to export data in CSV format, see Figure 1) is one such information particle. The semantic information, syntactically and lexically refined during the RE process, later informs development about the features to implement.

Information particles are located in **information storages**. Contrary to existing theories [2], [7], these storages can both be tangible (e.g., requirements specifications, user story backlogs, basically any RE artifact) or intangible (e.g., a stakeholder's prior domain knowledge or a developer's memory), as shown in Figure 1. The capacity and retention of tangible and intangible storages may differ, but they both serve the same purpose from an information-centric perspective, i.e., storing information particles.

Information particles can move between information storages during **information transfers**. Any activity that obtains information particles from one storage and moves it to another qualifies as an information transfer. For example, while *interviewing*, information particles will transfer from a customer to the requirements engineer (top left in Figure 1). Afterwards, while *documenting*, this requirements engineer transfers the obtained information particles to a requirements specification. How much and how quickly information can be transferred from one storage to another depends on factors like task complexity and skill [45].

Overall, the RE phase is a sequence of information transfers that move information particles from source information storages (e.g., stakeholders, competitor analyses, or field studies) to target information storages (e.g., software architects, developers, or testers), potentially via multiple different paths. Target information storages are those located just outside the boundary of RE, i.e., where problem-space information (i.e., requirements) are used to produce solution-space information (e.g., architecture, source code). For example, the *developer* in Figure 1 is an information storage outside of boundary of RE, as they use the problem-space information to implement the required features. Overall, the RE phase is—from an information-centric perspective—simply an **information flow** [37].

b) RE quality in the information-centric paradigm:

Given this framing, the *quality of RE* becomes the **effectiveness and efficiency** of the information flow, i.e., how much information arrives at the boundary of RE and how much resources (e.g., time) it costs. This definition subsumes previous requirements quality theories. In particular, the previous artifact-centric requirements quality theories are completely encompassed. Artifact-centric theories explain quality via the impact of requirements artifacts on subsequent activities [2], which maps to the effectiveness and efficiency of information transfers originating from tangible information storages. However, our proposed paradigm embeds this relationship in a larger-scale, holistic perspective where these effects co-exists with information transfers originating from intangible information storages.

C. Resolving the Anomaly

The paradigm elaborated in Section III-B can resolve the anomaly from Section III-A. In this scenario, the developer—i.e., an information storage just beyond the border of RE—requires information for the solution-design and has two ways of obtaining that information. They can either have the infor-

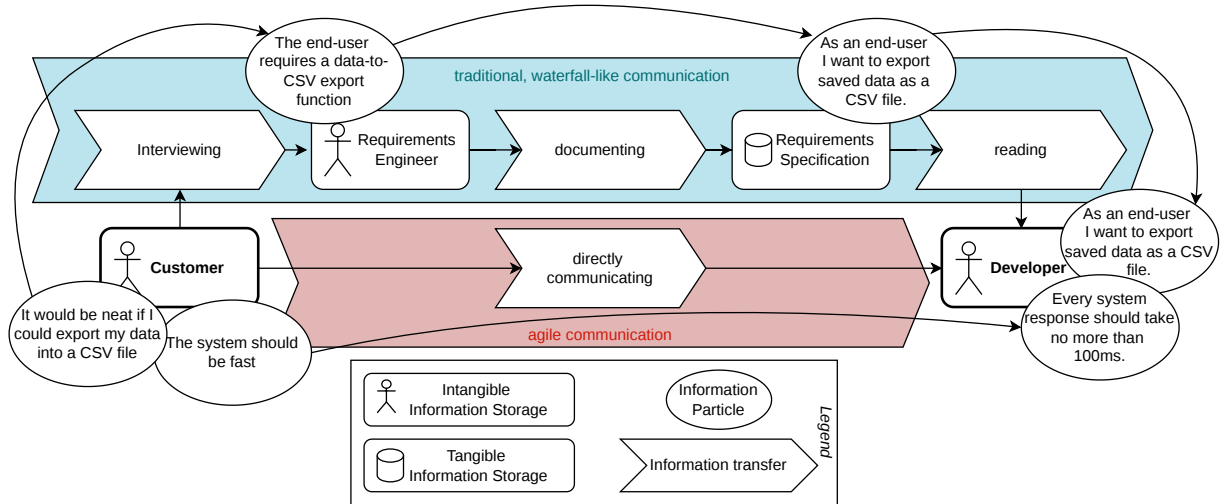


Fig. 1. Synthetic scenario showing two alternative paths for information flow

mation transferred by *reading* the (high-quality) requirements specification, a tangible information storage (via the top, blue route in Figure 1), or they can *directly communicate* with a stakeholder, an intangible information storage (via the bottom, red route in Figure 1). From the perspective of the developer, whichever way is more effective (i.e., yields more information) and efficient (i.e., requires less resources like time) is preferable. So, regardless of the organization's official process model, if the developer can obtain more information more quickly by directly communicating with the customer, the overall RE quality is higher if the developer bypasses the requirements specification. Alternatively, the customer may be more difficult to access or the developer may struggle to transfer sufficient information from the customer (e.g., because they are not trained in requirements elicitation). In this case, direct communication yields less information particles and/or requires more effort for reaching an equivalent yield. Consequently, the theory will predict that the developers most likely will not bypass the requirements specification. Overall, information follows the path of least resistance, which is how the information-centric paradigm resolves the anomaly.

However, the resolution is currently only narrative, i.e., based on plausible explanations. This does not suffice for evidence-based decisions, e.g., at what point of resistance the information will flow through a different path. To enable such decisions, the paradigm requires formalization.

IV. FORMALIZATION

Organizations could apply the paradigm of framing requirements as information particles and modeling their RE process as an information flow. This would allow them to (1) explain previously counter-intuitive phenomena like the one presented in Section III-A, and (2) predict how the information flow would change when calibrating the RE process, e.g., by allocating more time or higher-skilled engineers to a certain

transfer. Jointly, the application of the paradigm would serve capturing the concept of RE quality by relating RE process properties to the output of information.

Despite the existing literature on information flow modeling, no single approach directly maps to the RE context and fulfills the aforementioned goals. We do not aim to elaborate a conclusive and definitive approach in the scope of this study, but present potential forms in Section IV-A and illustrate the application of one in Section IV-B to illustrate its usefulness.

A. Approaches for Formalizing RE Information Flows

Table I contrasts two potential forms according to their notation, advantages, and disadvantages. The subsequent paragraphs briefly discuss each of them.

a) *Information particles as discrete, concrete items (type I)*: Most concretely, a formalization could describe the flow of concrete information particles directly just as exemplified in Section III and Figure 1. Information particles would represent individual requirements, both functional (e.g., "As an end-user I want to export saved data as a CSV file.") and non-functional (e.g., "Every system response should take no more than 100ms."). The formalism would model the flow of these particles from a set of information storages (e.g., customers, requirements reuse databases, or competitor analyses) through intermediate storages (e.g., requirements engineers, requirements specifications) until reaching the border of the RE process (e.g., a developer or a solution specification). First experiments evaluating whether information flows could be simulated with agentic AI can be found in [46]. However, for realistic projects such an information flow could currently only be determined in hindsight, i.e., once the RE phase has completed and the information particles are known. Retrospective case studies particular to one project would support a post-mortem analysis of identifying bottlenecks and understanding properties of the specific information flow.

TABLE I
COMPARISON OF TYPES OF FORMALIZATIONS OF RE INFORMATION FLOWS

Type	Advantages	Disadvantages	Use Cases
Type I: discrete, concrete particles	• High granularity and specificity • Directly empirically grounded	• Only retrospective • Low generalizability	• Retrospective analysis (e.g., identifying bottlenecks) • Modeling at semantic level (e.g. introduction of inconsistencies)
Type II: continuous distributions	• High generalizability	• Requires extensive empirical evidence to construct • Implies several mathematical assumptions	• Identification of optimal flows • Prospective analysis (e.g., simulating change)

b) *Information flow as a continuous distribution (type II):*

To allow not only retrospective but also prospective analysis (e.g., for decision making), discrete, concrete information particles need be abstracted to a continuous quantity. This continuous quantity would represent the *level of information*, reaching from 0 (i.e., no information at all) to 1 (i.e., all relevant information), and be described in continuous distributions. For example, abstracting from multiple cases of type I formalizations, organizations can estimate *what fraction* of the total information certain stakeholders usually contain rather than which exact information particles (which differ from project to project). Information transfers are then defined as functions describing how much information—on average—moves from one storage to another depending on certain attributes like complexity or skill. For example,—again abstracting from multiple type I cases—organizations can estimate the complexity and resource-consumption of recurring transfers like *interviewing* and *documenting* as well as their input-output relation of information. As the more abstract formalization type, it would be difficult to model phenomena specific to semantic properties (e.g., requirements overlap or conflict). On the other hand, however, this approach offers the greatest transferability of insights to other contexts. Abstracting one type II formalization from multiple type I formalizations produces a generic description of the RE phase as an information flow usable for simulation and decision-making.

B. *Exemplary Formalization*

To make the potential use case of applying the paradigm more tangible, we specify a formalization of the aforementioned type II (information as a continuous quantity) and simulate its use. We do not claim that this exemplary formalization is optimal for realizing type II formalizations, but only use it for demonstration purposes. As an overall scenario, we formalize the phenomenon described in Section III-A, i.e., the counter-intuitive situation that a well-specified requirements artifact will be used in one context but bypassed in another. The full details, source code, and figures can be found in our replication package [13].

We model the information flow with two elements:

- 1) **Information storage**, defined by the level of information $i \in [0, 1]$ that it contains

- 2) **Information transfer**, defined by the level of input information $i \in [0, 1]$, the complexity level of the transfer activity $c \in \mathbb{R}^+$, the skill of the person executing the transfer $s \in \mathbb{R}^+$, the maximum time needed to completely exhaustively complete the transfer $t_{max} \in \mathbb{R}^+$, and the actual time spent executing the transfer $t \in \mathbb{R}^+$

We realize an information transfer f as the cumulative probability distribution function of the beta distribution (CDF_β):

$$f(i, c, s, t_{max}, t) = \text{CDF}_\beta \left(\alpha = s, \beta = c, x = \frac{t}{t_{max}} \right) \cdot i$$

The CDF_β has several properties that make it eligible to represent one information transfer. Firstly, it produces an output between 0 and 1, which fits the quantity of *level of information*. Secondly, its shape is defined by two parameters α and β . Higher values of α move the probability mass towards the left, i.e., increase the information output. We define α with a factor representing a positive influence: The *skill* of a person is generally considered to improve the effectiveness of a task [45]. For example, a highly skilled requirements engineer will be able to extract more information while interviewing than an engineer with less skill. Vice versa, higher values of β move the probability mass towards the left, i.e., decrease the information output. Similarly, we define β with a factor representing a negative influence, namely the *complexity* of a task. The more complex a task, the more difficult it is to transfer information through it [45]. Skill and task complexity are arbitrarily chosen, but plausible and recurring context factors moderating SE phenomena [45]. The relationship between the invested time t and the maximum time t_{max} determines the investment of resources into a task. Spending the maximum amount of time on a task results in the output of all the input information. This is achieved by the factor $\cdot i$, which enforces that the maximum level of information produced by a task is the level of information provided as an input. In other words: an information transfer can only lose existing, but not produce new information.

Figure 2 visualizes this CDF_β for four separate configurations, contrasting how high and low skill ($s \in \{1; 5\}$) and high and low complexity ($c \in \{2; 6\}$) affect the information i a transfer produces (on the y-axis) given the invested time t (on the x-axis). When investing $t = 2.5h$ (red vertical line) into the transfer, the resulting information output (grey horizontal line)

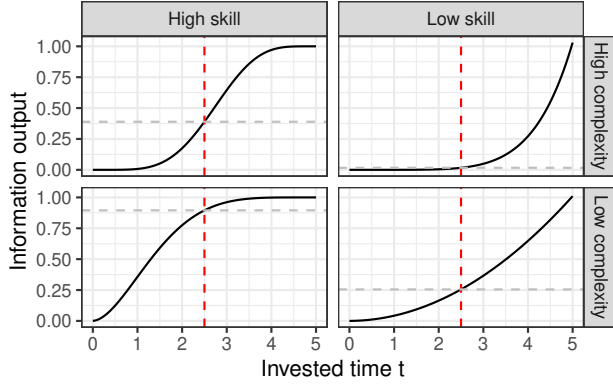


Fig. 2. CDF_{β} for four different configurations

would vastly differ depending on the configuration, ranging from 0.016 (high-complexity transfer performed with low skill, top right) to 0.895 (low-complexity transfer performed with high skill, bottom left).

Figure 3 visualizes the application of the formalization to the scenario described in Section III-A. It contains both information storages and transfers with their respective configuration. Each information transfer is characterized by the CDF_{β} -based function mapping input to output information. In essence, the figure shows two paths of how the information stored in one customer ($i = 0.8$) can flow to a developer:

- 1) The **traditional path**, where information is transferred
 - (1) to the requirements engineer through *interviewing*,
 - (2) then to a specification when *documenting* it, and (3) finally to the developer *reading* this specification.
- 2) The **agile path**, there information is immediately transferred to the developer through *directly communicating*.

The simulation in Figure 3 configures each transfer activity (interviewing, documenting, reading, and directly communicating) with arbitrary values for complexity c , skill s , time spent t , and maximum time for the activity t_{max} . For example, *interviewing* is considered a more complex activity ($c = 6$) than *reading* a specification ($c = 2$), and the requirements engineer has a higher skill *interviewing* ($s = 5$) than the developer *directly communicating* with the customer ($s = 3$). We compare the two flows across two scenarios, a waterfall-like and an agile process. The difference in the scenarios is operationalized in the complexity of the *directly communicating* task from a $c = 12$ (high complexity) to a $c = 5$ (medium complexity). The latter reflects that in an agile scenario, customers are typically more accessible and their feedback is easier to acquire. This configuration changes the degree of information transferred in the *directly communicating* task, as visualized in the bottom middle graph of Figure 3 comparing the agile and waterfall information transfer.

Executing the simulation explains the anomaly described in Section III-A. In the waterfall-like scenario, where directly communicating with the customer is difficult, the traditional path of transferring information produces more information

for the developer than the agile path ($0.59 > 0.24$). However, when the customer becomes more accessible to the developer (when the complexity of *directly communicating* drops below 6), this situation is reversed ($0.59 < 0.61$). As a consequence, a developer may bypass the requirements specification despite its high quality, simply because information experiences less resistance on the alternative path to the target.

This particular resolution determines the flow of information solely based on effectiveness (i.e., on the amount of information arriving at the target). Determining the flow based on efficiency (i.e., taking cost and resources into account) works similarly.

V. DISCUSSION

In the following, we discuss opportunities, limitations and challenges of the proposed theory.

A. Opportunities

The information-centric paradigm introduced in Section III provides a foundational perspective that can shift how we empirically study and evaluate quality in RE. Formalizing the information flow in an organization gives its management the ability to better understand its process and simulate the effect of changes. This way, an organization can determine the effect of calibrating the RE process. Connecting this to the example in Section IV-B, this could mean exploring which information transfer would improve the overall information output given additional time, reducing its complexity, or upskilling the responsible people.

B. Limitations

a) *Current limitations and possible extensions of the paradigm*: The paradigm elaborated in Section III contains but the fundamental concepts necessary to resolve the previously described anomalies. However, the information-centric paradigm of RE allows framing several other properties of the RE phase, including:

- **Decay and retention**: Information particles can lose value over time when the system or context changes [31]. Particularly, intangible information storages may lose information particles more quickly, which may affect the effectiveness of a transfer path. For example, people forget information or participants with intangible domain knowledge are no longer accessible.
- **Value and cost**: The value of a particle may depend on its contribution to the overall body of knowledge, similar to Shannon's information [47]. Modeling particle value by how distinct it is from an existing set of particles and how few information storages contain it could approximate its contribution to the RE process. Furthermore, differentiation of cost, e.g. cost of information acquisition, transformation, delay, or correction, is necessary.
- **Correctness, errors, misunderstandings**: In our model, the correct, required information is only modeled as present or absent. However, an incorrect or inconsistent information might have a different effect than a missing

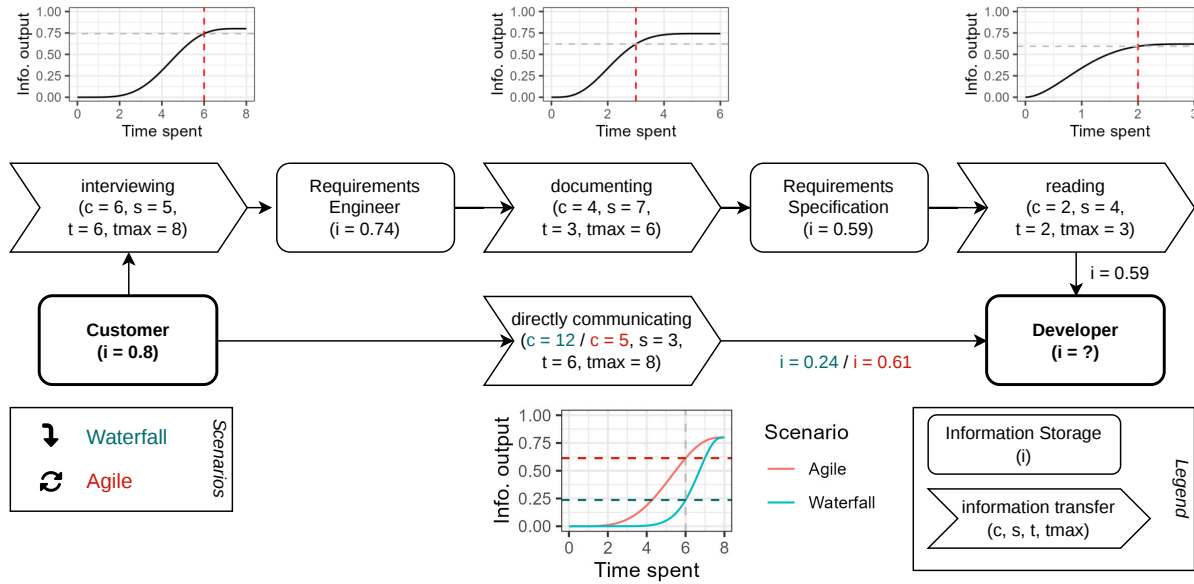


Fig. 3. Demonstration of Applying the Formalism

information. These implications—in particular the risks and consequences of introducing such defects—are not yet included in the presented work.

- **Information merge:** Information storages may receive information particles from multiple sources. For example, a requirements engineer may combine information particles elicited during an interview with particles from their domain knowledge. This may result in conflicts, which could be modeled in type I formalizations.
- **Prior knowledge:** Considering prior domain knowledge as information particles located in an intangible information storage implies a common information merge with other transfers. We need more research on understanding this relation in depth as a common instance in RE.
- **Iterations:** The same information storages may be involved in different information transfers at different points in time. As in agile development, iterative feedback cycles may help transfer more information from customers by presenting them the current solution space [43].
- **Simulation over time:** We expect that, in future work, type-I simulations would also allow to understand the flow of information over time.
- **Creativity:** Similarly, the RE phase may also unveil previously non-existent requirements by combining several information particles from different sources.

b) Limitations of the formalization: Every type of formalization approach presented in Section IV-A is inherently limited by the set of assumptions used to frame its syntax. For example, the presented demonstration in Section IV-B models information transfers with a specific function (here: CDF_{β}) and a specific set of parameters. Once again, we emphasize that the current demonstration does not make any claim of

representing the real world, it only serves to demonstrate the potential application.

C. Challenges

Current limitations include the difficulty of collecting empirical data that can be used to inductively implement the formalizations. For a scenario like Section IV-B to work and a formalization to be useful, its configurations (like the complexity- and skill-values) need to reflect real-world information storages and transfers. This implies three steps:

- 1) Identifying the factors that govern information flow in each information transfer step
- 2) Collecting data about these factors and their effect on the information flow in type I formalizations
- 3) Abstracting a type II formalization from these individual cases that models a generic RE process

While requiring considerable effort, we envision that such a path would allow operationalizing the envisioned RE quality paradigm in practice. Additionally, implementing the information-centric perspective in practice allows to validate the proposed paradigm empirically.

VI. CONCLUSION

Requirements are information, and the purpose of RE is to transfer that information from its sources (e.g., stakeholders) to its targets (e.g., developers and testers). As such, the effectiveness and efficiency of this information flow constitutes RE quality. Current quality theories of RE have a too narrow, often artifact-centric scope, which produces unexplainable anomalies like developers bypassing high-quality requirements specifications. Consequently, we propose a paradigm shift in RE quality theories, reframing it from the perspective of

information particles and their flow through the RE process. We advocate for using this paradigm as a basis for RE quality improvements in research and practice. We envision that this perspective has the potential to holistically frame RE quality and allow organizations to understand and optimize their RE process in the long-term.

Immediate future work is to empirically track information flow in companies between tangible and intangible information sources. Furthermore, the model requires formalizing of existing processing steps in RE, such as requirements refinement and other aspects discussed in the limitations section.

Data Availability Statement: All our data, scripts, and figures are available online [13].

REFERENCES

- [1] D. M. Fernández *et al.*, “Naming the pain in requirements engineering: Contemporary problems, causes, and effects in practice,” *Empirical software engineering*, vol. 22, no. 5, pp. 2298–2338, 2017.
- [2] H. Femmer, J. Mund, and D. M. Fernández, “It’s the activities, stupid! a new perspective on re quality,” in *International Workshop on Requirements Engineering and Testing*, 2015, pp. 13–19.
- [3] ISO/IEC/IEEE International Standard, “Systems and software engineering—life cycle processes—requirements engineering,” 2018.
- [4] S. Bühne, M. Glinz, H. van Loenhoud, and S. Staal, “CPRE Foundation Level - Syllabus – v.3.2.0,” 2024.
- [5] H. Femmer, D. M. Fernández, S. Wagner, and S. Eder, “Rapid quality assurance with requirements smells,” *Journal of Systems and Software*, vol. 123, pp. 190–213, 2017.
- [6] K. T. Phalp, J. Vincent, and K. Cox, “Assessing the quality of use case descriptions,” *Software Quality Journal*, vol. 15, pp. 69–97, 2007.
- [7] J. Frattini, L. Montgomery, J. Fischbach, D. Mendez, D. Fucci, and M. Unterkalmsteiner, “Requirements quality research: a harmonized theory, evaluation, and roadmap,” *Requirements Engineering*, 2023.
- [8] H. Femmer, J. Kučera, and A. Vetrò, “On the impact of passive voice requirements on domain modelling,” in *ESEM*, 2014.
- [9] J. Krisch and F. Houdek, “The myth of bad passive voice and weak words an empirical investigation in the automotive industry,” in *RE*, 2015, pp. 344–351.
- [10] J. Frattini and A. Frattini, “Adopting use case descriptions for requirements specification: an industrial case study,” in *RE*, 2025, pp. 179–191.
- [11] T. S. Kuhn, *The Structure of Scientific Revolutions: 50th Anniversary Edition*. Chicago, IL: University of Chicago Press, 2012, 4th edition.
- [12] C. Durugbo, A. Tiwari, and J. R. Alcock, “Modelling information flow for organisations: A review of approaches and future challenges,” *Int’l Journal of Information Management*, vol. 33, no. 3, pp. 597–610, 2013.
- [13] H. Femmer and J. Frattini, “Replication package,” <https://doi.org/10.5281/zenodo.20647995>, last accessed 2026-06-11.
- [14] K. Pohl, “The three dimensions of requirements engineering: a framework and its applications,” *Information systems*, vol. 19, no. 3, pp. 243–258, 1994.
- [15] O. I. Lindland, G. Sindre, and A. Solvberg, “Understanding quality in conceptual modeling,” *IEEE Software*, vol. 11, no. 2, pp. 42–49, 1994.
- [16] J. Krogstie, O. I. Lindland, and G. Sindre, “Towards a deeper understanding of quality in requirements engineering,” in *CAiSE*, 1995.
- [17] J. Krogstie, “Integrating the understanding of quality in requirements specification and conceptual modeling,” *ACM SIGSOFT Software Engineering Notes*, vol. 23, no. 1, pp. 86–91, 1998.
- [18] H. Femmer and A. Vogelsang, “Requirements quality is quality in use,” *IEEE Software*, vol. 36, no. 3, pp. 83–91, 2018.
- [19] R. Jeffries, “Essential xp: Card, conversation, confirmation,” *XP Magazine*, vol. 30, 2001.
- [20] M. Cohn, *User stories applied: For agile software development*. Addison-Wesley Professional, 2004.
- [21] G. Lucassen, F. Dalpiaz, J. M. E. van der Werf, and S. Brinkkemper, “Improving agile requirements: the quality user story framework and tool,” *Requirements engineering*, vol. 21, no. 3, pp. 383–403, 2016.
- [22] O. Karas, K. Schneider, and S. A. Fricker, “Representing software project vision by means of video: A quality model for vision videos,” *Journal of Systems and Software*, vol. 162, p. 110479, 2020.
- [23] C. F. Lange, “Improving the quality of uml models in practice,” in *Proceedings of the 28th International Conference on Software Engineering*, 2006, pp. 993–996.
- [24] M. Bano, D. Zowghi, A. Ferrari, P. Spoletini, and B. Donati, “Learning from mistakes: An empirical study of elicitation interviews performed by novices,” in *RE*, 2018, pp. 182–193.
- [25] E. Gottesdiener, *Requirements by collaboration: workshops for defining needs*. Addison-Wesley Professional, 2002.
- [26] E. Parra, C. Dimou, J. Llorens, V. Moreno, and A. Fraga, “A methodology for the classification of quality of requirements using machine learning techniques,” *Information and Software Technology*, vol. 67, pp. 180–195, 2015.
- [27] A. Ferrari, G. Gori, B. Rosadini, I. Trotta, S. Bacherini, A. Fantechi, and S. Gnesi, “Detecting requirements defects with nlp patterns: an industrial experience in the railway domain,” *Empirical Software Engineering*, vol. 23, no. 6, pp. 3684–3733, 2018.
- [28] J. Frattini, D. Fucci, R. Torkar, L. Montgomery, M. Unterkalmsteiner, J. Fischbach, and D. Mendez, “Applying bayesian data analysis for causal inference about requirements quality: a controlled experiment,” *Empirical Software Engineering*, vol. 30, no. 1, p. 29, 2025.
- [29] J. Mund, D. M. Fernandez, H. Femmer, and J. Eckhardt, “Does quality of requirements specifications matter? combined results of two empirical studies,” in *ESEM*, 2015.
- [30] P. F. Drucker, “The coming of the new organization,” *Harvard business review*, 1988.
- [31] H. Krcmar, “Informationsmanagement,” in *Informationsmanagement*. Springer, 2015, pp. 85–111.
- [32] R. Juric and J. Kuljis, “Engineering requirements through use cases in complex business environment,” *Requirements Engineering*, vol. 4, no. 2, pp. 65–76, 1999.
- [33] M. H. Burstein and D. E. Diller, “A framework for dynamic information flow in mixed-initiative human/agent organizations,” *Applied Intelligence*, vol. 20, no. 3, pp. 283–298, 2004.
- [34] C. D. Feinstein and P. A. Morris, “Information tree: A model of information flow in complex organizations,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 18, no. 3, pp. 390–401, 1988.
- [35] D. Braha and Y. Bar-Yam, “The statistical mechanics of complex product development: Empirical and analytical results,” *Management science*, vol. 53, no. 7, pp. 1127–1145, 2007.
- [36] L. Al-Hakim, “Modelling information flow for surgery management process,” *International Journal of Information Quality*, vol. 2, no. 1, pp. 60–74, 2008.
- [37] R. Krovi, A. Chandra, and B. Rajagopalan, “Information flow parameters for managing organizational processes,” *Communications of the ACM*, vol. 46, no. 2, pp. 77–82, 2003.
- [38] F. Lin and T. E. Cheng, “The structural theory of general systems applied in management: The total relationship flow management theorems,” *Int’l Journal of General Systems*, vol. 36, no. 6, pp. 673–681, 2007.
- [39] D. Ben-Arieh and M. A. Pollatscheck, “Analysis of information flow in hierarchical organizations,” *International Journal of Production Research*, vol. 40, no. 15, pp. 3561–3573, 2002.
- [40] S. Winkler, “Information Flow Between Requirement Artifacts. Results of an Empirical Study,” in *REFSQ*, 2007, vol. 4542, pp. 232–246.
- [41] K. Schneider, K. Stapel, and E. Knauss, “Beyond Documents: Visualizing Informal Communication,” in *Requirements Engineering Visualization*, 2008, pp. 31–40.
- [42] K. Petersen and C. Wohlin, “Context in industrial software engineering research,” in *ESEM*. IEEE, 2009, pp. 401–404.
- [43] B. Nuseibeh, “Weaving together requirements and architectures,” *Computer*, vol. 34, no. 3, pp. 115–119, 2001.
- [44] D. M. Fernandez, S. Wagner, K. Lochmann, A. Baumann, and H. de Carne, “Field study on requirements engineering: Investigation of artefacts, project parameters, and execution strategies,” *Information and Software Technology*, vol. 54, no. 2, pp. 162–178, 2012.
- [45] E. Arisholm, H. Gallis, T. Dyba, and D. I. Sjöberg, “Evaluating pair programming with respect to system complexity and programmer expertise,” *IEEE Transactions on Software Engineering*, vol. 33, no. 2, pp. 65–86, 2007.
- [46] H. Femmer and I. Esau, “The Software Engineering Simulations Lab: Agentic AI for RE Quality Simulations,” in *REFSQ’26*, 2026.
- [47] C. E. Shannon, “The mathematical theory of communication,” *MD Computing*, vol. 14, no. 4, pp. 306–317, 1997.