Requirements Engineering for Machine Learning

A study in Behavior-Driven Development

J.M. Rosenberg

Delft University of Technology

TU Delft

Requirements Engineering for Machine Learning

A Study in Behavior-Driven Development

by

J.M. Rosenberg

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Tuesday May 20, 2025 at 15:00.

Student number:	4839641	
Project duration:	September 1, 2024 – May 20, 2025	
Thesis committee:	Dr. C. C. S. Liem,	TU Delft, Multimedia Computing, supervisor
	A. Bartlett, Bsc.	TU Delft, Multimedia Computing, Daily supervisor
	Dr. C. E. Brandt,	TU Delft, Software Engineering

Cover: *"Portrait of person and AI robot"* by Freepik

An electronic version of this thesis is available at `http://repository.tudelft.nl/`.

TU Delft

Preface

This thesis officially marks the end of almost seven years of studying and, with that, my student life. There were many ups and downs, but I can gladly say that I look back with a smile. I would like to take this moment to look back at some of the highlights.

It all started in 2018, a new city and a fresh start. I quickly found a room, made friends, became active in a sports association, a study association, and from my second year onward, mainly in a student association. When Corona hit, the whole world changed, but studies continued as usual. With less physical contact also came less motivation. When things started going back to normal, so did my motivation. I received my bachelor's, made more friends than ever, and traveled as much as possible during vacation periods. Throughout my entire studies, I always had my parents and two sisters by my side, and for the past 3.5 years, my girlfriend Lucy. I am forever grateful for their unconditional love and support.

During the last eight months, I had a great time working on this research. I would like to thank my supervisors, Antony Bartlett, for our weekly meetings, and Dr. Cynthia Liem, for joining every other week with experienced input. I would also like to thank my friend Aleksander Buszydlik for meeting with me on several occasions to offer advice and for reviewing this document.

J.M. Rosenberg
Delft, May 2025

Contents

1 Introduction

With the increasing application of Artificial Intelligence (AI) and Machine Learning (ML) in software systems, traditional Requirements Engineering (RE) methods are faced with new challenges. Unlike traditional software, where requirements can be explicitly defined upfront, AI/ML systems depend on data-driven processes to find patterns and make decisions [29, 3]. This means that the behavior of the ML components may not necessarily be defined directly by human developers, but instead emerges from training data and learning algorithms, such as logistic regression and K-means clustering. This flexibility in ML systems also introduces uncertainty and unpredictability in its behavior. Without clear, predefined requirements, it is even harder to determine whether an ML system would fulfill the stakeholders' expectations. Furthermore, the lack of strict requirements for data scientists increases the risk of adverse and unintended consequences, such as biased decision-making [48], lack of transparency [27], and ethical concerns [40]. These risks demonstrate the need for RE practices that can satisfy requirements, such as fairness, explainability, and trustworthiness in AI/ML development.

RE is a crucial aspect of software engineering as it helps to establish explicit expectations between stakeholders regarding both functional and non-functional requirements of a system [34]. Here, functional requirements refer to what the system should do, while non-functional requirements refer to how the system should behave [16, 51]. By having a strict set of requirements, RE can guide the software development process to ensure that the final product matches the goals of the stakeholders. When applied to traditional software systems, RE is typically integrated into broader software engineering development cycles, such as those structured by Waterfall or Agile [26]. They provide a framework to structure an uncertain working process throughout development. However, these approaches are becoming less effective for RE in the context of AI and ML, where traditional requirements may no longer suffice [11]. In particular, AI/ML systems impact process governance and quality assurance, as their behavior is guided by data-driven and non-deterministic processes, rather than predefined logic [4, 22, 25]. This shift shows that there is a need for new engineering practices, such as Machine Learning + Operations (MLOps), which extend Development + Operations (DevOps) principles. DevOps is the integration and automation of software development and information technology operations. MLOps builds on that foundation by emphasizing the continuous monitoring, validation, and adaptation required for AI and ML systems [43, 37].

Requirements Engineering for Machine Learning (RE4ML) is a particularly challenging task because of the dynamic requirements needed. This is even more evident with requirements on explainability, trust, and fairness [27, 50], where expectations from data scientists, legislation, and supervisors are often unclear or subjective. In addition, the iterative nature of supervised learning has a tendency to resemble aspects of a test-driven development (TDD) process [35]. In this context, the training data acts like a test suite, which is determined before system development is started. Models are iteratively trained and refined to fit data points to labels up to a certain threshold. This process of optimization introduces dynamic requirements that change along with the data and algorithms, further complicating the process [32]. These challenges highlight the importance of RE methods that are de-

signed specifically for ML systems and ensure that the development process is equipped to deal with uncertainty and adaptability.

Another process that is used to set up requirements is `Data-Driven Development (DDD)` [31], which is based on the data and description thereof. By using large datasets, DDD builds a system using real-world data patterns rather than theoretical assumptions. However, DDD often focuses primarily on the data rather than user expectations or system behavior to develop the system. This is where `Behavior-driven development (BDD)` can jump in. Unlike TDD, which focuses on writing tests before writing code, or DDD, which emphasizes data as the primary driver of development, BDD focuses on defining system behavior in a way that is both executable and understandable for non-technical stakeholders. In this way, BDD stresses the importance of collaboration between developers and all stakeholders by using understandable language examples and taking their expectations into account [41, 18]. Furthermore, integrating BDD into ML projects can improve accountability and transparency, especially when ethical considerations and explainability are crucial [14].

In addition to a method for developing requirements, there are different ways to structure and visualize them. One method that does that effectively through the use of human-readable requirements and translating them into visual conceptual models is `Goal-Oriented Requirements Engineering (GORE)` [49]. By representing each goal in a model and linking it to metrics, GORE tries to show the underlying rationale for those goals explicitly. As a result, this method of visualizing tries to bridge the gap in terms of explainability between technical and non-technical stakeholders.

This research aims to explore methodologies and frameworks that address the challenges of RE4ML development. We used a mixed-method approach and conducted a series of interviews, combined with a survey, to understand crucial aspects such as dynamic requirement specification and aligning business goals with system behavior. Here, it should be emphasized that ML models can reinforce biases or have unintended consequences, especially when certain stakeholders are not taken into account in the process or are over- or underrepresented in the data. We found that concise yet clear BDD requirements, as well as visualizing those in a conceptual model, improve explainability for people with ML knowledge. This qualitative and quantitative evidence contributes to the existing knowledge on RE4ML using BDD and GORE. By applying these insights to real-world working cases with significant societal impact, such as the COMPAS[1] recidivism dataset, this research seeks to bridge the gap between frameworks for traditional and ML models.

1.1. Research Questions

Our research operates in the domain of RE4ML using the two complementary approaches of `Behavior-Driven Development (BDD)` and `Goal-Oriented Requirements Engineering (GORE)`. In this research, we focus these methods on aspects such as explainability, trust, fairness, and transparency, and all will be referred to under the umbrella term of “explainability and trust”. These aspects align with the characteristics of our working cases with high societal impact, such as ethical considerations and complex decision-making.

Through our related work, we identified two primary gaps. First, BDD is used in only a limited number of studies and is rarely applied to capture requirements related to explainability and trust. Secondly, GORE could visualize stakeholder needs better than current practices and lead to improved conceptual models.

Therefore, we would like to know how effective our BDD approach is, which types of tests can engage a diverse audience, and how well GORE can meet different stakeholder needs. To achieve these objectives, we have formulated the following research questions.

- RQ 1: To what extent can Behavior-Driven Development be used to identify requirements for “Explainability & Trust”?
- RQ 2: What type of tests can be used to encapsulate different needs of stakeholders?

[1] https://github.com/propublica/compas-analysis

- RQ 3: How well can we visualize different stakeholder needs regarding “Explainability” using Goal-Oriented Requirements Engineering?

1.2. Thesis Structure

The rest of the thesis is structured as follows:

- **Chapter 2** dives deeper into the concept of RE4ML, including several types of development and current shortcomings.
- **Chapter 3** explains the framework used in this research, together with how we went from interviews to conceptual models and a survey using our working cases.
- **Chapter 4** talks about the high-risk, societal datasets used in this research and the initial requirements we set for them.
- **Chapter 5** kickstarts the process for the interviews with experts in the field of ML and BDD.
- **Chapter 6** explains the two GORE modeling languages used to create the conceptual models for our working cases.
- **Chapter 7** describes the survey and shows its results using tables and visualizations.
- **Chapter 8** interprets the results and tries to answer our research questions as posed in Section 1.1.
- **Chapter 9** wraps up this research by addressing limitations, adverse impact, and giving pointers for future work.

2

Related Work

To understand what current research exists in the Requirements Engineering for Machine Learning (RE4ML) field, it is relevant to start with traditional Requirements Engineering (RE) standards to understand existing RE principles and their limitations. In the section that follows, we will continue by explaining the challenges and needs RE faces when talking about ML system development, including existing modeling languages. Hereafter, several existing types of development are presented. Lastly, this chapter will explain how different subjective requirements, such as explainability and trust, fall into place for ML as they are becoming more important.

2.1. Requirements Engineering for Traditional Software

Requirements engineering is relevant for developers to consider up front what they actually want to build, rather than erratically going into development and testing afterwards. In this section, we will explain more about the process of thinking about how a system should behave before it is implemented.

2.1.1. The Importance of Predefining Expectations

RE was designed to create a strict set of requirements that the system should adhere to before implementation [23]. In this way, developers could use engineering principles for previously unstructured software development processes [52]. A part of development could now be put into specifying what the system needs to do, rather than only testing its functionality afterward.

Unfortunately, there are still several limitations to RE in traditional software. The majority of the limitations are related to stakeholders, such as managers or programmers, who do not know what they want or are unable to adequately explain what they need. This can lead to incomplete or inaccurate requirements. Hence, when a new part of the system is designed, some functional requirements may have been missed beforehand. Additionally, Machine Learning (ML) can be considered as a new field for which new RE techniques need to be designed [19]. Unlike traditional software, where requirements are often static and explicitly defined, ML systems learn from data and evolve over time [3]. This means that in both traditional and ML systems, it is important to set requirements up-front, but for ML systems there needs to be a technique that allows for evolving requirements. This change seems necessary as traditional methods often assume that requirements remain fixed once defined. As such, existing approaches may fall short when applied to ML system development, where requirements can change during development. Therefore, it is important to understand different structuring methods for a development process.

2.1.2. Common Structuring Methods

How requirements are created in a software development process depends heavily on how the process is structured. The most widely used method today is Agile, which has superseded the more traditional Waterfall method. [46]. The method that one chooses impacts how requirements are maintained throughout the development.

The Waterfall method is a sequential design process in which progress flows downwards like a waterfall. This structuring method then goes through the phases of requirement gathering and analysis, design, coding, testing, and maintenance [26]. Here, one moves on to the next phase only when the current phase is completed, meaning that there are generally clear and documented objectives. Once a phase is completed, revisiting a previous phase to make changes is often difficult and costly. This lack of flexibility especially poses problems when the final system does not meet the expectations of the owner, but all previous versions did.

Another way to structure the process is Agile [26]. Unlike the Waterfall method, Agile focuses on continuously identifying, documenting, and managing the requirements for a project. By breaking the project into smaller cycles, called sprints, Agile allows changes to occur throughout development. This adaptability gives Agile the possibility to make changes more easily at a later stage. Furthermore, this adaptability also makes way for evolving requirements in ML. However, Agile also presents several challenges. It expects stakeholders to collaborate closely in each sprint to set requirements from the perspective of a product owner and a developer. This can cause unnecessary requirements to be added that seem important at the time.

2.2. Development vs. Machine Learning Operations

`Development + Operations (DevOps)` and `Machine Learning + Operations (MLOps)` are two software development methods that revolve around how to manage traditional and ML systems, respectively. While they share similar components, such as automation, `Continuous Integration (CI)`, and `Continuous Development (CD)`, MLOps introduces additional features that are unique to ML workflows [43].

DevOps is a set of practices that combines software development (Dev) and IT operations (Ops). It is a way to improve the collaboration between developers and operations teams, by automating parts of the software delivery process [33]. Its two key components are CI and CD. CI means that the code is in a central repository and that changes are merged frequently, and running tests on each merge to detect issues early on. CD, on the other hand, means that validated code can be released automatically, ensuring that the software can be released at any moment.

MLOps extends the principles of DevOps to the ML domain by addressing the difficulties of continuously deploying and maintaining ML systems. Next to the DevOps principles of CI/CD, MLOps introduces additional considerations, such as automated deployment, model monitoring, and model versioning [2]. The implementation of MLOps practices aims to bridge the gap between ML and DevOps, enabling an organization to deploy ML models more efficiently. However, adopting MLOps can be resource-intensive, and its benefits may only be found in systems where a high continuous deployment is needed.

Another development method that extends DevOps is `Testing + Operations (TestOps)` and introduces automated testing pipelines that not only verify code changes, but also identify and prioritize what will be tested, how, when, and by whom [13]. This process of automation in testing, as well as MLOps does in the ML field, shows the importance of new development methods needed for RE4ML.

2.3. Requirements Engineering Challenges in AI-based Systems

Belani et al. [4] discuss challenges that AI poses to RE processes, such as non-existent frameworks, unavailable or imbalanced datasets, and unclear (ethical) regulations. In their research, they call for the integration of `Goal-Oriented Requirements Engineering (GORE)` into the design of frameworks and explain the difficulties for each step in a RE process to go towards a RE4AI (RE for AI) taxonomy. GORE is a graphical modeling approach that takes high-level stakeholders' goals and systematically translates them into metrics that can meet the requirement [49]. Belani et al. stress the fact that, unlike traditional software systems, where the output is deterministic, AI systems are subject to probabilistic outcomes because of the learning nature of AI algorithms. This creates challenges when aligning system behavior with stakeholder needs. A GORE framework could potentially model these different needs during the development of a system. However, the unpredictability of AI models makes it difficult to create a one-size-fits-all RE approach, as more studies are needed to come up with evo-

lutions.

Nalchigar et al. [32] take this a step further by looking at the impact of ML on business goals and have created a new framework to conceptually model the requirements called GORE for Machine Learning (GR4ML). They found that existing RE practices, such as GORE, can be used to align AI-driven outcomes with business objectives. Unlike Unified Modeling Language (UML), which is a way to visually represent the design of a complex software system and is based on static diagrams, GR4ML offers textual visualizations. It structures these visualizations into three modeling views: business, analytics, and data preparation. This way of visualizing the different necessities of a system makes the GORE technique more usable in ML systems. However, they highlight that it is critical to manage the uncertainty of ML models by taking feedback loops into account throughout the development lifecycle. This makes RE4ML different from traditional software, as requirements tend to be more static and defined early on.

Another tool that is making an appearance in the GORE field is i* [45]. This modeling language focuses on the dependencies of relationships among various stakeholders within a system. It is useful for capturing and analyzing the needs of stakeholders while providing a simple yet necessary visualization of the system requirements. Just like in GR4ML, this provides the possibility of modeling high-level intentions in clear language.

In addition to looking at what shortcomings RE practices have for a whole system, Giray [15] evaluates the process from a different angle. He notes that AI is often built as a separate part of the system and studies the attention that these additions need. He concludes that this requires new approaches in RE to handle the dynamic nature of ML models, such as their need for frequent updates and retraining, while also ensuring compatibility with the rest of the software system.

Throughout this research, we use both GR4ML and i* as our primary conceptual modeling languages to explore these challenges for real-world practices. Simplified examples of both modeling languages can be found in Appendix A.

2.4. Types of Development

In the context of RE4ML, various development methods have been proposed to address specific challenges. One widely used technique in ML development is Data-Driven Development (DDD), which focuses on using historical, currently available data, and the descriptions thereof for system development [31]. Using data in this way, causes this technique to ensure that the system is built based on real-world data trends. However, biases or ethical concerns in the data may be missed during development. This approach is predominantly used in systems where data-driven insights are necessary for model training and validation, such as recommendation systems [44] and predictive analytics [54].

In addition to building the system around the data, one can also look at what the system is responsible for. In the paper “Towards Accountability-Driven Development for Machine Learning Systems” [14], Fung et al. introduce a framework called Accountability-Driven Development (ADD), which is used to enhance accountability in ML systems. They propose a combination of the DDD approach with a description of the behavior of the system using natural language to put accountability on the whole project team, rather than only on ML engineers. ADD is particularly useful for addressing the need for accountability in ML applications, especially in impactful domains such as healthcare and autonomous driving.

Another study that focuses more on system behavior explores the integration of human emotion (HE) into the development pipeline [10]. Through a case study, Curumsing et al. demonstrate how considering the emotional needs and responses of interviewees can lead to more user-friendly systems. This method can be relevant in applications such as home systems, where understanding emotions can result in more intuitive system behavior.

Both ADD and HE are methods based on the Agile development method called Behavior-Driven

Development (BDD) [41, 18]. This method stresses the importance of collaboration between developers and all stakeholders through natural language examples. Rather than focusing on the data like DDD does, it takes the expectations of stakeholders into account by encouraging teams to use conversation and concrete examples to set expectations for the system. Therefore, this practice is a good way to provide an understanding of requirements that match what people expect the system to do.

In this research, we will build on BDD because of its ability to take different stakeholder needs into account. This method can help improve the current ways of setting requirements for both technical and non-technical stakeholders.

2.5. Explainability and Trust in Requirements Engineering

Maalej et al. [30] introduce the need for explainability and transparency in ML system development. They test traditional RE techniques, such as trade-off analysis, in the context of ML and conclude that, while such techniques are still relevant, the complexity of AI makes it more difficult to implement them in practice. Ethical considerations, such as fairness, transparency, and bias mitigation, now play a more important role in ML system development than they did in traditional software, where functionality tends to be the dominant concern.

A major change identified by Kohl et al. [27] is the growing importance of Non-Functional Requirements (NFR) for ML. Specifically, NFR regarding fairness, explainability, and trustworthiness. They emphasize that explainability is becoming a critical NFR in ML systems, as stakeholders need to understand the rationale behind AI decisions. This is even more important in high-stakes domains, such as healthcare and finance. They argue that RE practices must evolve to incorporate explainability as a core requirement, which is less of a concern in traditional software systems.

These NFRs can sometimes be seen as Functional Requirements (FR) when talking about ML systems [12]. The general difference is that FRs refer to what the system should do, while NFRs refer to how the system should behave [16, 51]. An example of this could be the explainability of a system. When phrased explicitly (for example, “provide a visual feature importance chart when requested”), this requirement goes from a broad quality (NFR) to a specific feature (FR). This also shows that NFRs are essential, but for ML system development, we need to go to a stricter set, such as RE does.

Trustworthiness is another central theme in the literature. Heck et al. [20] propose that a comprehensive quality engineering toolbox should be designed for ML systems that uses tools, techniques, and guidelines to ensure that the system meets the necessary quality standards. These guidelines focus on ensuring that ML systems are reliable, fair, and safe for end-users to address issues like bias, accountability, and data security. This approach makes RE again different from traditional software, where quality is often easier to validate.

In the same area of explainability and trust, Bergelin et al. [5] investigate how industry standards for AI systems differ from academic frameworks. They highlight the need for more collaboration between industry and academia. They suggest that industry requirements, particularly in domains with significant societal impact such as healthcare and autonomous driving, should demand a high-level standard in terms of safety and accountability. In addition, thirty generalized requirements are elicited to improve the modeling, coding, testing, monitoring, and continuous development of AI systems.

By conceptualizing requirements that can be seen as both functional and non-functional, we are trying to fill the gap of missing RE practices for ML. We propose using an RE4ML framework that captures the evolving needs of explainability and trust in ML system development.

2.6. Requirements Engineering for Human-Centered AI

Research has emphasized the need for RE to include human-centered aspects [30, 1]. This is especially true when creating responsible and ethical AI systems, like government IT systems. They argue that traditional RE methods, which focus on the technical specifications of the software, need to be extended to take ethical considerations into account. This can improve fairness and reduce bias in ML

systems. Human-centered requirements should address issues such as inclusivity, transparency, and the social impact of ML decisions. This can ensure that AI matches human values and social expectations.

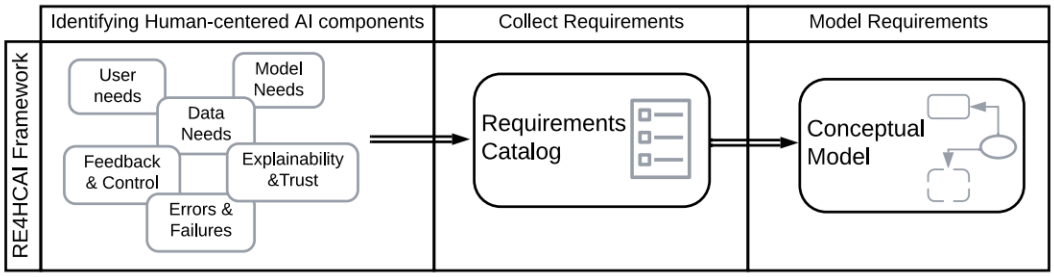


Figure 2.1: RE4HCAI framework to elicit and model requirements for human-centered AI. Adapted from [1].

Ahmad et al. [1] have created a framework to elicit and model requirements for `human-centered AI (HCAI)` called `RE4HCAI`. This framework consists of three parts, as can be seen in Figure 2.1. First, the HCAI components necessary for the system development are identified. They identified six HCAI components that generally apply to an ML system. In the second step, they collect requirements based on the system description for each of those components and put them in a catalog. In the last step, they conceptually model the requirements using UML. However, in their paper, they state that a GORE approach would be more suitable for visualizing stakeholder needs. GORE is different from UML because it focuses on the objectives of stakeholders behind the requirements, rather than visualizing the requirements only based on the structure of the systems that UML does [25]. In this way, NFR could be presented in a visualization using GORE.

3

Method

This chapter explains our mixed-method approach and how we adapted the RE4HCAI framework [1] to focus on explainability and trust. We included expert interviews and a survey into this framework to study the usability of `Behavior-Driven Development (BDD)` and `Goal-Oriented RE (GORE)` in `Machine Learning (ML)` to answer our research questions. Both the interviews and the survey are explained more thoroughly in Chapters 5 and 7, respectively. Figure 3.1 shows an overview of the research structure.

3.1. Adapting the RE4HCAI Framework

The method of this research extends the RE4HCAI framework [1], which originally consisted of three steps; Identifying HCAI components, requirements collection, and the creation of a conceptual model using UML. In this section, we discuss each step and highlight our additions to it.

Identifying Human-Centered AI Components

The focus of this study is on the "Explainability & Trust" part of the components identified in the RE4HCAI framework. These components also play a dominant role in the RE4ML field as seen in Chapter 2. Additionally, in this research, we have included safety in this component, as they often occur together as a requirement in the creation of ML systems [47, 40].

Collect Requirements

To start the RE process, we created a list of generic requirements that societal impactful systems should adhere to. These requirements can be found in Section 4.3. Our chosen systems needed to have an impact on society, so we decided on the COMPAS recidivism and a loan approval dataset, which are presented in Chapter 4. By starting with generic requirements for these systems, the BDD approach

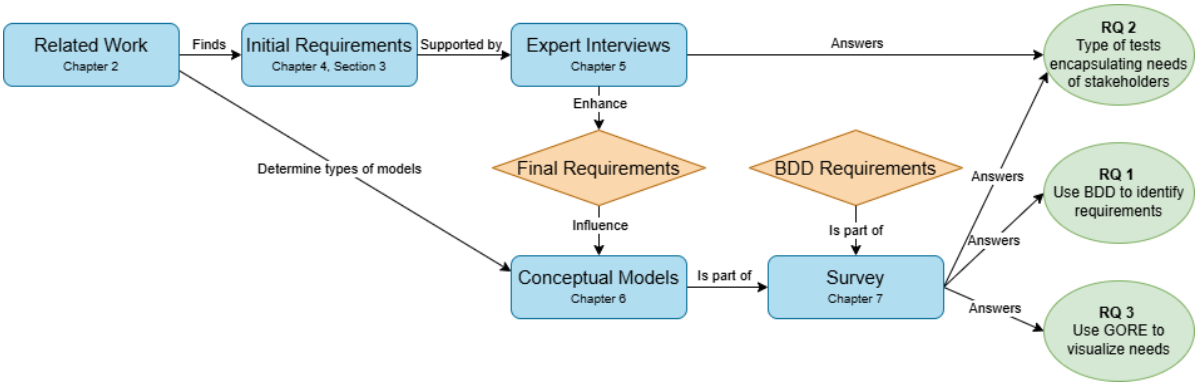


Figure 3.1: Flowchart illustrating the method of the research and showing which chapters address each research question.

for RE can easily be adapted to equivalent, high-risk systems and could then be adjusted to the specific model.

After the initial requirements were created, they had to be worked out. This means not only saying that the system should be accurate, but also giving a reason for why this requirement was created and how this is defined. In this way, it is possible to see when this requirement is met. To test when such a requirement is met, functional test cases can be created and run for the working cases. These tests should be extensive enough to capture needs from different stakeholder perspectives. Furthermore, a test may fail in certain scenarios due to under- or over-representation or in unethical situations.

As an addition to this step, we included expert interviews to support the creation of initial requirements for our working cases. In this way, we can verify, improve, and complete these requirements. These interviews are discussed in further detail in Chapter 5. They served as a necessary step to qualitatively show the applicability of this approach. These interviews also assessed trade-offs between objectives, such as fairness vs. accuracy, and identified potential gaps that were unclear beforehand. This addition of interviews gave more insight into explainability and trust using BDD (RQ 1) and provided information on the types of tests that can encapsulate stakeholder needs (RQ 2).

Model Requirements
With conceptual models, the workings of a system can be visualized. In the original RE4HCAI framework, UML was used to conceptually visualize the workings of a system. However, in this research, we have used GORE modeling languages as a way of visualizing our working cases (see Chapter 4). GORE is able to visualize the requirements in a lower level of abstraction, which is useful because more detail can lead to a higher level of explainability.

After the creation of these models, a survey was conducted (see Chapter 7). This extra step helped us evaluate the explainability of these conceptual models to answer RQ 3, as well as to evaluate the use of BDD for RE to answer RQ 1. The method for the interviews and the survey is discussed in the next sections.

3.2. From Interviews to a Conceptual Model

To better understand important aspects of the RE process, this study began with structured interviews in the context of BDD for ML. There were two objectives to these interviews, based on our research questions from Section 1.1. First, to understand what current practices the interviewees use and what they think about existing tools. Secondly, to collect qualitative feedback for our predefined baseline of requirements. These requirements can be used to assess the system's performance. The content and setup of these interviews are discussed in more detail in Chapter 5.

Based on conclusions from the interviews, we were able to verify, improve and complete the initial requirements to obtain the final list of requirements. These requirements support the creation of a conceptual model, which is the final step of the RE4HCAI framework.

3.2.1. Expert Interviews

Expert interviews were conducted to verify the requirements of the RE process. These interviews were an important step in determining the correctness and completeness of the set of initial requirements for our working cases. These insights helped us identify potential gaps in the requirements, validate them, and ensure that they aligned with both practical and ethical concerns.

We contacted people who we expected to have experience with either ML system development or those who have used BDD in system development. In total, three experts joined the study. By focusing on experts with different practical experience in the ML field, it was possible to capture both client and user needs.

To capture input on definitions and experience, a semi-structured interview approach was chosen to maintain consistency across interviews while also allowing for exploratory discussions based on the

insights of the participants. The interviews lasted about one hour and had both closed and open-ended questions to allow for thorough discussions. All interviews were recorded via Teams and manually transcribed based on these recordings. The results of these interviews were analyzed using a thematic analysis [9] along the focus areas of explainability and trust. We used a deductive approach to gain insight in our research question about explainability of BDD (RQ 1) and answer parts of our research question on what types of tests show different stakeholder needs (RQ 2).

3.2.2. Conceptual Model Development

Based on the requirements identified through the literature and expert interviews, we developed two conceptual models for our working case of the COMPAS recidivism dataset. These models serve as an abstract illustration of various relationships between different requirements and stakeholders. We applied different `Goal-Oriented Requirements Engineering (GORE)` techniques to construct these conceptual models that capture the complexity of the requirements. The conceptual models and what they entail can be found in Chapter 6.

The first conceptual modeling language used was GR4ML [32] (see Section 2.3), and is designed to integrate data science workflows with business goals by dividing the ML pipeline into three separate views. The Data Preparation View depicts the transformation and filtering of raw data, while the Analytics Design View shows the model training and evaluation steps. Finally, the Business View models the ML results to specific decision targets in the domain models. These views aid in explaining workflows (and thus explainability) where otherwise no visualizations would be created. This allows data scientists and operations teams to pinpoint where monitoring or explainability can be improved.

Our second conceptual modeling language is i* [45] (see Section 2.3), which places more emphasis on the social and organizational aspects of actors in the system and their goals and dependencies. Each actor node is attached to soft and hard goals, which are requirements, such as explainability and fairness. They contain links indicating specific metrics that achieve those goals. This notation is great for exposing trade-off tensions, such as model accuracy versus transparency. It also improves the visualization of dependencies for each stakeholder. This makes it easier to understand which requirement depends on which stakeholder.

By integrating these visualizations, we aim to depict requirements that otherwise would have been missed in the development process. This should eventually help government and companies in thinking about how such requirements fit into societal impactful system.

3.3. Survey Design

Following the conclusions of the requirements from the interviews, a survey was carried out. The purpose of this survey was to evaluate the perceived advantages and limitations of using BDD and GORE for ML development. The combination of interviews and a survey ensured that insights were collected both at the beginning as well as after the creation of the conceptual models. This should provide a good understanding of the effectiveness of the proposed approach.

To ensure that participants have a basic understanding of how the systems should work, we asked people to participate in the survey who have a basic understanding of ML. This gave us a wide pool of developers, data scientists, and other participants who are, or have been, active in ML projects.

Before we could answer our research questions, we first had to understand what aspects of GORE and BDD support explainability in requirements. Therefore, our survey (see Appendix D) asked the participants demographic questions, in which we also included knowledge about ML, RE, and BDD. The survey continued with questions about general explainability caused by BDD. Next, we went over the explainability from our GORE conceptual models and what they did and did not like about those visualizations. Lastly, we went over BDD requirements and asked how explainable each requirement is on a scale of one to ten and why they rated it that way. These open-answer explanations were then analyzed to see what aspects of a requirement can improve explainability. These insights aim to obtain quantitative data to help answer RQ 1 and RQ 3.

3.3.1. Evaluation of Survey Results

For the evaluation of the survey responses, we first looked at the direct answers. This relates to all questions that contain a preference or a rating. These insights could then be used to capture general information, such as preferred visualization format and rating of explainability for a conceptual model and BDD requirement. This has set the first step into answering RQ 1 and RQ 3

Next, we dove into the open-ended answers related to the conceptual models. This provided answers to which aspects of the two conceptual models, created using GORE, participants liked and disliked. To further analyze these results, we split up the participants based on ML model knowledge. This gave a better understanding of the explainability of each conceptual model and made it possible to answer RQ 3.

Lastly, we analyzed the ratings of the BDD requirements based on the one-to-ten rating as well as the open-ended responses. Just like with GORE, we could use these open-ended answers to find the aspects that show explainability and trust to answer RQ 1.

4

Working Cases

This chapter explains the two datasets used in this study: the COMPAS recidivism dataset and a loan approval dataset. These datasets were chosen based on their societal high-risk to highlight the importance of explainability and trust in ML system development, which was our focus in the RE4HCAI framework (see Section 3.1). After describing the datasets, we introduce the initial requirements we set for these models. These requirements are verified and improved after the expert interviews in Chapter 5. With these final requirements, the conceptual models were created for the survey in Chapter 7 to test GORE and BDD as explained in Sections 3.2.2 and 3.3.1.

4.1. COMPAS Recidivism

The COMPAS[1] (Correctional Offender Management Profiling for Alternative Sanctions) dataset consists of information about criminal defendants in the U.S. justice system, including demographic data, previous offenses, and recidivism outcomes. This dataset is widely used to study the predictive algorithm developed by Northpointe, which calculates the probability of recidivism. Regardless, this model is criticized for its potential (racial) biases, as revealed in an analysis by the non-profit investigative journalism ProPublica [28]. ProPublica collected 80.000 data points from Broward County and trained an ML model that uses 53 features. Among other findings was the disproportionate classification of African-American defendants as high-risk compared to Caucasian defendants, regardless of actual results. Such biases may cause unfair sentencing and systemic discrimination within the criminal justice system.

The COMPAS dataset is particularly relevant for studying the explainability of BDD in ML due to its real-world application and its significant impact on individuals and society. It demonstrates the importance of requirements such as fairness, transparency, and ethical considerations in the design of predictive models. By studying the explainability of BDD in this context, we can work towards developing more transparent ML models.

4.2. Loan Approval

The purpose of the loan approval dataset[2] is to use the financial history of an applicant to determine whether a loan request should be approved or rejected. It consists of 4.296 entries that contain thirteen features of applicants, including income, credit score, assets, and employment details, along with the amounts of the loan and whether it was approved or not. Unlike the COMPAS dataset, this dataset does not contain racial features, making it a useful comparison to study the impact of various forms of bias in ML models. Researching this dataset allows us to explore biases that may appear in characteristics such as gender, education, or income levels.

[1] https://github.com/propublica/compas-analysis
[2] https://www.kaggle.com/datasets/architsharma01/loan-approval-prediction-dataset/data

In addition, the need for explainability is important in this context because financial institutions need to justify decisions to customers and comply with regulations. Getting more approvals is also profitable for the lender. This dataset serves as a practical example for understanding how ML models can balance accuracy, fairness, and interpretability in high-stakes applications.

4.3. Initial Requirements

To test the explainability of GORE as explained in Section 3.3.1, we need an initial set of requirements for our working cases. These requirements should be generic enough to apply to both of our systems, yet specific enough to be able to generate a functional test case from them. Based on the related work of Chapter 2, we have identified four prominent requirements on accuracy, robustness, explainability, and fairness. It is good to keep in mind that with these requirements, we try to show that it is possible to go from a requirement to functional test cases, meaning that this is not an exhaustive list. For each requirement, we will give the definition we used in this research, together with its importance in high-risk societal ML systems. These requirements and definitions will be further refined and supported by the expert interviews in Chapter 5.

Accuracy

- **Definition**: The system must achieve high predictive performance, with metrics such as precision, recall, and the F1 score being above a defined threshold.
- **Importance**: Accuracy seems to be a central requirement for most ML systems [51, 8], as people want the system to perform well, and it is often easily measurable. Depending on the type of metric used, one generally wants to show that the system performs close to what is expected. Based on literature [42] and our interviews, it should be noted that a high accuracy does not always mean that the system performs well, as there could be a bias towards a certain group and still be correct most of the time.

Robustness

- **Definition**: Predictions must be stable over time for individuals with similar profiles, ensuring that minor changes in input do not lead to disproportionately large differences in predictions.
- **Importance**: Next to accuracy, the system should be stable over time. In this way, the system can be seen as being consistent over time, even if the input data has minor impactful changes. This requirement underlines adversarial-robust ML research [17].

Explainability

- **Definition**: The system must provide explanations for individual predictions to enhance explainability and trust among non-technical users.
- **Importance**: Although this requirement is a little more difficult to test, we still consider it important that a system discloses information on why a decision is made. This requirement is a large topic in the current RE4ML field, as seen in our related work [27, 40]. By reasoning on explanations, systems can become more transparent, and therefore more trustworthy to everyone who is impacted by the system.

Fairness

- **Definition**: The system must make predictions without showing bias towards a group, without having a fundamental or ethical foundation to support this bias.
- **Importance**: Where accuracy would give a score of how well the system performs, it is good to keep in mind that with a high accuracy score, people may no longer look at the underlying reasoning of the system. Even though some research tries to set a requirement for fairness [4], multiple formal definitions exist. We decided to have the second part of the definition so that the system would be fair to groups for which the system has no reason to be unfair.

5

Expert Interviews

Building on the research questions of Section 1.1 and the mixed-method approach from Chapter 3, expert interviews were conducted to study the use of Behavior-Driven Development (BDD) in Machine Learning (ML) system development. In this chapter, we outline the setup and protocol for the interviews, the techniques used to analyze the collected data, and the perceived advantages and limitations of using BDD principles in Requirements Engineering (RE).

5.1. Setup and Protocol

The interviews were designed to find crucial elements in the creation of a ML model regarding explainability and trust. This follows the RE4HCAI framework as per our method. These findings helped us understand several explainability aspects of BDD (RQ 1) and what types of tests we can use in RE (RQ 2).

Before conducting interviews and working with potentially sensitive data, we sought approval from the Human Research Ethics Committee (HREC) from the Delft University of Technology[1]. After a consultation with one of the data stewards, we made final adjustments to adhere to the confidentiality of the university and data protection policies. HREC approved our submission, deeming the project to pose minimal risk regarding possible data breaches.

5.2. Participant Selection

Since the goal of the interviews was to improve the requirements using BDD, the participants had either a good understanding of how ML models are designed or how BDD is used in system development. Therefore, the main criterion was that a participant had worked with ML models or BDD for several years and shall be referred to as an ML expert.

Each participant then signed an Informed Consent Form (ICF) and received information about the models discussed in the interviews. This ensured that the participant had sufficient knowledge about the content during the interview. They were provided with the ICF several days before the scheduled date of the interview. At the latest, the form was signed at the beginning of the meeting.

Eventually, we conducted three one-hour interviews in February and March 2025. All interviews were held online and were recorded via Teams. Based on these recordings, we made manual transcriptions and sent them to the participants. The participants had one week to rectify any answers, but no rectifications were made. All three have worked in the computer science field, with experience ranging from 13 to 25 years, primarily from research in industry branches. They shall be referred to as interviewees 1, 2, and 3.

[1] https://www.tudelft.nl/over-tu-delft/strategie/integriteitsbeleid/human-research-ethics

5.3. Interview Structure

All three interviews followed the same structure to get insights into BDD practices and ML development. The full set of questions can be found in Appendix B. At the start of each interview, a general introduction was given, stating the goals of this research and the added value of the interviews. After the introduction, there were four sections of questions:

General info, where we asked about job title and years of experience to determine expertise, followed by asking about their experience on ML projects.

Trust, Safety and Explainability in ML, where we asked what "Trustworthy AI" means to them and what general requirements they would set for societal impactful systems.

RE and BDD, in which we went over practices they use and are familiar with in an ML development process.

Use Cases, which went into setting expectations for the two use cases in this research. We discussed our requirements, any additions they had, and the priorities within these requirements. This was the largest part of the interview.

5.4. Perceived Advantages of BDD

We stated the goal of BDD as: "To write requirements in plain language that all stakeholders can understand". When asked if they believed this type of development to be useful, all three responded that they do. The reasons behind this were all along the lines of being able to describe requirements in text. With this comes the fact that it enforces stakeholders to at least discuss the requirement specifications with the data scientist.

> *So I think what's interesting is that BDD uses different scenarios... The other thing about BDD that I think is interesting is the conversation that comes out of it that is valuable.*
> *–Interviewee 1*

5.5. Perceived limitations of BDD

Despite its benefits, the participants also noted several limitations and downsides to using BDD. One being that even though a language-based approach is useful, interviewee 3 points out that there are still many things that can go wrong when implementing requirements created by BDD. One example can be that a definition is still not clear enough to be translated into a functional test case.

> *Plain language is super important... [But] you can still mess it up in many different ways*
> *–Interviewee 3*

Another difficulty that became clear is the tools available for RE, or rather, the lack thereof. Only interviewee 1 mentioned having used a tool named Cucumber for this process. Interviewee 3, however, was familiar with certain tools, such as LIME and SHAP, but did not actively use them in system development.

5.6. Requirements Takeaways

When going over the initial requirements, the interviewees generally agreed with this list. They also pointed out that for the goal of showing if BDD could improve explainability, it did not matter which requirements would be set. The definitions for explainability and fairness also seemed good, with an emphasis on the second part of the sentence for fairness; "without having a fundamental or ethical foundation to support this bias."

For accuracy, interviewee 2 pointed out that it may be better not to aim for a perfect score. This would result in humans needing to check the results of the system. For the sake of this research, we kept the definition as is, but kept in mind that high accuracy scores may not always be the best choice.

> *If you tell him [the data scientist] that it is wrong 10% of the time on purpose, then he will need to review everything.*
> *–Interviewee 2*

Lastly, the definition of robustness seemed to get the most discussion. This had to do with the term "minor changes". Two of the interviewees mentioned that changing an important feature in the data for someone could lead to significant changes in the output. An example could be that changing income slightly, yet below a certain threshold, could lead to not getting a loan. Therefore, we will change the phrasing of "minor changes" to "insignificant changes".

5.7. General Takeaways

For the general impression throughout all interviews, it was visible that no participant would trust an ML model to make impactful decisions without a human in the loop. This holds especially for the COMPAS working case, due to its harmful impact when it shows bias. Furthermore, setting requirements for ML systems is a difficult task in general. When asked what the interviewees see as trustworthy AI, we could see that it is not only important to have requirements for what the system should do, but also if what it is doing is correct, and that it should not do more than is required.

> *AI does what it was asked to do. It does only what it was asked to do, and it does what it was asked to do correctly.*
> *–Interviewee 2*

Based on the questions in the section of "Trust, Safety and Explainability in ML" during these interviews, we can say that while non-technical end-users could generally understand the basic ideas behind an ML model, they often do not understand the architecture of ML systems. In contrast, technical end-users are able to explain the workings of ML systems, but still will not always fully understand why a specific output is generated. This shows the need for conceptualizations that all users should be able to understand.

Other important takeaways include biases in ML training. Interviewee 2 mentioned that for the COMPAS dataset, if you do not want race as a feature, you should not train on it. You may lose some accuracy, but that should be fine. Interviewee 3 added to this point that features are often highly correlated, so removing a feature may not remove a bias in the training data. An example that interviewee 3 gave was that in the United States, race and postal code are highly correlated, so removing race may not remove a racial bias.

6

Conceptual Models

In this chapter, we built on our initial requirements from Section 4.3 and a minor definition update for robustness from the expert interviews as stated in Section 5.6, to develop two conceptual models. These models were created using GR4ML (Section 6.1) and i* (Section 6.2) modeling languages to visualize the workflow of the COMPAS recidivism dataset. Both frameworks use `Goal-Oriented Requirements Engineering (GORE)` methods and can therefore be used to enhance the understanding of ML systems to both technical and non-technical users. These conceptual models helped answer how well GORE allows us to visualize different stakeholder needs (RQ 3) for better explainability. Here, explainability refers to the requirements that should improve this and was tested through a survey in Chapter 7.

6.1. GR4ML

As explained in Section 2.3, GR4ML [32] is a conceptual modeling framework to simplify the elicitation, design, and development of ML systems. It consists of three views: the Business View, the Analytics Design View, and the Data Preparation View. Each view addresses different aspects of the system to address different stakeholder needs in a single visualization. This framework is flexible in design in that it can be built top-down, bottom-up, or hybrid.

Our visualization of COMPAS using GR4ML is shown in Figure 6.1 and was made bottom-up. First, the Data Preparation View was made and consisted of filtering and normalizing the data by the data scientist to achieve the final data used in the model. The Analytics Design View then uses this data and runs a logistic regression on it to generate the final prediction model for the Business View. These results, together with a judge's need to maximizing sentencing accuracy, provide a risk score.

The Analytics Design view is the most interesting part for this research, as it contains elements to evaluate the classification process. The classification is evaluated on the four requirements we set, which each contain one or more indicators. These indicators are tests that can be run at any time to evaluate the system and can indicate a pass (green), a warning (yellow), or a failure (red). Explainability is set as a soft goal as it does not have a metric for an automated test. Here, a soft goal is a goal that is considered desirable but the condition to achieve it is not sharply defined. A soft goal is considered to be fulfilled if there is sufficient positive evidence for its fulfillment and little evidence against it.

6.2. i*

The i* (i-star) framework [45] is a modeling approach that focuses on the dependencies of relationships among various stakeholders within a system. It is useful for capturing and analyzing the needs of stakeholders while providing a simple yet necessary visualization of the system requirements.

In the visualization in Figure 6.2, all stakeholders (suspect, judge, data scientist, and government) are directly shown as actors and agents. An agent is an actor who has a concrete physical appearance.

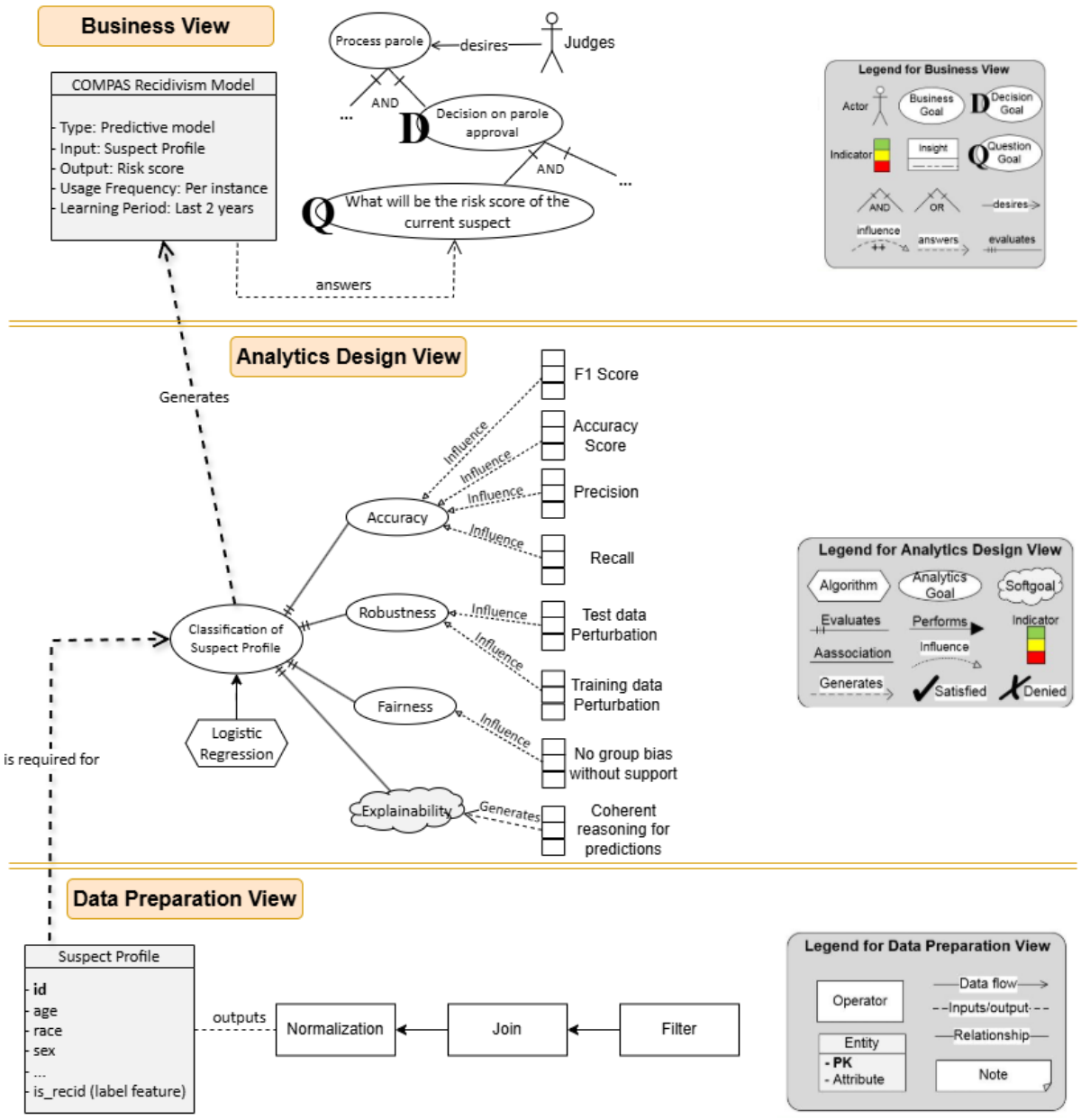


Figure 6.1: Conceptual model of COMPAS using the GR4ML model, which from bottom to top goes through the phases of Data Preparation, Analytics, and Business Views.

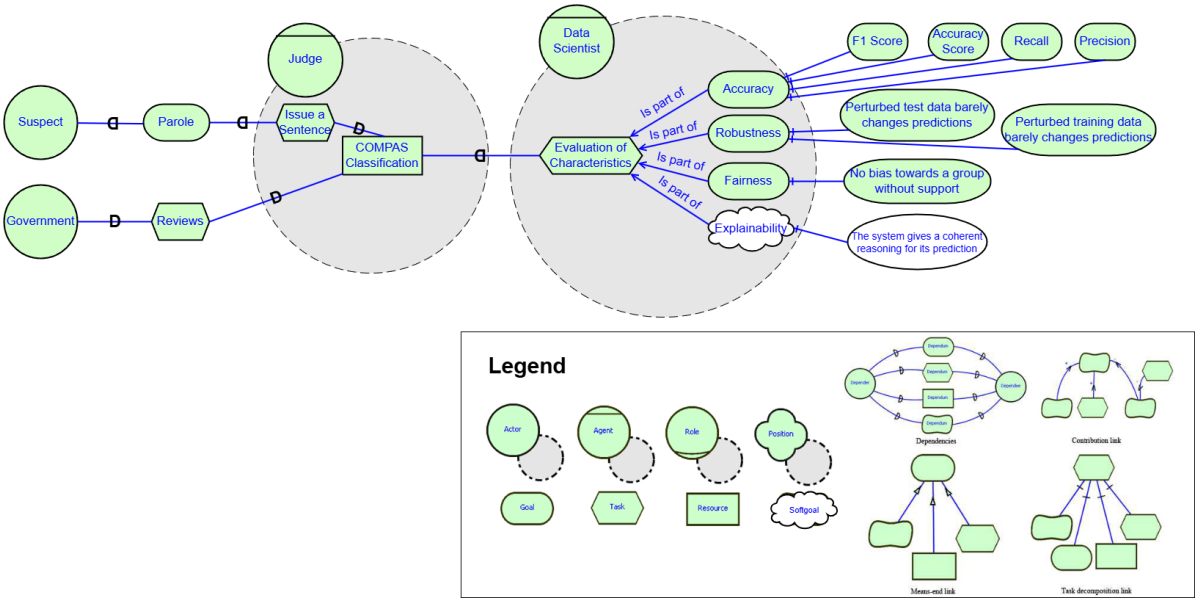


Figure 6.2: Conceptual model of COMPAS using the i* model, which shows the dependencies and tasks of all stakeholders.

The blue lines with a D on them are directional dependency links, which means that one element depends on the effect of another.

In the data scientist circle, the four requirements are shown as part of the evaluation that should be done. Each of them has one or more means-end links from set requirements. These requirements are more text-wise, giving us more flexibility with their interpretations. Like in GR4ML, explainability is shown as a soft goal.

7

Survey Results

The objective of this research is to provide qualitative and quantitative information on the existing knowledge of Requirements Engineering for Machine Learning (RE4ML) using Behavior-Driven Development (BDD) and Goal-Oriented Requirements Engineering (GORE). We conducted a series of interviews with ML experts (see Chapter 5), which provided us with a qualitative basis for our survey. The survey was then used to find information on BDD requirements and conceptual GORE models using the modeling languages GR4ML and i*, in a quantitative setting. These insights allowed us to answer RQ 1 and RQ 3. The entire survey can be found in Appendix D, and the setup and results are discussed in this chapter.

7.1. Survey Structure

The survey starts with some general information questions to analyze the demographics of the response group. Then, there are several multiple-choice questions to understand the expertise of the respondents on this topic. Following this, respondents answered open and multiple-choice questions on two conceptual models of the COMPAS dataset, GR4ML and i*, which are explained in Section 6. These questions focus on the understandability of the conceptual models created by us. Both models were made using GORE techniques. Next, there are examples of six human-readable BDD requirements using the given-when-then approach [53]. Respondents could rate these requirements on a scale from one (very low explainability) to ten (very high explainability) and explain their responses. This allowed us to review what aspects of BDD participants find explainable and could be useful in practice. These six requirements and the aspects we wanted to test are listed in Table 7.1. Lastly, the participants were asked to create a BDD requirement on fairness themselves.

The complete set of survey questions can be found in Appendix D. The answers to these questions are analyzed in the rest of this chapter to answer our research questions.

7.2. Participant Demographics and Knowledge

The survey was distributed among people with different levels of understanding in software development, ML models, and requirements engineering. This diversity helped us get different perspectives on the system. People with more knowledge in the field could tell us if the visualization would be specific enough for them to use it. On the other hand, people with less understanding could inform us if the visualization gives enough explainability of how the system works, without the need for a deeper understanding of how ML systems work.

Our survey consisted of 20 participants, almost all of whom had at least a basic understanding of ML. Their demographics are shown in Table 7.2. All participants were between the ages of 18 and 34, and there is a proper gender representation for the Computer Science field [24], with 13 (65%) male and 7 (35%) female respondents.
.

Table 7.1: BDD-style Requirements and Their Purposes.

Requirement	Aspect
Requirement 1: **Given** the COMPAS model predicts a high risk of reoffending for a defendant, **When** the actual recidivism outcome is later verified, **Then** at least 90% of high-risk predictions should be correct (precision), **And** the system does not falsely label too many low-risk cases as high-risk.	A long requirement validates that positive risk predictions are both reliable and not overly conservative.
Requirement 2: **Given** the COMPAS model has missing data, **When** the model generates a risk score for an individual, **Then** the system should work as if it has no missing data.	Very compact requirement that handles missing data in a possibly ambiguous manner.
Requirement 3: **Given** minor variations in input data (e.g., slight changes in age or number of prior offenses), **When** the COMPAS system calculates risk scores, **Then** the score should not fluctuate significantly unless the change is meaningful.	Compact, yet explained, requirement that tests our updated definition of fairness from the interviews.
Requirement 4: **Given** a defendant receiving a high-risk score, **When** a judge or lawyer reviews the COMPAS output, **Then** the system should provide a clear explanation of the contributing factors (e.g., prior convictions, age).	A compact start and explained result for explainability on an individual level.
Requirement 5: **Given** the historical data is known to have a bias towards a certain group, **When** the data scientist re-trains the model, **Then** the data should be normalized towards this group.	A medium-long requirement for the data scientist.
Requirement 6: **Given** the system provided a risk score with a reasoning, **When** the judge reviews the explanation, **Then** the model should show a raw probability score (e.g., “risk score: 0.84”).	A medium-long requirement that provides a quantitative value as a result, without further explanation.

Table 7.2: Summary of participant demographics and ML knowledge.

Category	Count
Age Group	
18–24	11
25–34	9
Gender	
Male	13
Female	7
Identify as a Minority	
Yes	3
Maybe	1
No	16
Education Level Received	
University - Bachelor	11
University - Master	9
Field of Expertise	
Computer Science	11
Mathematics	4
Engineering	3
AI	1
Law	1
Familiarity with ML models	
Not at all familiar	1
Somewhat familiar	9
Familiar	3
Very familiar	7
Expert	0

Table 7.3: Participant awareness of RE and BDD.

Methodology	Familiar	Heard, not fully understand	Never heard
Requirements Engineering	9	5	6
Behavior-Driven Development	2	12	6

As shown in Table 7.3, almost half of our respondents (45%) said they were familiar with RE. The rest were split between having heard about it without full understanding (25%) and never having heard of it (30%). In contrast, BDD was less well-known, with only two participants (10%) being familiar with BDD, twelve (60%) having heard of it without full understanding, and six (30%) having not previously encountered the term. Although BDD and RE were unfamiliar to six participants (30%), only two (10%) did not know either concept.

Table 7.4: Survey responses on perceived confidence using human-readable rules and BDD-style scenarios in the creation of ML development.

Response	Using human-readable rules	Using BDD-style scenarios
Yes	15	16
Maybe	4	4
No	1	0

Table 7.4 illustrates that a majority (75%) responded that they would be more confident using an ML model if clear, human-readable rules were provided prior to prediction. Four respondents were uncertain about this, and one disagreed. Expressing ML decisions through real-world, BDD-style scenarios makes those decisions easier to understand, according to sixteen participants. Four participants were not sure, yet no one responded with no here. In total, twelve participants answered yes to both questions.

Table 7.5: Survey responses for preferred format of an ML model visualization.

Visualization format	Count
Visual flowcharts showing decision steps	13
Interactive tools where users can test different inputs	3
Technical reports with full model details	3
Simple text descriptions of decisions	1
Other	0

As a last question in the opening part of the survey, we asked the participants which format they preferred for ML model visualization. A fast majority (65%) responded with Visual flowcharts showing decision steps, as can be seen in Table 7.5. There was a tie for second place with three votes each for interactive tools and technical reports.

7.3. Model Explainability

To understand what aspects of the GORE models participants like and dislike, we asked the participants to rate our created models using GR4ML (Figure 6.1) and i* (Figure 6.2) for the COMPAS dataset. First, we asked for a rating in terms of explainability on a scale of one (very low explainability) to ten (very high explainability), and afterward, open-ended questions about what they liked and what they would change about the visualization.

Table 7.6: Survey responses for which stakeholder needs are being met in the two conceptual models.

Stakeholder	GR4ML	i*
Suspect	2	11
Government	9	14
Judge	16	12
Data Scientist	11	18

One aspect to discuss before going into both models is for whom the participants believe the visualization is explainable. In Table 7.6, we can see that for GR4ML, participants believe this to be the case

for the judge (80%), while only about half believe that this is the case for the government (45%) and the data scientist (55%), and only two participants (10%) believe that it is explainable for the suspect. For i* on the other hand, eighteen participants (90%) believe that it is explainable for the role of a data scientist, fourteen (70%) for the government, twelve (60%) for the judge, and eleven (55%) for the suspect.

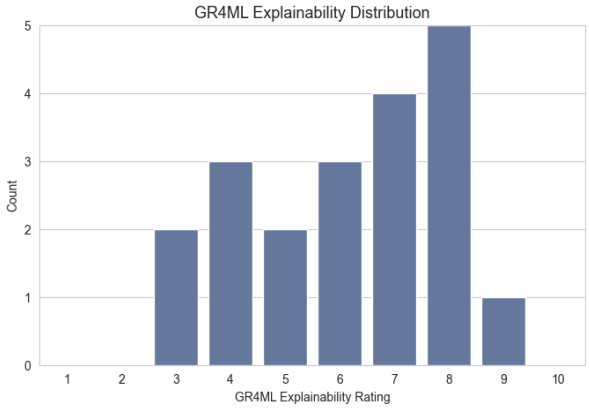


Figure 7.1: Explainability rating distribution from the survey of the GR4ML model for the COMPAS dataset.

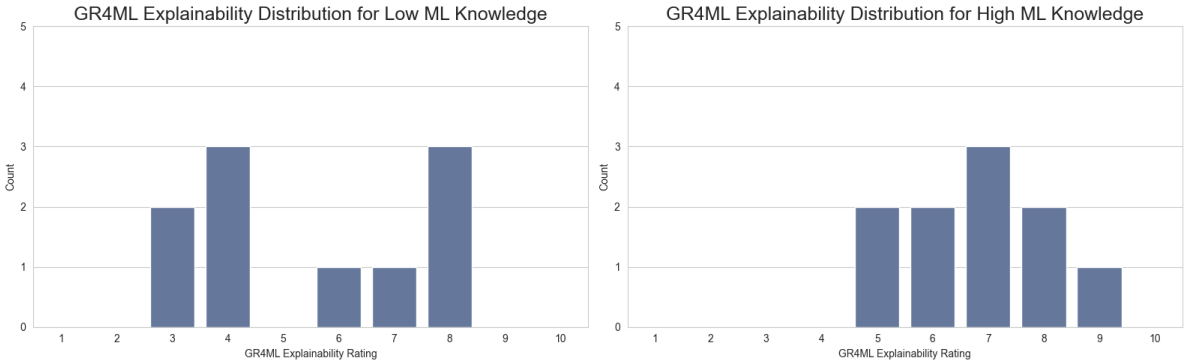


Figure 7.2: Explainability rating distribution from the survey of the GR4ML model for the COMPAS dataset, with on the left participants with low ML knowledge and on the right participants with high ML knowledge.

7.3.1. Results for GR4ML

Figure 7.1 shows the ratings of explainability for the GR4ML model. We can also look at the explainability based on ML knowledge and split the participants into two groups. This split can indicate whether there is a difference in perceived explainability based on different ML knowledge. The first group will be referred to as “low ML knowledge” and are the participants who answered that they are not at all familiar or somewhat familiar with ML models. The second group has “high ML knowledge” and are the participants who answered that they are familiar or very familiar. Both groups consist of ten participants. The split explainability ratings can be seen in Figure 7.2. We can take the average of both the combined view as well as the separated views to compare between the groups. The combined view has an average of 6.15, the low ML knowledge of 5.5, and the high ML knowledge of 6.8. On average, the high ML knowledge group scored slightly higher, although there are more extremely high answers for the group with low ML familiarity.

When asked what the participants liked about the visualization in an open-ended question, most responded with an answer about the split views, making the flow of the system easy to identify. It was also mentioned multiple times that this is especially the case for the data scientists in the data preparation view.

On the downside, the participants believe that the analytics design view is not specific enough. It is mentioned that it is unclear how the indicators impact the goals and what they actually mean. It is mentioned multiple times that more text could resolve this issue. Another point is that a lot is going on in each view, such as large legends, and the arrows in each view are different.

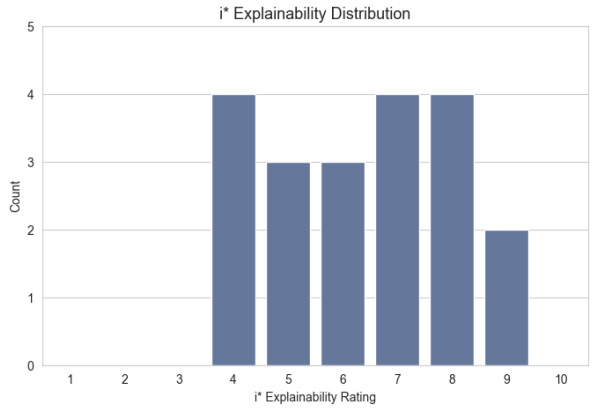


Figure 7.3: Explainability rating distribution from the survey of the i* model for the COMPAS dataset.

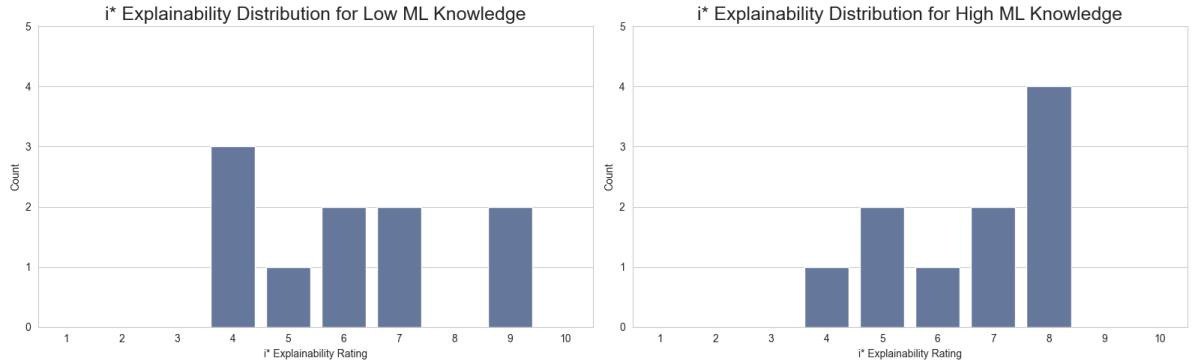


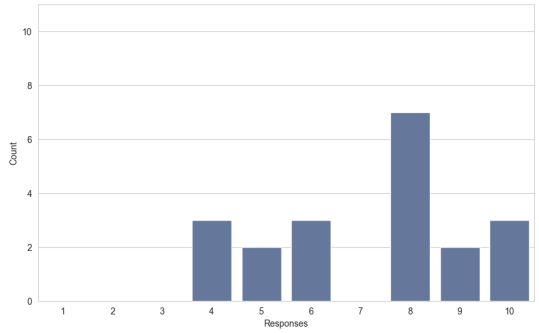
Figure 7.4: Explainability rating distribution from the survey of the i* model for the COMPAS dataset, with on the left participants with low ML knowledge and on the right participants with high ML knowledge.

7.3.2. Results for i*

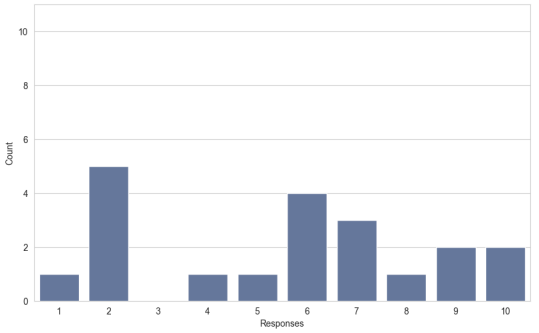
The explainability ratings for the i* model can be seen in Figure 7.3. In this figure, the overall ratings appear to almost follow a uniform distribution between four and nine, with an average score of 6.35. When again looking at the split of ML knowledge in Figure 7.4, it is visible that participants with high ML knowledge score this model higher (average of 6.6) compared to participants with low ML knowledge (average of 6.1).

In the open-ended question on what participants liked about the i* conceptual model, they indicated that they liked the clear structure of the way stakeholders are connected to each other. The other aspect they liked was the use of more text in the requirements.

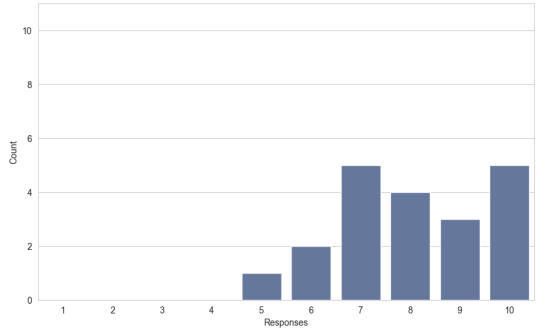
The main thing participants would change in this visualization is the directional dependency links. It is mentioned that it is unclear what additional information this direction adds. Some further points are the legend, which contains unused elements, and the low resolution of the image.



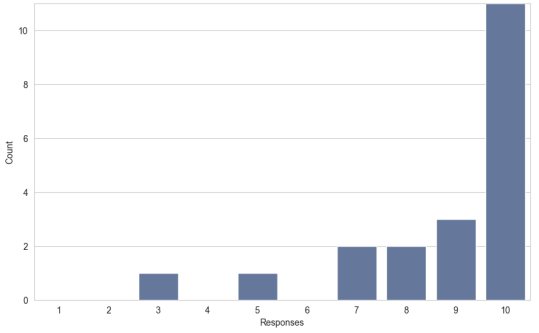
(a) Requirement 1: High risk prediction



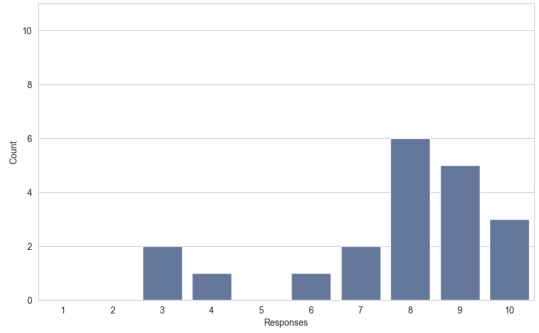
(b) Requirement 2: Missing data handling



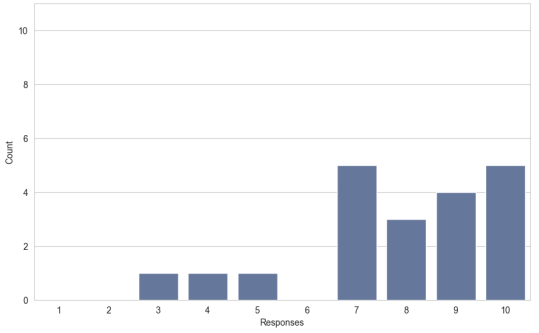
(c) Requirement 3: Input stability



(d) Requirement 4: Explanation of risk scores



(e) Requirement 5: Handling biased data



(f) Requirement 6: Raw probability scores

Figure 7.5: Participant ratings for the six BDD requirements, using a “given-when-then” structure. These requirements correspond with questions 22 to 32 from the survey in steps of 2.

7.4. Results for BDD Requirements

Our survey asked participants to rate six “given-when-then” requirements (see Table 7.1) in terms of explainability on a scale of one (very unexplainable) to ten (very explainable) and provided optional comments justifying their scores. Afterward, we tasked them with creating such a BDD requirement themselves. In the next two sections, we will examine the ratings of these requirements and then show the results of the BDD requirements that the participants designed.

7.4.1. Rating of Six BDD Requirements

The responses of explainability to each of these requirements are shown in Figure 7.5. Requirement 2 has the lowest scores with an average of 5.55, while requirements 3 and 4 score much higher with an average of 8.05 and 8.75, respectively. The other three requirements have average scores ranging from 7.20 to 7.85.

For requirement 2, there are two common answers on why the participants scored it low. One being

that it is unclear how this requirement could be realized. The other being that even though what it meant with the requirement is clear, having this as a requirement reduces explainability. These comments match the intended aspect of this requirement.

On the other hand, requirement 4 is liked by participants because it is a concise, clear, and necessary requirement. Requirement 3 also received the same positive feedback, but scored slightly lower because participants indicated that the term “meaningful” is unclear. The compact aspect of both requirements was intended, but just like requirement 2, the phrasing of words needs to be precise.

The mid-range requirements (1, 5, and 6) give further understanding of the explainability of a BDD scenario. Requirement 1 integrated precision and false-positive constraints into a single requirement. Participants appreciated having numeric thresholds, but felt that multiple statements should be split into multiple requirements.

For requirements 5 and 6, it should be noted that they received only 13 and 12 justifications, respectively. These justifications were also generally shorter (for example, “clear”, “great”, and “Understand”). These are a lot fewer responses and are much shorter than the first two requirements, each of which received 17 responses and were all full sentences. This is probably due to these being the final open-ended questions in a survey, of which the median response time was 21:30 minutes.

In requirement 5, participants appreciated the explicit fairness trigger. On the other hand, two participants rated this lower and mentioned that the normalization method should be specified. These comments do not directly relate to our aspects for this requirement.

The participants who rated requirement 6 low clarified that having an absolute value would remove necessary reasoning for a score, which matches with our aspect of it. However, the participants who rated it higher do mention that the expectation of this requirement is clear.

Overall, justifications for the ratings of the participants often match the set aspects of the requirements. Aspects such as compact requirements and quantifiable criteria contribute to higher explainability ratings, whereas intertwined and ambiguously phrased requirements reduce the perceived explainability.

7.4.2. Creating a BDD Requirement

Out of the twenty participants, nineteen provided us with a self-made BDD requirement related to fairness. All responses as provided by the participants[1] are shown in Table 7.7. It can be seen that most of these requirements use exactly the “given-when-then” structure and consist of three short sentences. Furthermore, we analyzed these requirements by their participant number (nr. column), based on a grouping we made below.

First, we can filter out those requirements we deem to be unclearly posed. We see that respondents 3 and 6 assume a probability score, next to the risk score, which is generally not the case for ML systems. Respondent 8 did not provide a requirement; the requirement of participant 15 was incomplete; and the requirement of participant 16 was ambiguously phrased.

Next, four participants (1, 2, 10, 18) provided a similar requirement that can be seen as robust. All mentioned that when two cases have similar characteristics, they should be classified the same. Participants 13 and 14 generalized this requirement to a group level, but participant 13 specified this even further by clarifying a fairness metric for false positives having to be close to each other.

Then there are participants 9 and 12 who want to warn the judge when certain data features seem to influence the results too much. There is also participant 11, who says not to let the system score when there is not enough data present for an individual. All three can therefore be seen as warnings.

Participants 4 and 19 both mentioned that the classification becomes fairer when the data scientist ac-

[1] No editing or spelling correction has been made on the original text

counts for a known bias. Therefore, we can group these together as fairness by accounting for biases.

Participants 5 and 20 also mentioned a way for the classification to become fairer, this time by feature selection. By only taking crime-related features into account, and for the other features, only use them when it can be argued.

Lastly, there are participants 7 and 17. Participant 17 mentions not to use features such as race and sex. However, as mentioned by Interviewee 2 in Section 5.7, features are generally correlated, so removing this feature may not improve fairness for the classification. Similarly, participant 7 says that the classification should function as normal when the data is missing. Although both requirements are understandable and executable, they may not improve fairness and are grouped as ineffective.

Table 7.7: BDD requirements as written by the survey participants on the topic of fairness and using the “given-when-then” format.

nr.	Answer to survey question: “Can you now try to give such a requirement for Fairness, using this BDD approach of given-when-then.”
1	Given two cases with the exact same input except one input parameter. When the judge would look at the risk scores. Then he/she be able to understand what caused the difference in the risk scores or caused them to be the same.
2	Given the compas model is about to give a high risk score, when a similar case in the past turned out to not be of high risk, then the score should be flagged to show this difference.
3	Given a high risk score, when the probability score is low (e.g. below 0.5), then a warning should be given.
4	Given the system is not always fair, when the data scientist tries to make it more fair, then it should be possible to adjust the model and for difference to he explainable
5	Given the system is asked to provide a risk score for a suspect When the score is provided Then protected characteristics (e.g. gender, race, age) should not have been taken into account unless a good reason exists, which should then be provided in the explanation (e.g. age >70, therefore unlikely to commit violent offences)
6	Given the system provides a risk score. When the pprobability of that risk score is below a threshold (e.g. 0.80). Then the risk score should not be informed to the judge. And the judge makes the judgement without the system.
7	Given that the system does not have all the criminal data of a person. When the risk calculation is made. Then the person should only be judge on the data present.
8	-
9	Given the model shows weights for all user profile components When a high-risk score is the outcome Then give a fairness warning if certain components (like race or neighbourhood) are over a predefined threshold
10	Given two individuals. When they have the same input data, except a sensitive attribute (e.g. ethnicity). Then the system should not classify them differently.
11	Given the fact that such a system is used on people in a precarious situation When the system does not have enough information to operate reliably Then it should refuse to score an individual.
12	Given that the system makes predictions about members of minority and majority groups based on some protected attribute, when the decision is issued, then the judge should be informed to what extent the protected attribute influenced the prediction
13	Given a dataset containing defendants from different racial groups who have no prior offenses and are charged with similar crimes, When the COMPAS model predicts the risk of recidivism for these defendants, Then the false positive rate for each racial group should not differ by more than 5 percentage points (e.g., if the false positive rate for white defendants is 15%, it should be between 10% and 20% for Black and Hispanic defendants).
14	Given that the COMPAS model has been trained on a dataset that includes diverse demographic groups, When the system generates a risk score for a defendant, Then the risk score should not be significantly biased towards any particular demographic group (e.g., race, gender, socioeconomic status).
15	Given that there are circumstances that can affect the sentencing, When the data is input, Then the system can account for those for
16	Given a bies towards one group, when someone from any other group is convicted and shown a low risk then should they be reviewed as well?
17	Given that the training data contains a Bias based on gender. When the COMPAS system calculates risk scores. Then gender data is not taken into account when calculating risk scores.
18	Given that two individuals belong to socially defined different groups, when they share equal insensitive personal attributes, then the model should output the same risk score.
19	given the system outputs a score, when the bias of the support group is accounted for, then the judge can issue a sentence
20	Given the data is containing biasses, When generating a score, Then use only the parameters involved directly in crime patterns

8

Conclusion

This research examined the use of `Behavior Driven Development (BDD)` and `Goal-Oriented Requirements Engineering (GORE)` in identifying and visualizing requirements for aspects of explainability and trust in ML system development. We used a mixed-methods approach with expert interviews and a survey to answer three research questions.

RQ 1: To what extent can Behavior-Driven Development be used to identify requirements for "Explainability & Trust"?
It was found that clear and concise human-readable requirements, such as the "given-when-then" BDD scenarios, can make ML system development more explainable. This can be seen in our results in Chapter 7 and adds quantitative information to the limited available research for BDD [7]. Furthermore, for the questions on rating BDD requirements, there are two requirements that the participants found more explainable, three that the participants found to have a decent level of explainability, and one that scored low. As discussed in Section 7.4.1, requirements 3 and 4 are concise, and people agree that this is how the system should behave. On the other hand, requirement 2 had a low rating, the main reason being that it makes the system biased.

These justifications for the ratings make us conclude two things for this research question. (1) Participants prefer short and direct requirements over requirements that may contain multiple parts. One reason for this can be that longer sentences lead people to believe that the requirement can be split up, as well as the requirement possibly becoming vaguer with more text. The other conclusion is that (2) participants rated requirements lower if they disagreed with their content or would not know how to implement it. This means that the requirement may be very clear, yet it was rated lower. This limitation is further discussed in Section 9.1.

RQ 2: What type of tests can be used to encapsulate different needs of stakeholders?
Our expert interviews and survey responses suggest that the blend of functional and non-functional requirements comprehensively addresses stakeholder needs. Functional tests can directly capture accuracy and robustness (e.g., high-risk predictions must be correct at least 90% of the time), while non-functional BDD scenarios can justify requirements for "explainability" ("provide a human-readable justification") and "fairness" ("normalize outcomes across protected groups"). Interviewees noted the importance of edge-case tests, such as the absence of some data or slight variations of data inputs, to ensure model stability in real-world scenarios. Thus, balancing BDD tests with the requirements of accuracy, explanation, robustness, and fairness adequately addresses diverse stakeholder needs in ML system development.

RQ 3: How well can we visualize different stakeholder needs regarding "Explainability" using Goal-Oriented Requirements Engineering?
For our last research question, we started by looking at how current research models these components that can be seen as human-centered. Although UML is often used because it is an industry standard

practice that is easy to understand, it was advised to model these using GORE [1], since GORE is able to show stakeholder needs.

Our survey indicated that participants prefer requirements to be visualized using visual flowcharts over other options, such as interactive tools and technical reports. The use of GORE would therefore be able to improve explainability. We tested the explainability through two GORE languages, GR4ML and i*.

In our results, we can see that i* reflects the needs of most stakeholders, while GR4ML reflects the needs of the judge, but neglects the needs of the suspect. This confirms that GR4ML is designed to align with business goals [32]. This focus on business goals unfortunately, means that it does not improve explainability toward all stakeholders.

i* on the other hand shows more promising results, as it seems to be more explainable to different stakeholders, as survey participants like the direct connections between the stakeholders, as well as text in the visualization to give more explanations. This becomes even more evident when looking at the ratings of participants with ML knowledge, who score this visualization much higher than participants with low ML knowledge. Unfortunately, we did not see the same trend in the GR4ML visualization.

In general, aspects such as clear dependencies, information on roles, and textual requirements can improve the explainability. However, aspects such as deeper explanations on how the requirements affect the system should be added to the conceptual models, and irrelevant or overcomplicated legends should be modified to improve the explainability of the conceptual model. In addition, the responses highlight concerns about fairness and bias in ML models. The participants emphasized the need for transparency in handling missing data, minor variations in input, and historical biases. This aligns with the larger goal of ensuring ethical ML systems.

9 Discussion

Our results show that Behavior-Driven Development (BDD) requirements and Goal-Oriented Requirements Engineering (GORE) modeling tools can improve explainability and trust in Machine Learning (ML) system development. In addition to the results and conclusions, there are still some limitations and recommendations for future research, which we discuss in this chapter. After that, we will mention ethical considerations and the adverse impact of our conclusions.

9.1. Limitation

From a modeling perspective, both GR4ML and i* have major shortcomings in their representation of complex ML requirements. First, for i*, there appears to be no consistent language for it [45, 38], making it difficult to compare different visualizations. Although GR4ML is better documented, there is less existing research using this language, making it difficult to diverge from the standard design [32]. Furthermore, the two models used in this research were made by computer scientists, not professional designers. This could limit the understanding for people without a computer science background.

Next, for the explainability ratings of the six BDD requirements in our survey, differentiating between textually poorly structured requirements and those that are undesirable or difficult to implement in a system, turned out to be complicated. The low explainability scores that the participants gave could stem from either of the two reasons. This ambiguity complicates the interpretation of our results, as we cannot adequately classify low ratings as clear requirements or as practical constraints.

Other limitations are related to survey design. In our survey, we did not assess participants' English proficiency, which may have influenced their understanding of both the BDD scenarios and the GORE diagrams. As a result, it is difficult to distinguish lower explainability ratings due to language barriers from those reflecting inherent shortcomings in our requirements.

9.2. Future Research

Even though GR4ML and i* show promising ways to visualize complex requirements, such as explainability, there remains a need for a unified conceptual modeling tool. Studies point out that Requirements Engineering for Machine Learning (RE4ML) frameworks often ignore cohesive modeling standards and lifecycle considerations, including audit log submission or model retraining triggers, in their diverse modeling notations [36, 19]. Future work should not only focus on developing a new modeling tool but also explore the aspects of multiple GORE languages based on specific needs. For example, a project that should be disclosed only to stakeholders could benefit more from GR4ML.

Furthermore, we tested the explainability of the models only on people who have ML knowledge. Other stakeholders for systems like the COMPAS dataset, such as the defendant, should be taken directly into account. This could be done through multi-stakeholder XAI workshops [6] and would improve explainability and trust of the system to people who are not directly involved in the development process.

In addition, systematic experiments could compare BDD against other techniques to find specific benefits and trade-offs. One comparison could be to test the requirements elicitation process against Data-Driven Development. Another would be to test a final model of BDD to a Test-Driven Development approach. Controlled user studies could measure these results and find which aspects of each development approach improve explainability.

9.3. Ethical Considerations

As noted by Henrich et al. [21], the majority of participants in scientific research are drawn from `Western, Educated, Industrialized, Rich, and Democratic (WEIRD)` societies. This is also the case for the interviews and the survey in this research. This demographic homogeneity raises concerns about possible, unintentional bias in the elicitation of requirements and perceptions of explainability for our working cases. Fortunately, these requirements did not have a large impact on the intentions of the designs, but for future research, these concerns should be taken into account to ensure broader explainability and trust.

9.4. Adverse Impact

Although our research intends to improve the transparency and accountability of the ML lifecycle, there is a chance that organizations implement our recommendations superficially, engaging only in surface-level discussions with stakeholders. Poorly constructed BDD requirements or GORE diagrams might provide a false sense of confidence while concealing serious problems related to fairness or safety within the model. Although thinking about requirements will always be better than not articulating anything, intentionally creating false confidence could prove to be more dangerous when ML systems make final decisions in high-stakes fields.

In addition, as these approaches gain popularity, tool developers may sell simplified, commercially available templates that are not fit for the complexities of high-stakes domains. Imposing these negative consequences can be avoided by pairing any modeling framework with robust training, continuous audits, and governance policies.

Bibliography

[1] Khlood Ahmad et al. “Requirements engineering framework for human-centered artificial intelligence software systems”. In: *Applied Soft Computing* 143 (2023), p. 110455.

[2] G Araujo et al. “Professional Insights into Benefits and Limitations of Implementing MLOps Principles. arXiv 2024”. In: *arXiv preprint arXiv:2403.13115* (2024).

[3] Rob Ashmore, Radu Calinescu, and Colin Paterson. “Assuring the machine learning lifecycle: Desiderata, methods, and challenges”. In: *ACM Computing Surveys (CSUR)* 54.5 (2021), pp. 1–39.

[4] Hrvoje Belani, Marin Vukovic, and Željka Car. “Requirements engineering challenges in building AI-based complex systems”. In: *2019 IEEE 27th International Requirements Engineering Conference Workshops (REW)*. IEEE. 2019, pp. 252–255.

[5] Johan Bergelin and Per Erik Strandberg. “Industrial requirements for supporting AI-enhanced model-driven engineering”. In: *Proceedings of the 25th International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings*. 2022, pp. 375–379.

[6] Umang Bhatt et al. “Machine learning explainability for external stakeholders”. In: *arXiv preprint arXiv:2007.05408* (2020).

[7] Leonard Peter Binamungu and Salome Maro. “Behaviour driven development: A systematic mapping study”. In: *Journal of Systems and Software* 203 (2023), p. 111749.

[8] Manal Binkhonain and Liping Zhao. “A review of machine learning algorithms for identification and classification of non-functional requirements”. In: *Expert Systems with Applications: X* 1 (2019), p. 100001.

[9] Victoria Clarke and Virginia Braun. “Thematic analysis”. In: *The journal of positive psychology* 12.3 (2017), pp. 297–298.

[10] Maheswaree Kissoon Curumsing et al. “Emotion-oriented requirements engineering: A case study in developing a smart home system for the elderly”. In: *Journal of systems and software* 147 (2019), pp. 215–229.

[11] Fabiano Dalpiaz and Nan Niu. “Requirements engineering in the days of artificial intelligence”. In: *IEEE software* 37.4 (2020), pp. 7–10.

[12] Vincenzo De Martino and Fabio Palomba. “Classification and challenges of non-functional requirements in ML-enabled systems: A systematic literature review”. In: *Information and Software Technology* (2025), p. 107678.

[13] Michal Doležel. “Defining TestOps: collaborative behaviors and technology-driven workflows seen as enablers of effective software testing in DevOps”. In: *Agile Processes in Software Engineering and Extreme Programming–Workshops: XP 2020 Workshops, Copenhagen, Denmark, June 8–12, 2020, Revised Selected Papers 21*. Springer. 2020, pp. 253–261.

[14] Chiu Pang Fung et al. “Towards accountability driven development for machine learning systems”. In: *CEUR Workshop Proceedings*. Vol. 2894. CEUR-WS. 2021, pp. 25–32.

[15] Görkem Giray. “A software engineering perspective on engineering machine learning systems: State of the art and challenges”. In: *Journal of Systems and Software* 180 (2021), p. 111031.

[16] Martin Glinz. “On non-functional requirements”. In: *15th IEEE international requirements engineering conference (RE 2007)*. IEEE. 2007, pp. 21–26.

[17] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. “Explaining and harnessing adversarial examples”. In: *arXiv preprint arXiv:1412.6572* (2014).

[18] Abhimanyu Gupta, Geert Poels, and Palash Bera. “Generating multiple conceptual models from behavior-driven development scenarios”. In: *Data & Knowledge Engineering* 145 (2023), p. 102141.

[19] Umm-e- Habiba et al. “How mature is requirements engineering for AI-based systems? A systematic mapping study on practices, challenges, and future research directions”. In: *Requirements Engineering* (2024), pp. 1–34.

[20] Petra Heck and Gerard Schouten. “Defining Quality Requirements for a Trustworthy AI Wildflower Monitoring Platform”. In: *2023 IEEE/ACM 2nd International Conference on AI Engineering–Software Engineering for AI (CAIN)*. IEEE. 2023, pp. 119–126.

[21] Joseph Henrich, Steven J Heine, and Ara Norenzayan. “The weirdest people in the world?” In: *Behavioral and brain sciences* 33.2-3 (2010), pp. 61–83.

[22] Hans-Martin Heyn et al. “Requirement engineering challenges for ai-intense systems development”. In: *2021 IEEE/ACM 1st Workshop on AI Engineering-Software Engineering for AI (WAIN)*. IEEE. 2021, pp. 89–96.

[23] E Hull, K Jackson, and J Dick. “Defining requirement engineering”. In: *Requirements engineering, Springer Science & Business Media, London* (2010), pp. 6–8.

[24] *ICT Specialists in employment*. May 2024. URL: `https://ec.europa.eu/eurostat/statistics-explained/index.php?title=ICT_specialists_in_employment` (visited on 05/07/2025).

[25] Fuyuki Ishikawa and Yutaka Matsuno. “Evidence-driven requirements engineering for uncertainty of machine learning-based systems”. In: *2020 IEEE 28th International Requirements Engineering Conference (RE)*. IEEE. 2020, pp. 346–351.

[26] Mohamad Kassab, Joanna DeFranco, and Valdemar Graciano Neto. “An empirical investigation on the satisfaction levels with the requirements engineering practices: Agile vs. waterfall”. In: *2018 IEEE International Professional Communication Conference (ProComm)*. IEEE. 2018, pp. 118–124.

[27] Maximilian A Köhl et al. “Explainability as a non-functional requirement”. In: *2019 IEEE 27th International Requirements Engineering Conference (RE)*. IEEE. 2019, pp. 363–368.

[28] Jeff Larson et al. *How we analyzed the compas recidivism algorithm*. May 2016. URL: `https://www.propublica.org/article/how-we-analyzed-the-compas-recidivism-algorithm`.

[29] Lucy Ellen Lwakatare et al. “From a data science driven process to a continuous delivery process for machine learning systems”. In: *Product-Focused Software Process Improvement: 21st International Conference, PROFES 2020, Turin, Italy, November 25–27, 2020, Proceedings 21*. Springer. 2020, pp. 185–201.

[30] Walid Maalej, Yen Dieu Pham, and Larissa Chazette. “Tailoring requirements engineering for responsible AI”. In: *Computer* 56.4 (2023), pp. 18–27.

[31] Walid Maalej et al. “Toward data-driven requirements engineering”. In: *IEEE software* 33.1 (2015), pp. 48–54.

[32] Soroosh Nalchigar, Eric Yu, and Karim Keshavjee. “Modeling machine learning requirements from three perspectives: a case report from the healthcare domain”. In: *Requirements Engineering* 26 (2021), pp. 237–254.

[33] Tyron Offerman et al. “A study of adoption and effects of DevOps practices”. In: *2022 IEEE 28th International Conference on Engineering, Technology and Innovation (ICE/ITMC) & 31st International Association For Management of Technology (IAMOT) Joint Conference*. IEEE. 2022, pp. 1–9.

[34] Dhirendra Pandey, Ugrasen Suman, and A Kumar Ramani. “An effective requirement engineering process model for software development and requirements management”. In: *2010 International Conference on Advances in Recent Technologies in Communication and Computing*. IEEE. 2010, pp. 287–291.

[35] Annibale Panichella and Cynthia CS Liem. “What are we really testing in mutation testing for machine learning? a critical reflection”. In: *2021 IEEE/ACM 43rd International Conference on Software Engineering: New Ideas and Emerging Results (ICSE-NIER)*. IEEE. 2021, pp. 66–70.

[36] Zhongyi Pei et al. “Requirements engineering for machine learning: A review and reflection”. In: *2022 IEEE 30th International Requirements Engineering Conference Workshops (REW)*. IEEE. 2022, pp. 166–175.

[37] Karthik Pelluru. “Advancing software development in 2023: the convergence of MLOps and DevOps”. In: *Advances in Computer Sciences* 6.1 (2023), pp. 1–14.

[38] Fabio Penha et al. “A Proposed Textual Model for i-Star.” In: *iStar*. 2016, pp. 7–12.

[39] Fábio Penha et al. “Actor’s social complexity: a proposal for managing the iStar model”. In: *Journal of Software Engineering Research and Development* 6 (2018), pp. 1–30.

[40] Khansa Rasheed et al. “Explainable, trustworthy, and ethical machine learning for healthcare: A survey”. In: *Computers in Biology and Medicine* 149 (2022), p. 106043.

[41] Shexmo Santos et al. “Using Behavior-Driven Development (BDD) for Non-Functional Requirements”. In: *Software* 3.3 (2024), pp. 271–283.

[42] David Sculley et al. “Hidden technical debt in machine learning systems”. In: *Advances in neural information processing systems* 28 (2015).

[43] Rakshith Subramanya, Seppo Sierla, and Valeriy Vyatkin. “From DevOps to MLOps: Overview and application to electricity market forecasting”. In: *Applied Sciences* 12.19 (2022), p. 9851.

[44] Sumanth Tatineni. “Recommendation Systems for Personalized Learning: A Data-Driven Approach in Education”. In: *Journal of Computer Engineering and Technology (JCET)* 4.2 (2020).

[45] Miguel A Teruel et al. “CSRML: a goal-oriented approach to model requirements for collaborative systems”. In: *Conceptual Modeling–ER 2011: 30th International Conference, ER 2011, Brussels, Belgium, October 31-November 3, 2011. Proceedings 30*. Springer. 2011, pp. 33–46.

[46] Theo Thesing, Carsten Feldmann, and Martin Burchardt. “Agile versus waterfall project management: decision model for selecting the appropriate approach to a project”. In: *Procedia Computer Science* 181 (2021), pp. 746–756.

[47] Ehsan Toreini et al. “The relationship between trust in AI and trustworthy machine learning technologies”. In: *Proceedings of the 2020 conference on fairness, accountability, and transparency*. 2020, pp. 272–283.

[48] Benjamin Van Giffen, Dennis Herhausen, and Tobias Fahse. “Overcoming the pitfalls and perils of algorithms: A classification of machine learning biases and mitigation methods”. In: *Journal of Business Research* 144 (2022), pp. 93–106.

[49] Axel Van Lamsweerde. “Goal-oriented requirements engineering: A guided tour”. In: *Proceedings fifth ieee international symposium on requirements engineering*. IEEE. 2001, pp. 249–262.

[50] Hugo Villamizar, Tatiana Escovedo, and Marcos Kalinowski. “Requirements engineering for machine learning: A systematic mapping study”. In: *2021 47th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*. IEEE. 2021, pp. 29–36.

[51] Andreas Vogelsang and Markus Borg. “Requirements engineering for machine learning: Perspectives from data scientists”. In: *2019 IEEE 27th International Requirements Engineering Conference Workshops (REW)*. IEEE. 2019, pp. 245–251.

[52] Zhiyuan Wan et al. “How does machine learning change software development practices?” In: *IEEE Transactions on Software Engineering* 47.9 (2019), pp. 1857–1871.

[53] Tannaz Zameni et al. “From BDD scenarios to test case generation”. In: *2023 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*. IEEE. 2023, pp. 36–44.

[54] Jing Zhang et al. “A novel data-driven stock price trend prediction system”. In: *Expert Systems with Applications* 97 (2018), pp. 60–69.

A

Working Examples of GORE conceptual models

To get a better understanding of what such conceptual models look like, we show existing examples of the ones used in this research: GR4ML and i*.

A.1. Working Example of GR4ML

As stated above, this GR4ML framework uses three modeling views, which together serve as the viewpoints of business people, data scientists, and data engineers [32]. We have taken a simple banking model from their website as a working example.

Business View The business view shows how a case worker aims to process applications and is measured by the average resolution time indicator. This is supported by answering decision questions on the credit application, which in turn is supported by a question goal about the risk score from the predictive model.

Analytics Design View This simplified analytics design view shows that decision trees and support vector machines are used as algorithms for the classification of an applicant. It also shows measurable indicators for precision and accuracy and soft goals for interpretability and tolerance for missing values.

Data Preparation View This data preparation view shows that filter and join operations are used, among others, to generate the application profile.

Linking the views Figure A.1 shows that the applicant profile is required to perform the classification of the applicant profiles, which will then generate a credit risk predictive model.

A.2. Working Example of i*

In this conceptual model, the stakeholders are all represented directly as actors. The insurance company and the physician are depicted as agents because they have a physical representation and are stand-alone. The patient is an actor who has multiple connections, but these are replaced by a domain for the connections of the patient.

In this simplified model, you can start at any actor and walk around the graph following the links. It is visible, for example, that the insurance company depends on the resource of premium payment, which in turn depends on the patients' task to buy insurance. The goal of a patient is to get well when he is treated, where being treated is a task of a physician.

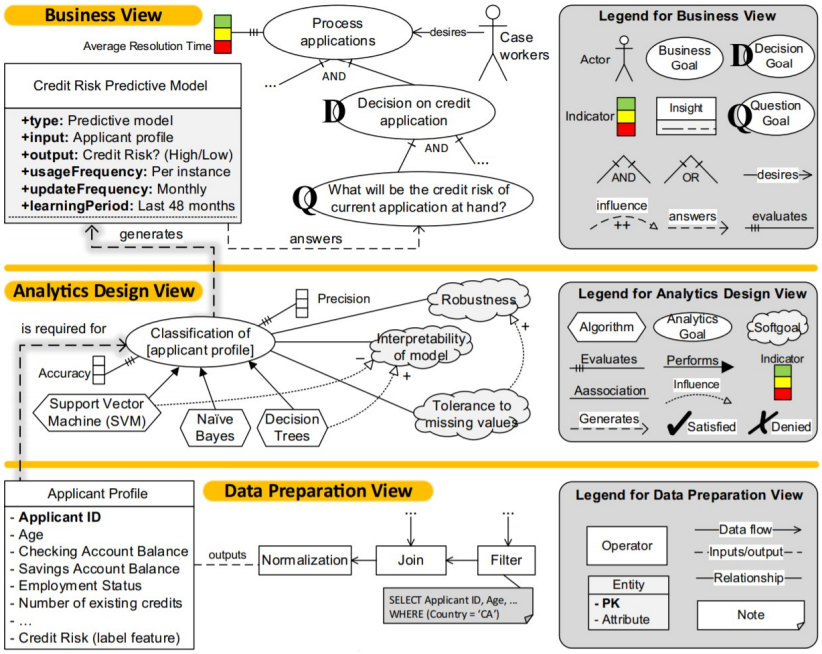


Figure A.1: Working example of the GR4ML conceptual framework, adapted from [32].

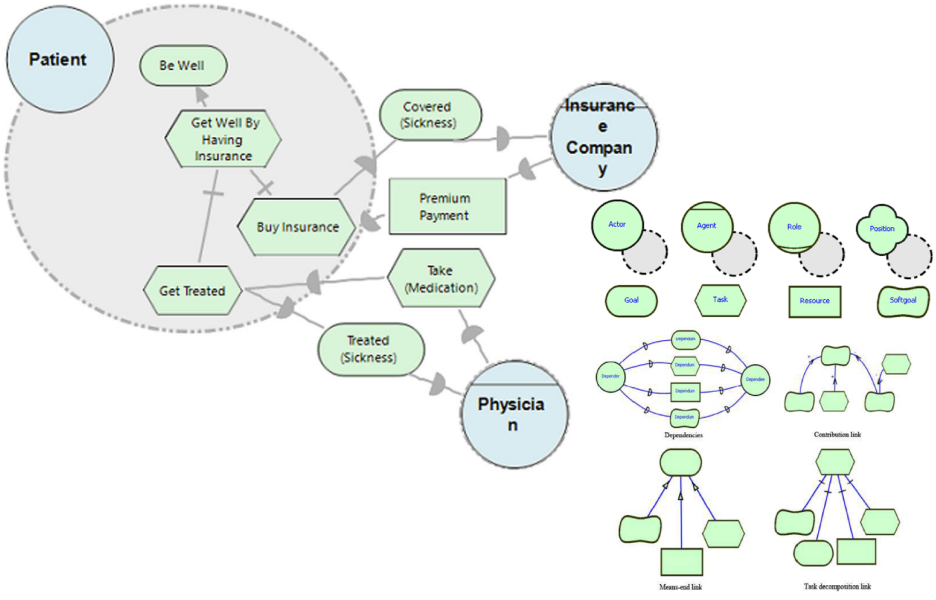


Figure A.2: Working example of the i* conceptual framework, adapted from [39].

B

Interview Questions

Introduce research objectives (5 min)
Thank you for participating in this interview. I will first briefly explain our research purpose and the plan for the interview today, and then we can get right into it.
As communicated earlier via email, I would like to talk today about Requirements Engineering for Machine Learning. I would namely like to research if we can improve setting requirements for black box systems, by using Behavior-Driven Development. But in order to set initial requirements, I first need to understand how these are developed in practice. In this way, I can create a conceptual model for 2 use cases and see how good the explainability becomes using BDD.
Therefore, my research questions for these interviews are as follows:

1. What type of tests can be used to encapsulate different needs of stakeholders?
2. To what extent can Behavior-Driven Development be used to identify requirements for "Explainability & Trust"?

So, how will the interview look like today? We will start the interview with a short inquiry into your background with Requirements Engineering. Then we will go over what is deemed important for trust, safety and explainability. Next, we will cover engineering practices. Lastly, which is the biggest part, we go over 2 use cases for RE. The whole interview should take approximately 1 hour. If it is okay with you, I will record this interview so that I can transcribe it afterwards in peace and make sure that I have taken the right messages. This transcription will not be shared with others. Any outcomes of this interview will also be pseudonymized. During the interview, you can also always make an "off the record" statement that I will not include in my research. Do you have any questions about my research, the interview, or the informed consent document? Okay, so is it okay that I turn on the recording now?

General info (5 min)

1. What is your job title?
2. How long have you worked in the computer science field?
3. Have you worked with ML models as part of your job?
4. Have you been involved in developing or assessing ML models for high-stakes applications, such as judicial, finance, or medical?
5. How challenging was it for you to set requirements for this system? [1-10]

Trust, Safety and Explainability in ML (15 min)

1. What does "trustworthy AI" mean to you in practice?
2. What factors are essential for building trust in the predictions of a recidivism model? [Transparency, interpretability, fairness, accountability + other]

3. Would you apply the same factors to a loan approval model? [Transparency, interpretability, fairness, accountability + other]
4. Have you applied any transparency methods for a model?
5. In your experience, how much do end-users understand ML models and their limitations? [1-10]

RE and BDD (10 min)

1. Have you used formalized frameworks like RE4HCAI or methodologies such as BDD or DDD in ML projects? [RE4HCAI, CRISP-DM, LIME, SHAP, BDD, DDD]
2. How do you currently approach defining requirements for ML models?
3. With BDD the goal is to write requirements in plain language that all stakeholders can understand. Do you think this is/would be useful in ML development to assist explainability?

Use Cases: COMPAS recidivism and Loan Approval (25 min)
COMPAS
The COMPAS (Correctional Offender Management Profiling for Alternative Sanctions) recidivism algorithm is a tool used in the US criminal justice system to assess the risk of an individual recidivating: reoffending after being previously arrested. According to ProPublica, who used 80k criminal records from Broward County, the model uses 53 data points to determine a risk score of 1 (low risk) to 10 (high risk). This model then showed a strong bias towards African-Americans, but also a bias towards men and younger people.

1. Given COMPAS as a use case, do you think judicial institutions should disclose more about how their recidivism algorithm works?
2. In your view, what are the main ethical concerns when using ML for criminal risk assessment?
3. The COMPAS model has been criticized for its racial bias. How would you assess and mitigate such bias, for example, if we were to remove race, how do you think this would affect the overall output of the model?
4. What methods do you believe are most suitable for evaluating fairness in recidivism prediction models? [Demographic parity, Equalized odds, Equality of opportunity, fairness through unawareness, other]
5. The initial requirements we set were: Accuracy, Robustness, Explainability, Fairness.

    - **Accuracy**: The system must achieve high predictive performance, with metrics such as precision, recall, and the F1 score being above a defined threshold.
    - **Robustness**: Predictions must be stable over time for individuals with similar profiles, ensuring that minor changes in input do not lead to disproportionately large differences in predictions.
    - **Explainability**: The system must provide explanations for individual predictions to enhance explainability and trust among non-technical users.
    - **Fairness**: The system must make predictions without showing bias towards a group, without having a fundamental or ethical foundation to support this bias.

    Would you agree with this list and the definitions, or make amendments to it?
6. How would you weigh these requirements against each other in practice?

Loan Approval
The loan approval dataset is basically what it says. It is a dataset that uses financial data to determine if someone should be approved for a loan or not.

1. Do you think financial institutions should disclose more about how their approval algorithm works?
2. In your view, what are the main ethical concerns when using ML for loan approval assessment?

3. How would you assess and mitigate income bias, for example if we were to remove income, how do you think this would affect the overall output of the model?
4. What methods do you believe are the most suitable for evaluating fairness in loan approval prediction models? [Demographic parity, Equalized odds, Equality of opportunity, fairness through unawareness, other]
5. The initial requirements we set were the same: Accuracy, Robustness, Explainability, Fairness. Would you agree with having the same list here, or make amendments to it?
6. How would you weigh these requirements against each other in practice?

Both

1. How do fairness concerns in financial models compare to those in criminal justice models? Would you recommend different fairness metrics for them?
2. If a model performs well, but misclassified certain groups, how (if so) should this be handled?
3. What limitations, if any, are there in using historical data to train risk assessment models?

C

Invitation Letter for Interviews

Dear ...,

My name is Jaron Rosenberg and I am a master's thesis student at TU Delft, supervised by dr. Cynthia Liem (Multimedia Computing group at the Faculty of Electrical Engineering, Mathematics, and Computer Science).

I am writing to you to inquire about the possibility of scheduling an interview on requirements engineering and best practices of it in your domain.

More specifically, my research focuses on the topic of "Requirements Engineering for Machine Learning". The purpose of this research study is to identify the ability of behavior-driven development (BDD) to set requirements for a black box system. You do not need to be an expert in BDD already. We will be asking you to set and evaluate requirements for the COMPAS recidivism dataset and a loan approval dataset, using BDD.

If you are open to dedicating one hour of your time in the coming weeks for an interview or if you can recommend any of your colleagues to discuss this topic with me, I would be very grateful to receive a message from you at the (email) address above. I will provide you with more details about the interview ahead of time, including a consent form explaining how the outcomes of our discussion would be used in my research.

Thank you for considering my request. If you would like to know anything else before making your decision, please do not hesitate to contact me.

Yours sincerely,
Jaron Rosenberg

D

Survey Questions

Requirements Engineering for Machine Learning

You are being invited to participate in a research study titled “Requirements Engineering for Machine Learning”. This study is being done by Jaron Rosenberg for a Master thesis at the TU Delft and is supervised by Dr. Cynthia Liem.

The purpose of this research study is to identify explainability of a conceptual model, using behaviour-driven development for a black-box system, and will take you approximately *15-20 minutes* to complete. The data will be used and published in the TU Delft repository. We will be asking you both closed- and open-ended questions to evaluate requirements and a conceptual model for the COMPAS recidivism dataset.
As with any online activity the risk of a breach is always possible. To the best of our ability your answers in this study will remain confidential. We will minimize any risks by pseudonymising any personal data, where only age and study experience will be recorded to establish expertise. Your participation in this study is entirely voluntary and you can withdraw at any time. You are free to omit any questions. Since the survey is anonymous, it is not possible to withdraw your answers after submission.

By clicking to the next page you automatically agree to this opening statement.

Vereist

General Information

This section collects demographic data to analyze responses based on background and expertise.

1

What is your age group? *

- ◯ 18-24
- ◯ 25-34
- ◯ 35-44
- ◯ 45-54
- ◯ 54+

2

What gender do you associate with? *

- ◯ Male
- ◯ Female
- ◯ Other
- ◯ Prefer not to say

3

Do you identify as a minority? *

- ◯ Yes
- ◯ No
- ◯ Maybe

4

What's your highest level of education received? *

- ◯ High School
- ◯ University - Bachelor
- ◯ University - Master
- ◯ PhD or higher
- ◯ Andere

5

What's your field of expertise? *

- ◯ Business
- ◯ Computer Science
- ◯ Engineering
- ◯ Finance
- ◯ Law
- ◯ Mathematics
- ◯ Social Siences
- ◯ Andere

6

How familiar are you with Machine Learning models? *

- ◯ Not at all familiar
- ◯ Somewhat familiar
- ◯ Familiar
- ◯ Very familiar
- ◯ Expert

7

Have you heard of Requirements Engineering? *

- ◯ Yes, I am familiar with it
- ◯ I have heard of it but do not fully understand it
- ◯ No, I have never heard of it

8

Have you heard of Behavior-Driven Development? *

- ◯ Yes, I am familiar with it
- ◯ I have heard of it but do not fully understand it
- ◯ No, I have never heard of it

Explainability and BDD in ML models

This section explores how Bahvior-Driven Development (BDD) can improve explainability in Machine Learning (ML) decision-making systems.

9

Would you be more confident using an ML model if it followed clear, human-readable rules before making predictions?

- ◯ Yes
- ◯ No
- ◯ Maybe

10

BDD allows ML models to be developed with real-world scenarios in mind (e.g., "If a loan applicant has repaid past loans and has a stable job, they should be approved"). Would such an approach make ML decisions easier to understand?

- ◯ Yes
- ◯ No
- ◯ Maybe

11

What format would you prefer for a visualization of an ML model?

- ◯ Simple text descriptions of decisions
- ◯ Visual flowcharts showing decision steps
- ◯ Interactive tools where users can test different inputs
- ◯ Technical reports with full model details
- ◯ Andere

COMPAS Recidivism Model

The COMPAS (Correctional Offender Management Profiling for Alternative Sanctions) recidivism algorithm is a tool used in the US criminal justice system to assess the risk of an individual recidivating: reoffending after being previously arrested. According to ProPublica, who used 80k criminal records from Broward County, the model uses 53 data points to determine a risk score of 1 (low risk) to 10 (high risk). This model then showed a strong bias towards African-Americans, but also a bias towards men and younger people.

In the next 2 sections we will ask you questions based on how this algorithm should work.

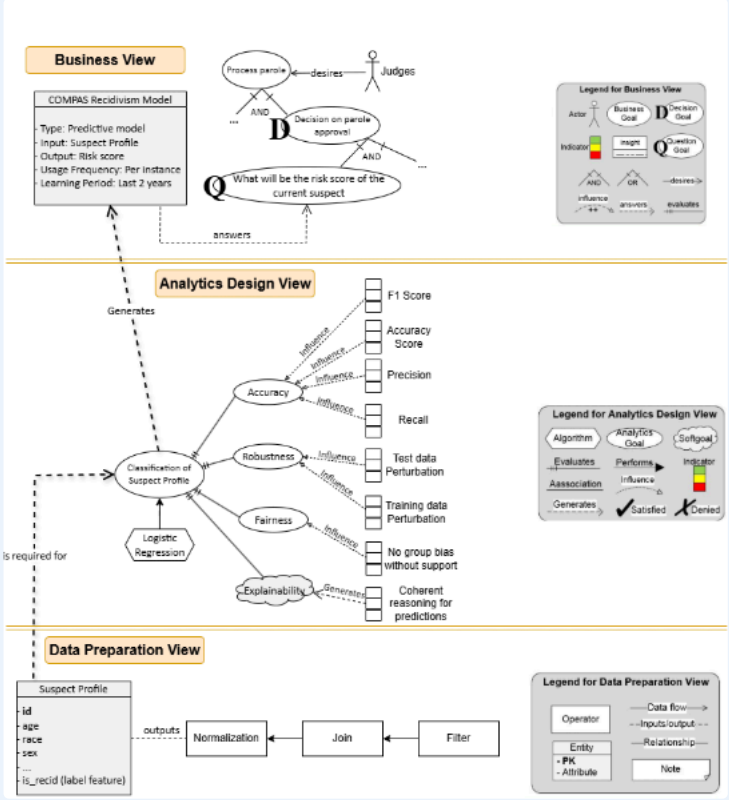
Conceptual visualizations using Goal-Oriented Requirements Engineering

In this section 2 visualizations are shown that are created based on explainability for stakeholders. Your task is to determine the explainability for both visualizations.
- GR4ML: This framework (Goal-Oriented Requirements Engineering for Machine Learning) uses 3 stages to show the flow of a system. First is the 'Data Preperation View' visualizing how the data is processed before it is used. Next is the 'Data Analytics View', which contains the algorithm and requirements for the system. Lastly, is the 'business view' at the top and this represents the usage of the algorithm into the decision-making process.
More information on the GR4ML framework can be found at: https://www.cs.toronto.edu/~soroosh/gr4ml_introduction.html
- i*: The i* (i-star) framework focusses on the intentional (why?), social (who?), and strategic (how?) of a system. The idea behind it is to visualize each stakeholder and their direct workings and requirements to both new users and experienced users.
More information on the i* model can be found at: https://link.springer.com/chapter/10.1007/978-3-642-24606-7_4

12

GR4ML is a conceptual modeling framework and consists of three views: the Business View, the Analytics Design View, and the Data Preparation View. Each view addresses different aspects of the system to address different stakeholder needs in a single visualization. This framework is flexible in design in that it can be built top-down, bottom-up, or hybrid. This visualization was made bottom-up.
First, the data preparation view was made and consisted of filtering and normalizing the data by the data scientist to achieve the final data used in the model.
Next, the analytics view uses this data and runs a logistic regression on it to generate the final prediction model for the business view.
These results, together with a judge's need to sentence as many people as possible correctly, gives a risk score.
The analytics view is the most interesting part for this research, as it contains elements to evaluate the classification process. The classification is evaluated on the four requirements we set, which each contain one or more indicators. These indicators are tests that can be run at any time to evaluate the system and can indicate a pass (green), a warning (yellow), or a failure (red). Explainability is set as a soft goal as it does not have a metric for an automated test. Here, a soft goal is a goal that is considered desirable but is not strictly measured through automated metrics.

(If the image is pixelated you can view it here: https://photos.app.goo.gl/WEX2b4GfBB3wL8zs6)



13

Which stakeholders' needs are being met? *

- [ ] Suspect
- [ ] Government
- [ ] Judge
- [ ] Data Scientist

14

How explainable would you rate this recidivism model visualization? *

1	2	3	4	5	6	7	8	9	10

15

What do you like about this visualization?

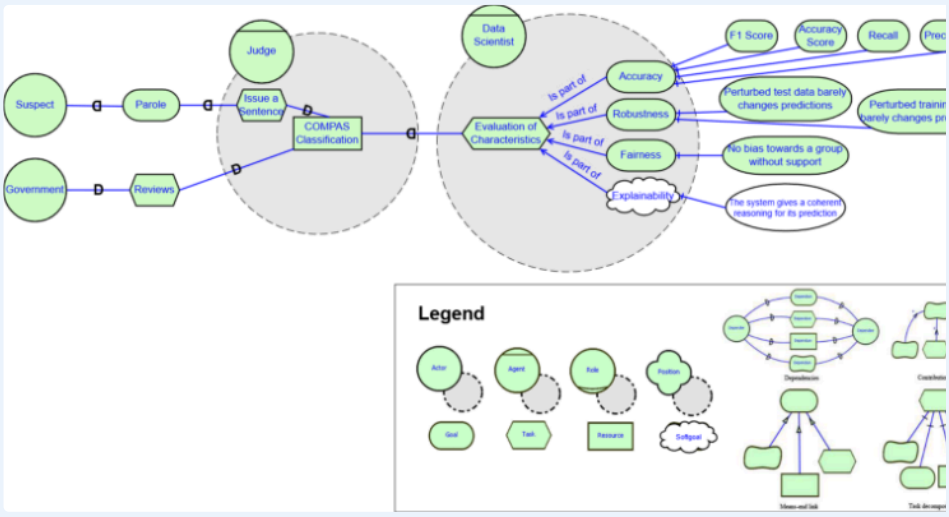
16

What would you change in this visualization?

17

The i* (i-star) framework is a modeling approach that focuses on the dependencies of relationships among various stakeholders within a system. It is useful for capturing and analyzing the needs of stakeholders while providing a simple, yet necessary visualization of the system requirements.
In this visualization, all stakeholders (suspect, judge, data scientist, and government) are directly shown as actors and agents. An agent is an actor who has a concrete physical appearance. The blue lines with a D on them are directional dependency links, which means that one element depends on the effect of another.
In the data scientist circle, the four requirements are shown as part of the evaluation that should be done. Each of them has one or more means-end links from set requirements. These requirements are more text-wise, giving us more flexibility with their interpretations. Like in GR4ML, explainability is shown as a soft goal.

Which stakeholders needs are being met?
(If the image is pixelated you can view it here: https://photos.app.goo.gl/6aifvdt2Z6gxTvui7)



18

Which stakeholders' needs are being met? *

- [ ] Suspect
- [ ] Government
- [ ] Judge
- [ ] Data Scientist

19

How explainable would you rate this recidivism model visualization? *

1	2	3	4	5	6	7	8	9	10

20

What do you like about this visualization?

21

What would you change in this visualization?

Behavior-Driven Development Explanations

In this last section we provided some examples of human-readable Behavior-Driven Development (BDD) requirements for the COMPAS recidivism model. For each requirement we ask you several questions on how understandable you believe it is and as a last task we ask you to write one yourself.

The question everywhere is: **How understandable do you believe this requirement is?**

22

Given that the COMPAS model predicts a high risk of reoffending for a defendant,
When the actual recidivism outcome is later verified,
Then at least 90% of high-risk predictions should be correct (precision),
And the system does not falsely label too many low risk as high-risk (False Positives). *

1	2	3	4	5	6	7	8	9	10

23

Can you explain your answer?

24

Given the COMPAS model has missing data,
When the model generates a risk score for an individual,
Then the system should work as if it has no missing data. *

1	2	3	4	5	6	7	8	9	10

25

Can you explain your answer?

26

Given minor variations in input data (e.g., slight changes in age or number of prior offenses),
When the COMPAS system calculates risk scores,
Then the score should not fluctuate significantly unless the change is meaningful. *

1	2	3	4	5	6	7	8	9	10

27

Can you explain your answer?

28

Given a defendant receiving a high-risk score,
When a judge or lawyer reviews the COMPAS output,
Then the system should provide a clear explanation of the contributing factors (e.g., prior convictions, age). *

1	2	3	4	5	6	7	8	9	10

29

Can you explain your answer?

30

Given the historical data is known to have a bias towards a certain group,
When the data scientist re-trains the model,
Then the data should be normalized towards this group. *

1	2	3	4	5	6	7	8	9	10

31

Can you explain your answer?

32

Given the system provided a risk score with a reasoning,
When the judge reviews the explanation,
Then the model should show a raw probability score (e.g. "risk score: 0.84"). *

1	2	3	4	5	6	7	8	9	10

33

Can you explain your answer?

34

Can you now try to give such a requirement for **Fairness**, using this BDD approach of given-when-then.
Given ...
When ...
Then ... *

Thank you for filling out this survey

35

Is there anything else you would like to say?