Software requirements for AI-based systems

SOEN 691: Engineering AI-based Software Systems

Emad Shihab, Diego Elias Costa
Concordia University

What are software requirements?

- Description of **features** and **functionalities** of the target system
  - E.g., a library system needs to handle user accounts, reserve books, …
- Convey the **expectations** of users from the software product
  - E.g., online access, support hundreds of users, millions of items..

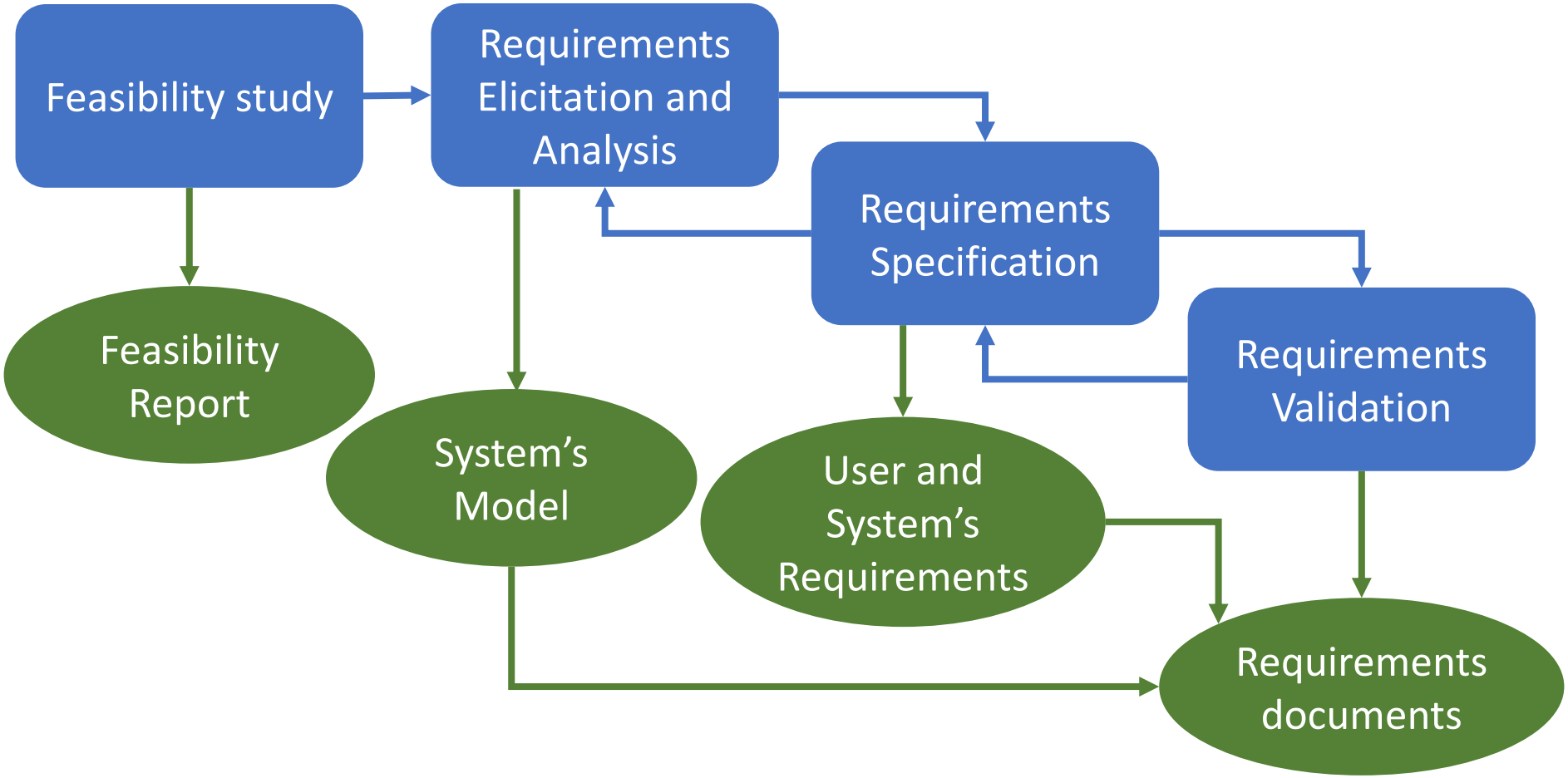
Requirements Engineering Process (RE)

- Requirement Engineering is the process of defining documenting and maintaining the requirements.

- It is a negotiation process involving
  - Developers
  - Users
  - Customers

Kotonya, Gerald; Sommerville, Ian (September 1998). *Requirements Engineering: Processes and Techniques*

The Requirements Engineering Process



What are some of the challenges in RE for AI-based software systems?

Feasibility study

Requirements Elicitation and Analysis

Requirements Specification

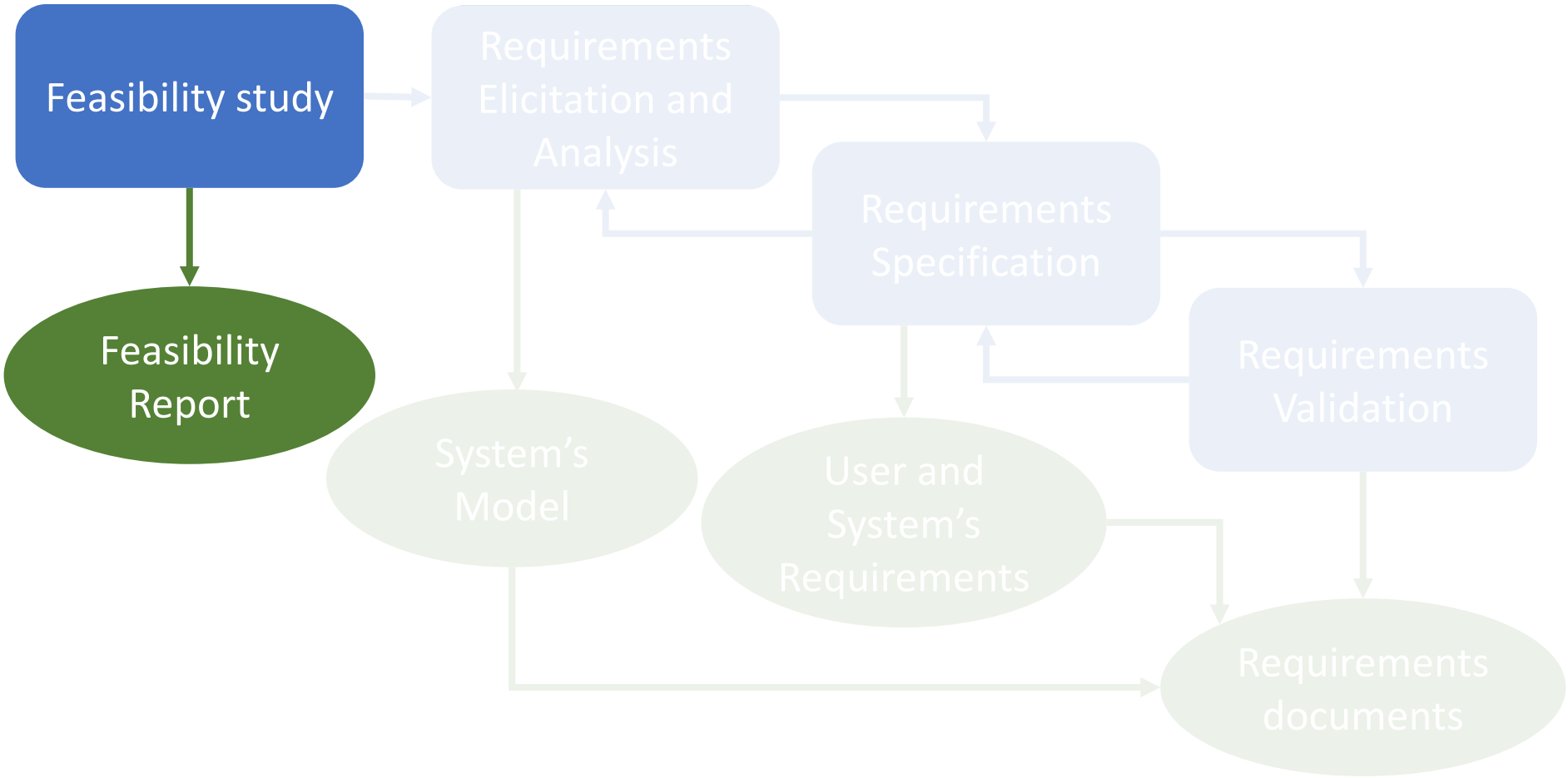
Requirements Validation

The role of RE in AI-based systems

- RE is essential to **understand risks** and **mitigate mistakes**

- Understand
  - Feasibility of the system
  - User interactions
  - Safety requirements
  - Security and privacy requirements
  - Fairness requirements

The Requirements Engineering Process



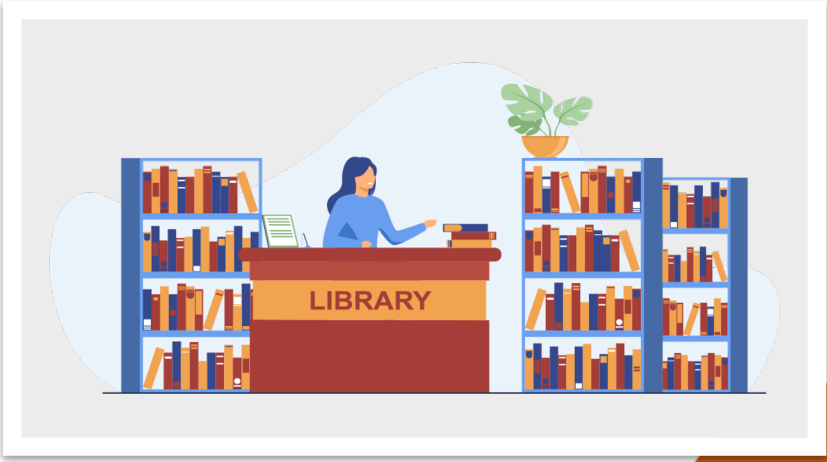
The Feasibility Study

- Analysis of **value** x **cost**
  - Value brought by developing the target system
  - Cost of the target system's development/maintenance

- Developing AI-based systems is expensive!
  - If you have the option to deploy simpler solutions, do it

What Type of Problems Require AI?

Money transferring



Book reservation

Problems that Need AI Solutions

Problems in which solutions require **frequent** updates

- Big problems
- Open-ended problems
- Time-changing problems
- Intrinsically hard problems

Big Problems

- Too many variables/conditions to be completed in one shot

Reason about the content of webpages

Play chess

Open-ended problems

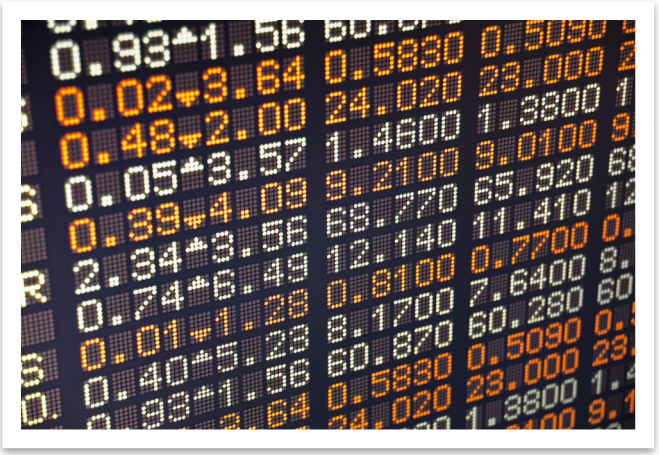
- Some problems do not have a fixed solution

Making driving more efficient

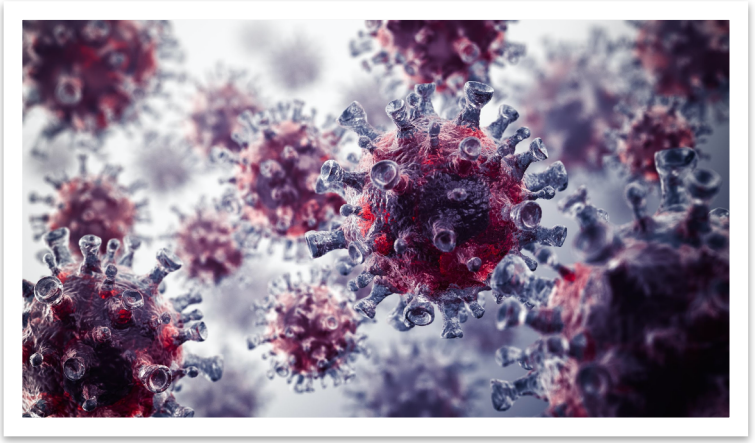
Ecologically sustainable production

Time Changing Problems

- Sometimes the right answer today is wrong tomorrow



System to predict stock prices…

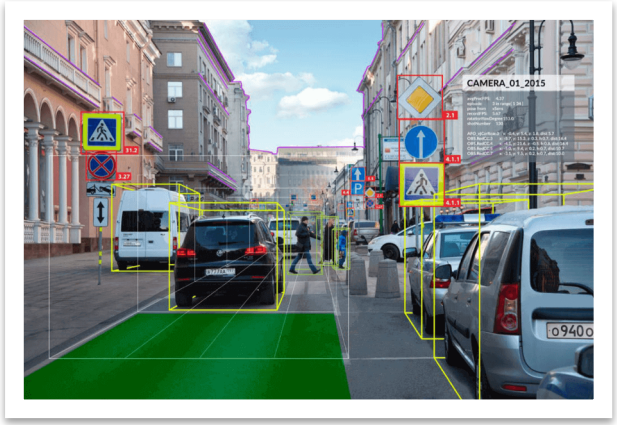


…and then a pandemic happens

Intrinsically Hard Problems

- Some problems are so hard that even humans can't quite figure out how to solve them

Speech recognition



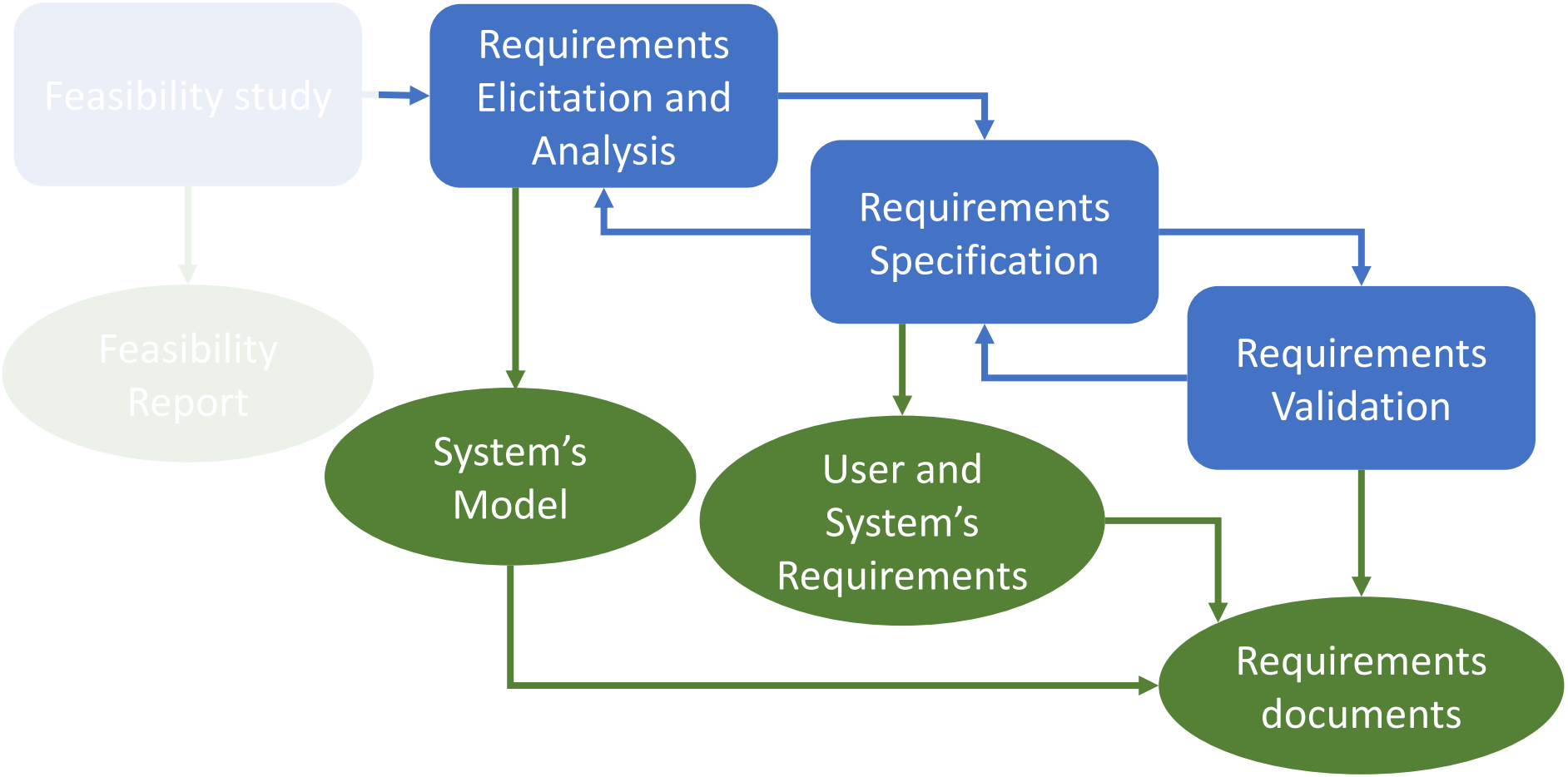
Object recognition

Problems that need AI solutions
(revisited)

Problems in which solutions require **frequent** updates

- Big problems
- Open-ended problems
- Time-changing problems
- Intrinsically hard problems

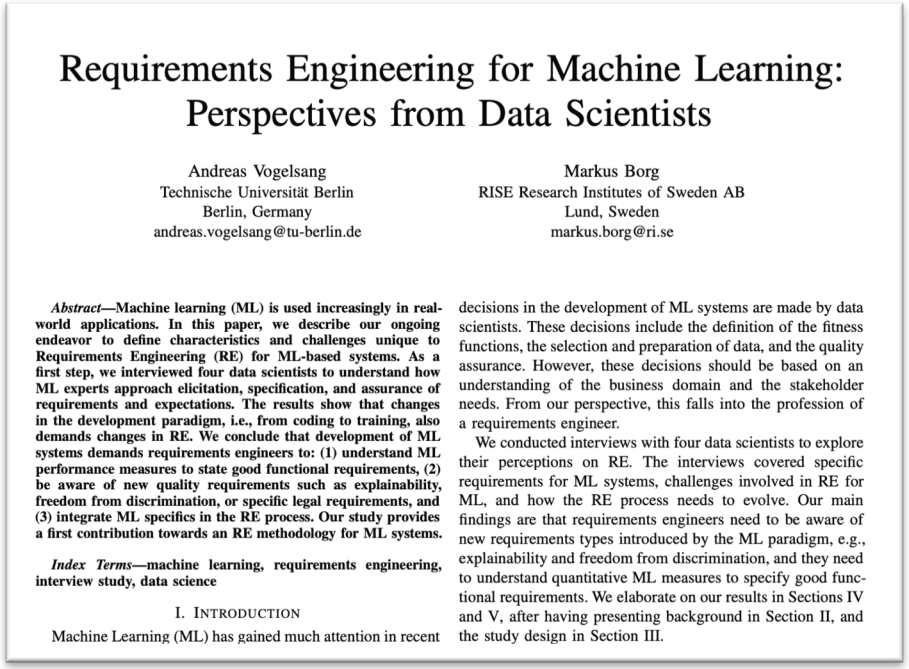
The Requirements Engineering Process



Requirements Engineering for Machine Learning

Focus on two major aspects:

1. Challenges in Requirements for ML Systems
2. RE process for ML Systems



Requirements Engineering for Machine Learning: Perspectives from Data Scientists

Andreas Vogelsang
Technische Universität Berlin
Berlin, Germany
andreas.vogelsang@tu-berlin.de

Markus Borg
RISE Research Institutes of Sweden AB
Lund, Sweden
markus.borg@ri.se

***Abstract*—Machine learning (ML) is used increasingly in real-world applications. In this paper, we describe our ongoing endeavor to define characteristics and challenges unique to Requirements Engineering (RE) for ML-based systems. As a first step, we interviewed four data scientists to understand how ML experts approach elicitation, specification, and assurance of requirements and expectations. The results show that changes in the development paradigm, i.e., from coding to training, also demands changes in RE. We conclude that development of ML systems demands requirements engineers to: (1) understand ML performance measures to state good functional requirements, (2) be aware of new quality requirements such as explainability, freedom from discrimination, or specific legal requirements, and (3) integrate ML specifics in the RE process. Our study provides a first contribution towards an RE methodology for ML systems.**

***Index Terms*—machine learning, requirements engineering, interview study, data science**

I. Introduction

Machine Learning (ML) has gained much attention in recent decisions in the development of ML systems are made by data scientists. These decisions include the definition of the fitness functions, the selection and preparation of data, and the quality assurance. However, these decisions should be based on an understanding of the business domain and the stakeholder needs. From our perspective, this falls into the profession of a requirements engineer.

We conducted interviews with four data scientists to explore their perceptions on RE. The interviews covered specific requirements for ML systems, challenges involved in RE for ML, and how the RE process needs to evolve. Our main findings are that requirements engineers need to be aware of new requirements types introduced by the ML paradigm, e.g., explainability and freedom from discrimination, and they need to understand quantitative ML measures to specify good functional requirements. We elaborate on our results in Sections IV and V, after having presenting background in Section II, and the study design in Section III.

Study Design

- Semi-structured interviews
- Interviewed four data scientists (P1 – P4)
  - P1 and P2 are data scientist researchers
  - P3 and P4 work in industry
- Thematic code analysis of the responses
  - Co-validated across multiple annotators

Challenges in Requirements for ML Systems

The authors enumerate five major challenges in setting up requirements for ML systems

- Setting functional requirements
- Explainability
- Freedom for Discrimination
- Legal and Regulatory Requirements
- Data Requirements

Setting Functional Requirements

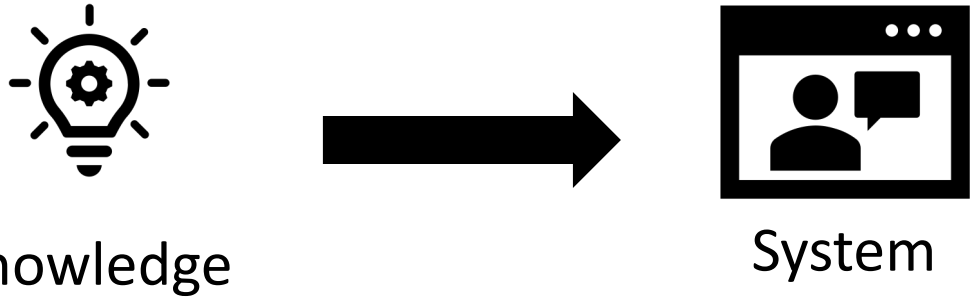
Challenge: How to set the quality expectations for an unknown ML problem?

- Is 40% precision a reasonable target for recommending movies?
- And for credit card fraud detection?

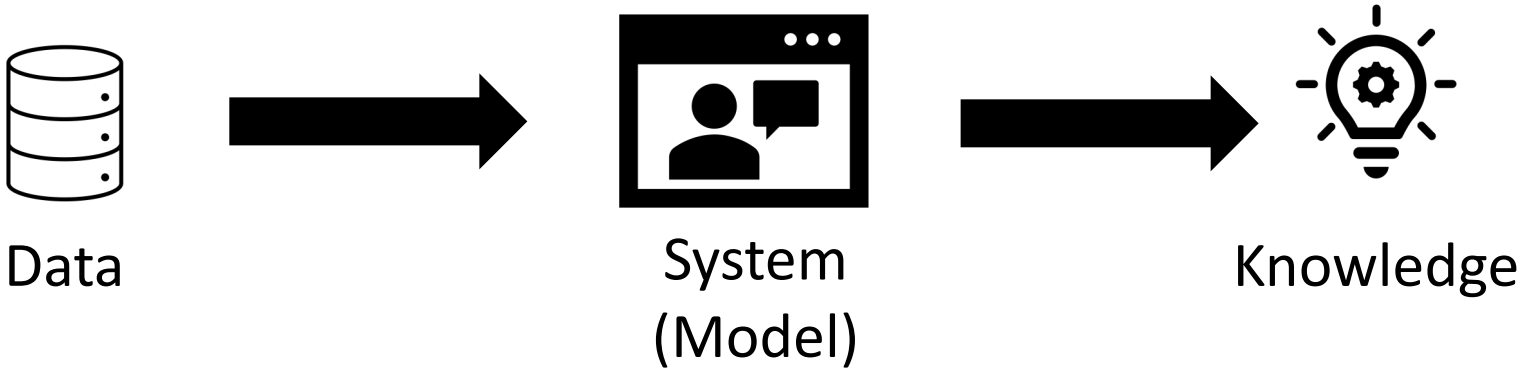
- Engineers need to help clients set reasonable targets
  - Domain understanding
  - Statistics
  - Computer Science

Explainability

Traditional Systems



AI/ML Systems



Explainability

Challenge: Comprehend ML systems

1. Need to explain the model
2. Need to explain single predictions

Explainability may be even **more important** than predictive power

- Simpler models are easier to explain
- Less features are easier to explain

Explainability (cont'd)

Challenge: Comprehend ML systems

1. Need to explain the model
2. Need to explain single predictions

Engineers need to **explicitly list** explainability requirements from a user's point of view

- What predictions require explanation?
- How should the system explain to users?

Freedom from Discrimination

- AI systems are designed to discriminate
  - Identify recurring **patterns**
  - Apply those patterns to **judge** about unseen data

- Some types of discrimination are **unacceptable** by our society and law
  - Insurance systems or job recruiting systems
    - Gender
    - Race

Freedom from Discrimination (cont'd)

Challenge: Ensure that the ML system does not discriminate using **protected characteristics**

- Discrimination is more implicit in AI-systems
- AI algorithms amplify discrimination biases

- Engineers must elicit and identify the **protected characteristics** that must not be used by the ML system
  - Before collecting data
  - After analyzing the important features (more efficient)

Legal and Regulatory Requirements

Challenge: Ensure that the ML system is collecting and using data under legal terms

- GDPR states that personal data can only be used in ways specified by **explicit consent**

General **D**ata **P**rotection **R**egulation

Legal and Regulatory Requirements (cont'd)

- GDPR vs ML systems
  - All features (data) require explicit consent
  - But not all features are useful to the end model
  - Collecting and not using the data as initially described is illegal

- Engineers working with ML systems need to be **on top of legal requirements**
  - Data lineage should show that no illegal features are used

Data Requirements

- The behavior of traditional system is given by the **code**
- The behavior of an ML system is given by the **data** + model

Challenge: Training data needs specified/validated requirements

- Requirements for data quantity
- Requirements for data quality

Data Quantity Requirements

- More data is not always better
  - E.g., more data about authentic credit card transactions will not help you better detect fraudulent ones.

- Diversity of data is key
  - Some domains regulate the minimum size of training data
    - E.g., likelihood of loan losses require at least 5 years of data

Data Quality Requirements

- Garbage-in -> Garbage-out

- Quality attributes
  - Completeness (does it cover the range of values?)
  - Consistency (is data consistently represented?)
  - Correctness (can we trust the data?)

- **Red flags**
  - Public datasets -> rarely well maintained
  - Human classification -> biases

Data Requirements (cont'd)

- In practice, we cannot always control data-quality aspects
  - Should one risk build a system on top of unreliable data?
- Engineers need to identify and specify requirements regarding
  - Data sources
  - Data collection and format
  - Ranges of data

Example: Credit Report

Let us discuss these aspects in an ideal credit report ML system:

- Setting functional requirements
- Explainability
- Freedom for Discrimination
- Legal and Regulatory Requirements
- Data Requirements

Impact of ML on RE Activities

Requirements Elicitation and Analysis

Requirements Specification

Requirements Validation

Requirements Elicitation and Analysis

Elicitation

- Identify all possible relevant sources of data
- Are there domain-related protected characteristics?

Analysis

- Establish performance measures
- Define data collection process

Requirements Specification

- Set quantitative targets
- Set data requirements
  - Quantity and quality
- Define Explainability from the user's point of view
- Set policies to ensure freedom from discrimination
- Legal and regulatory constraints

Requirements Validation

- Requirements may also change with time
  - E.g., government subsidies may reduce the risk of credit request with purpose of education

- Analyze operational data
- Retrain ML models
- Detect data anomality

Impact of ML on RE Activities

Requirements Elicitation and Analysis

- Elicit data sources
- List protected characteristics
- Discuss performance measures
- Discuss data collection process

Requirements Specification

- Quantitative targets
- Data requirements
- Explainability
- Freedom from discrimination
- Legal constraints

Requirements Validation

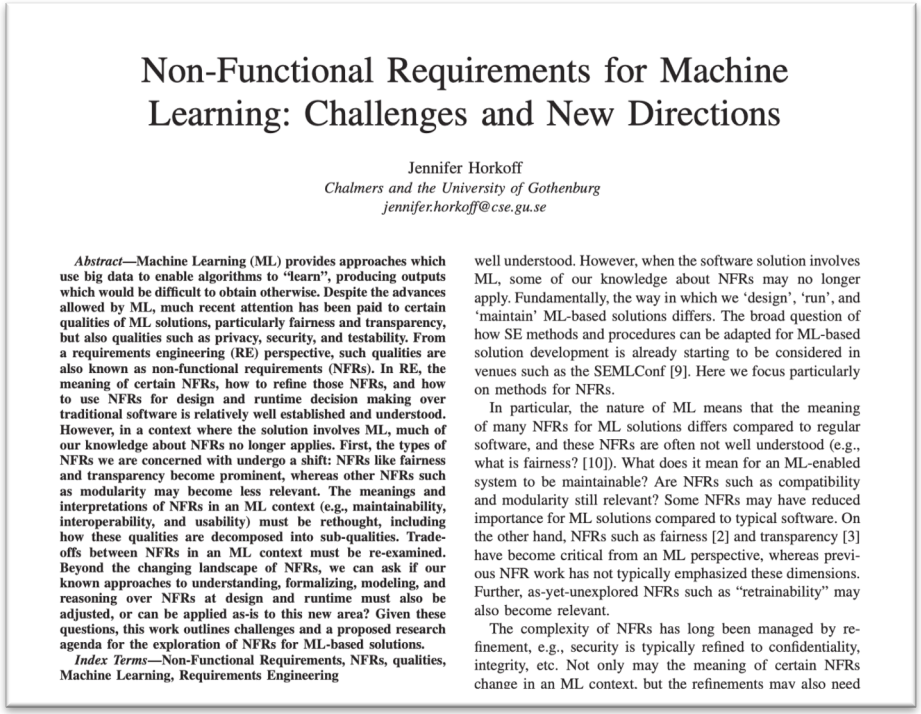
- Analyze operational data
- Look for bias in data
- Retrain ML models
- Detect data anomalies

Open Discussion

Non-Functional Requirements for Machine Learning

Focus on:

1. Listing the challenges to conceive and measure NFR
2. Provide some avenues for research



Non-Functional Requirements for Machine Learning: Challenges and New Directions

Jennifer Horkoff
Chalmers and the University of Gothenburg
jennifer.horkoff@cse.gu.se

***Abstract*—Machine Learning (ML) provides approaches which use big data to enable algorithms to "learn", producing outputs which would be difficult to obtain otherwise. Despite the advances allowed by ML, much recent attention has been paid to certain qualities of ML solutions, particularly fairness and transparency, but also qualities such as privacy, security, and testability. From a requirements engineering (RE) perspective, such qualities are also known as non-functional requirements (NFRs). In RE, the meaning of certain NFRs, how to refine those NFRs, and how to use NFRs for design and runtime decision making over traditional software is relatively well established and understood. However, in a context where the solution involves ML, much of our knowledge about NFRs no longer applies. First, the types of NFRs we are concerned with undergo a shift: NFRs like fairness and transparency become prominent, whereas other NFRs such as modularity may become less relevant. The meanings and interpretations of NFRs in an ML context (e.g., maintainability, interoperability, and usability) must be rethought, including how these qualities are decomposed into sub-qualities. Trade-offs between NFRs in an ML context must be re-examined. Beyond the changing landscape of NFRs, we can ask if our known approaches to understanding, formalizing, modeling, and reasoning over NFRs at design and runtime must also be adjusted, or can be applied as-is to this new area? Given these questions, this work outlines challenges and a proposed research agenda for the exploration of NFRs for ML-based solutions.**

***Index Terms*—Non-Functional Requirements, NFRs, qualities, Machine Learning, Requirements Engineering**

well understood. However, when the software solution involves ML, some of our knowledge about NFRs may no longer apply. Fundamentally, the way in which we 'design', 'run', and 'maintain' ML-based solutions differs. The broad question of how SE methods and procedures can be adapted for ML-based solution development is already starting to be considered in venues such as the SEMLConf [9]. Here we focus particularly on methods for NFRs.

In particular, the nature of ML means that the meaning of many NFRs for ML solutions differs compared to regular software, and these NFRs are often not well understood (e.g., what is fairness? [10]). What does it mean for an ML-enabled system to be maintainable? Are NFRs such as compatibility and modularity still relevant? Some NFRs may have reduced importance for ML solutions compared to typical software. On the other hand, NFRs such as fairness [2] and transparency [3] have become critical from an ML perspective, whereas previous NFR work has not typically emphasized these dimensions. Further, as-yet-unexplored NFRs such as "retrainability" may also become relevant.

The complexity of NFRs has long been managed by refinement, e.g., security is typically refined to confidentiality, integrity, etc. Not only may the meaning of certain NFRs change in an ML context, but the refinements may also need

Non-Functional Requirements (NFR)

- Quality or attribute which is non-functional

- Traditional Software
  - Maintainability
  - Performance
  - Compatibility
  - Security
  - […]

- What about NFR in Machine Learning systems?

Motivation

It is hard to **conceptualize** traditional NFR in the context of Machine Learning systems.

For example:

- What does it mean for an ML system to be maintainable?
- Are NFR such as modularity and compatibility still relevant?
- What is fairness in the context of an ML system?

Motivation (cont'd)

Some NFRs are emphasized more in Machine Learning system than in traditional systems:

- **Fairness** and **transparency** are essential for high-stake ML systems
  - They are less emphasized in traditional systems

Trade-offs are not so well known:

- Security x performance is a common trade-off in traditional systems
- Do similar trade-offs exist in ML systems?

Core Problem

Current understanding of NFR and methods are not directly applicable to NFR in ML based system!

Knowledge

Rethink the concepts of NFR for ML systems

Measurement Methods

Revamp techniques for this new paradigm

Study Design

- Argumentative (persuasive) research paper
- Discusses 7 challenges of the treatment of NFR for ML
  - 4 Challenges related to the knowledge
  - 3 Challenges related to measurement methods
- Outline 6 research objectives and plans

Related work

- NFR discussed in the context of ML systems
  - Accuracy and Performance
  - Fairness
  - Transparency
  - Security and Privacy
  - Testability
  - Reliability

Some NFR were overlooked by the community

- Maintainability, sustainability, modularity …

C1: How to define NFR in ML contexts

Take fairness as an example:

- What is fairness?
- How can fairness be required from a ML model?
- Does fairness change depending on
  - The domain of the problem
  - The type of model algorithm used

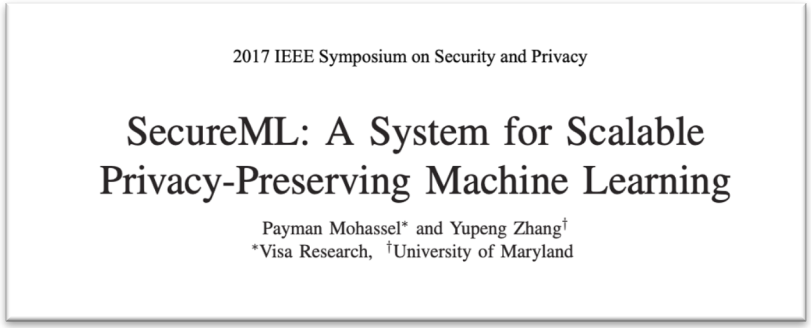
The same problem is applicable to other NFRs

- Portability, Usability…

C2: We need a better understanding of NFR trade-offs

Some works have tried to tackled some trade-offs:

- Privacy preserving vs Algorithm speed

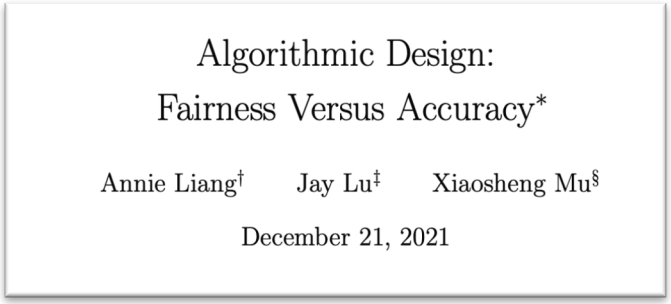


2017 IEEE Symposium on Security and Privacy

SecureML: A System for Scalable Privacy-Preserving Machine Learning

Payman Mohassel* and Yupeng Zhang†
*Visa Research, †University of Maryland

- Fairness vs algorithm accuracy



Algorithmic Design: Fairness Versus Accuracy*

Annie Liang† Jay Lu‡ Xiaosheng Mu§

December 21, 2021

The Measure and Mismeasure of Fairness: A Critical Review of Fair Machine Learning*

Sam Corbett-Davies, Stanford University
Sharad Goel, Stanford University

August 14, 2018

C3: How to measure NFRs?

- We know how to measure performance
  - Accuracy, F1 score, ROC AUC
- How to measure
  - Fairness?
  - Modifiability?
  - Transparency?
  - Security?

C4: Effect of ML algorithms on NFR

We should choose the algorithm based on their inherent NFR qualities

- Interpretability
  - Linear models (e.g., logistic regression)
  - Tree-based solutions (e.g., random forest)

- But what about retrainability? Fairness?
  - Are there models that are inherently better in some NFRs than others?

C5: Express and specify NFR in the models

- We need to express the NFR in terms of ML processes
  - Data curation
  - Data training and retraining
  - Optimizations
  - Choice of the ML algorithm (C4)
  - …
- Which process do I follow to achieve certain NFR?

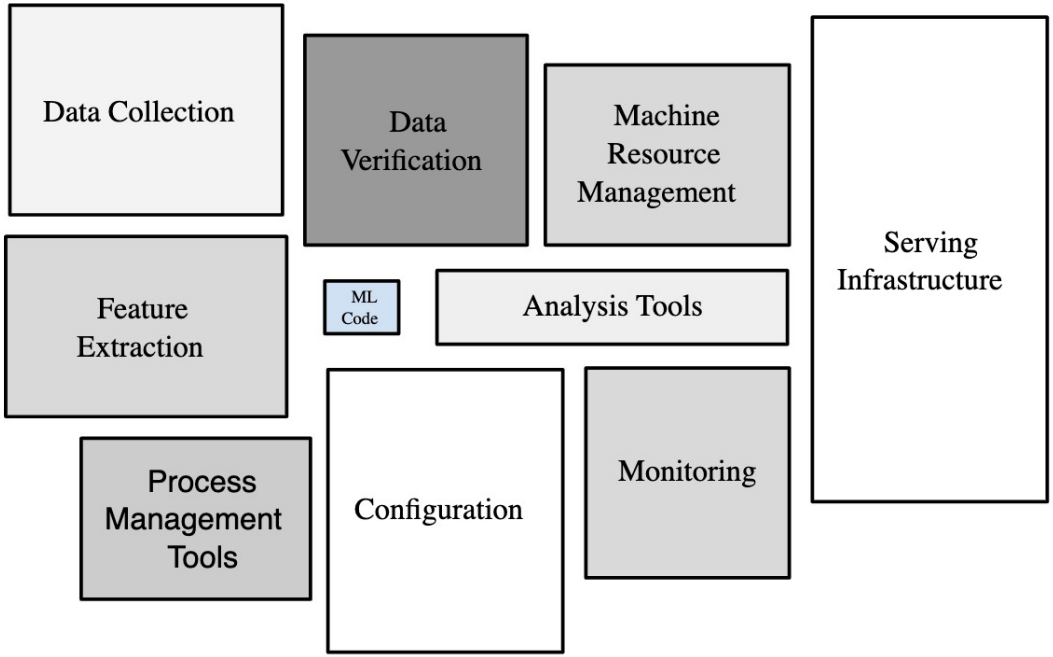
C6: How evolution affect the NFR of our ML solutions?

Evolution in ML system comes from many sources:

- Data changes
  - E.g., pandemic has changed global economy
- Requirements change
  - E.g., songs need to be shorter to become a hit in 2022
- Actors reaction (feedback loops)
  - E.g., trading in stock market must react to other buyers/sellers

C7: How ML solutions integrate with systems

NFR must consider the ML component within the system.



https://developers.google.com/machine-learning/crash-course/production-ml-systems

Research Directions

1. Explore and define NFR for ML
   - Survey ML literature
   - Ask ML experts about NFR
   - Consider existing NFR refinements
2. Create a catalogue of NFR for ML
3. Collect measures of NFR
   - Contextual measures

Research Directions (cont'd)

4. Define methods to express NFRs for ML solutions
   - Conceptual underpinning
   - Graphical and textual syntax

5. Define methods to reason over NFRs for ML solutions
   - Runtime monitoring methods

6. Create methods to deal with NFR in evolving ML
   - Methods for changing data and requirements

Open Discussion

Homework

- Two papers are posted on Moodle
- For one of the papers, write a summary (aprox. 1/3 of a page)
- For the other paper, write a critique, which includes a summary, at least 3 strong points and at least 3 weaknesses (aprox. 1 page).
- Submit your summary and critique on Moodle by Friday, Feb. 4th at 5pm