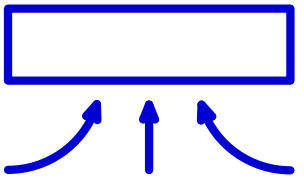
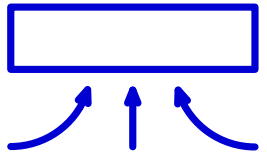
Requirements Engineering I

Chapter 11

RE Support and Guidance



Chapter roadmap



11.1 RE tools

The little helpers

11.2 RE Standards

Are they known, used and useful?

11.3 RE syllabi and body of knowledge

Standardizing RE knowledge and skills

11.4 AI for RE

Harnessing AI for RE

11.5 Requirements Engineering Ethics

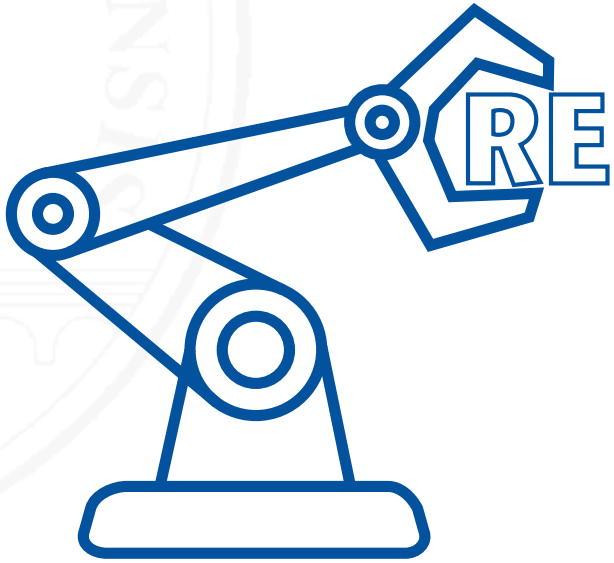
Ethic principles and dimensions mapped to RE

11.1 Requirements engineering tools

[Carrillo de Gea et al. 2011]

What can be supported by a RE tool?

- Elicitation (e.g., analysis of textual artifacts)
- Documentation (generating and editing requirements work products)
- Modeling (primarily model editors)
- Management (Store and retrieve, prioritize, trace,...)
- Validation (finding quality problems, simulators, model checkers,...)



Support levels for RE tools

- Editors
  - Text editors, spreadsheet tools, graphic drawing tools
- Database-level
  - Requirements management tools for organizing, storing, retrieving and tracing requirements
- Language & method-based
  - Tools supporting specific requirements languages, e.g., drawing state machine diagrams, or specific methods, e.g., validation with model-checking
- AI-powered
  - See Chapter 11.4

Which RE tool should I use / buy?

[Bruckhaus, Madhavji, Janssen, Henshaw 1996]

- No general recommendation possible
- Depends on what the tool(s) shall support
- An RE tool does not automatically improve productivity
- An up-to-date list of requirements tools is maintained at the VOLERE website:

  https://www.volere.org/requirements-tools/

11.2 RE Standards

IEEE 830-1984 IEEE Guide to Software Requirements Specifications

- The first RE standard – very good by its time
- Revised 1993 and 1998
- IEEE 830-1998 is officially retired, but still in use, in particular for documenting requirements

ISO/IEC/IEEE 29148, originally from 2011, revised 2018

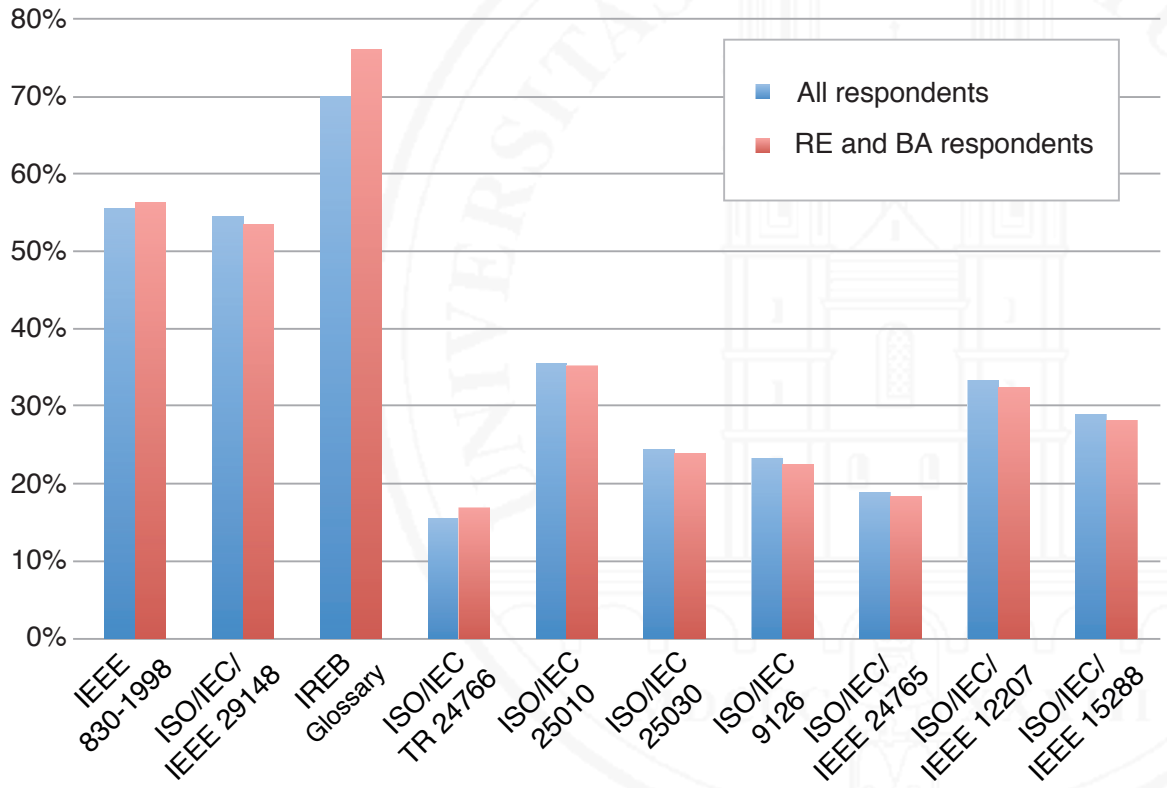
- A very heavyweight, document- and process-centric standard
- Does not work well for participative and lightweight RE processes

[IEEE 1998]
[ISO/IEC/IEEE 2018]

Knowledge and use of RE-related standards

[Franch, Glinz, Méndez and Seyff 2022]

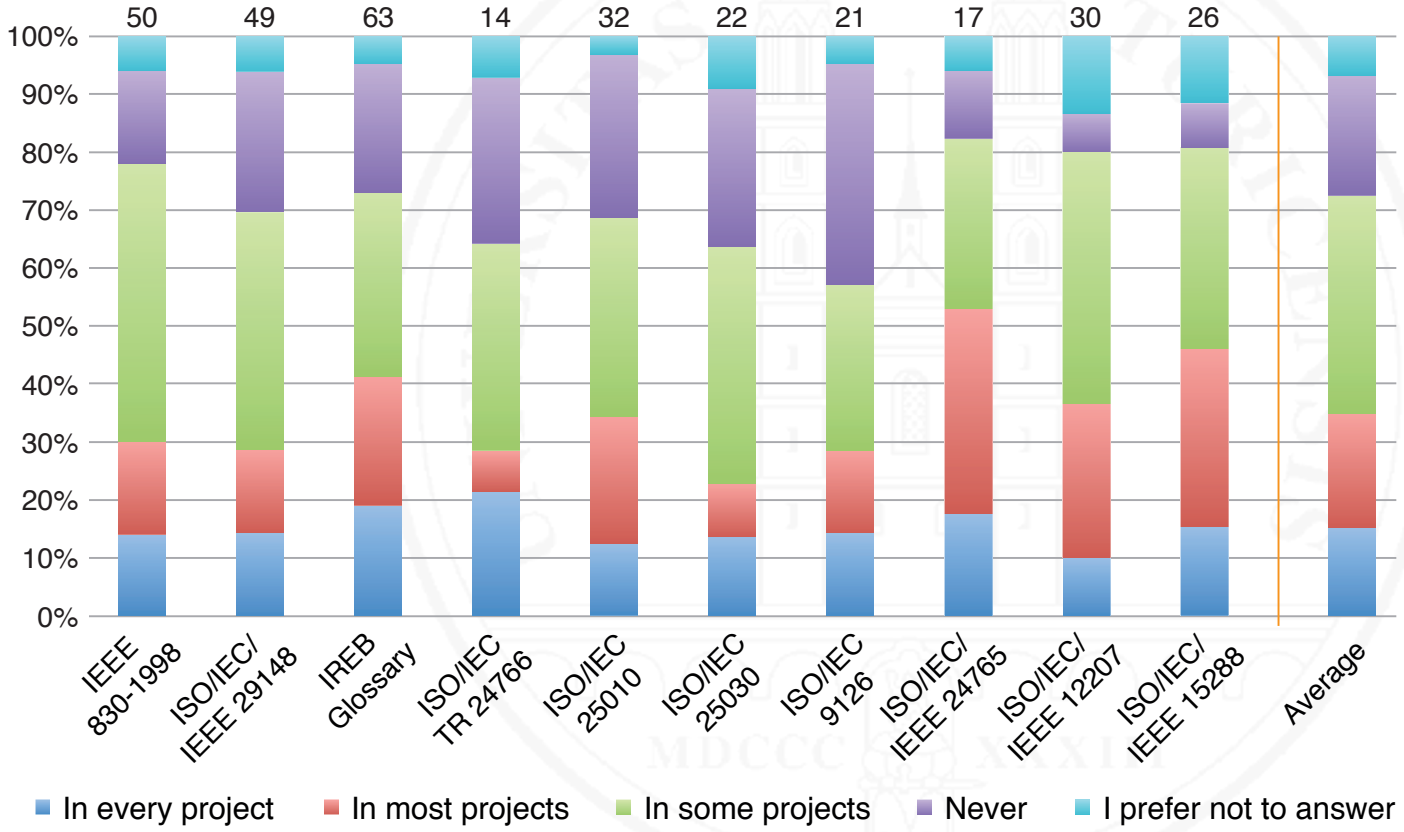
Results from an empirical study:



→ The knowledge of RE-related standards is rather low

Knowledge and use of RE-related standards – 2

Frequency of usage



→ The known standards are barely used

Related standards

Quality standards, particularly in conjunction with quality requirements

- ISO/IEC 25010 System and Software Quality Requirements and Evaluation: Quality Models
- ISO/IEC 25030 Software Product Quality Requirements and Evaluation: Quality Requirements
- ISO/IEC 9126 Software Engineering – Product Quality: Quality Model (superseded, predecessor of ISO/IEC 25010)

System and software engineering standards, e.g.,

- ISO/IEC/IEEE 12207 on software life cycle processes
- ISO/IEC/IEEE 15288 on system life cycle processes
- ISO/IEC/IEEE 24765 on systems & software engineering vocabulary

Domain-specific standards

Domain-specific standards may impact Requirements Engineering

Example:

ISO 26262 Road Vehicles — Functional Safety

If a customer or regulator demands compliance of a system with ISO 26262, then traceability between requirements and test cases is mandatory.

11.3 RE syllabi and body of knowledge

There is no Requirements Engineering Body of Knowledge (RE BoK) document

The IREB CPRE – Certified Professional for Requirements Engineering – foundation level

is a de facto basic RE BoK, consisting of a syllabus, a handbook and a glossary

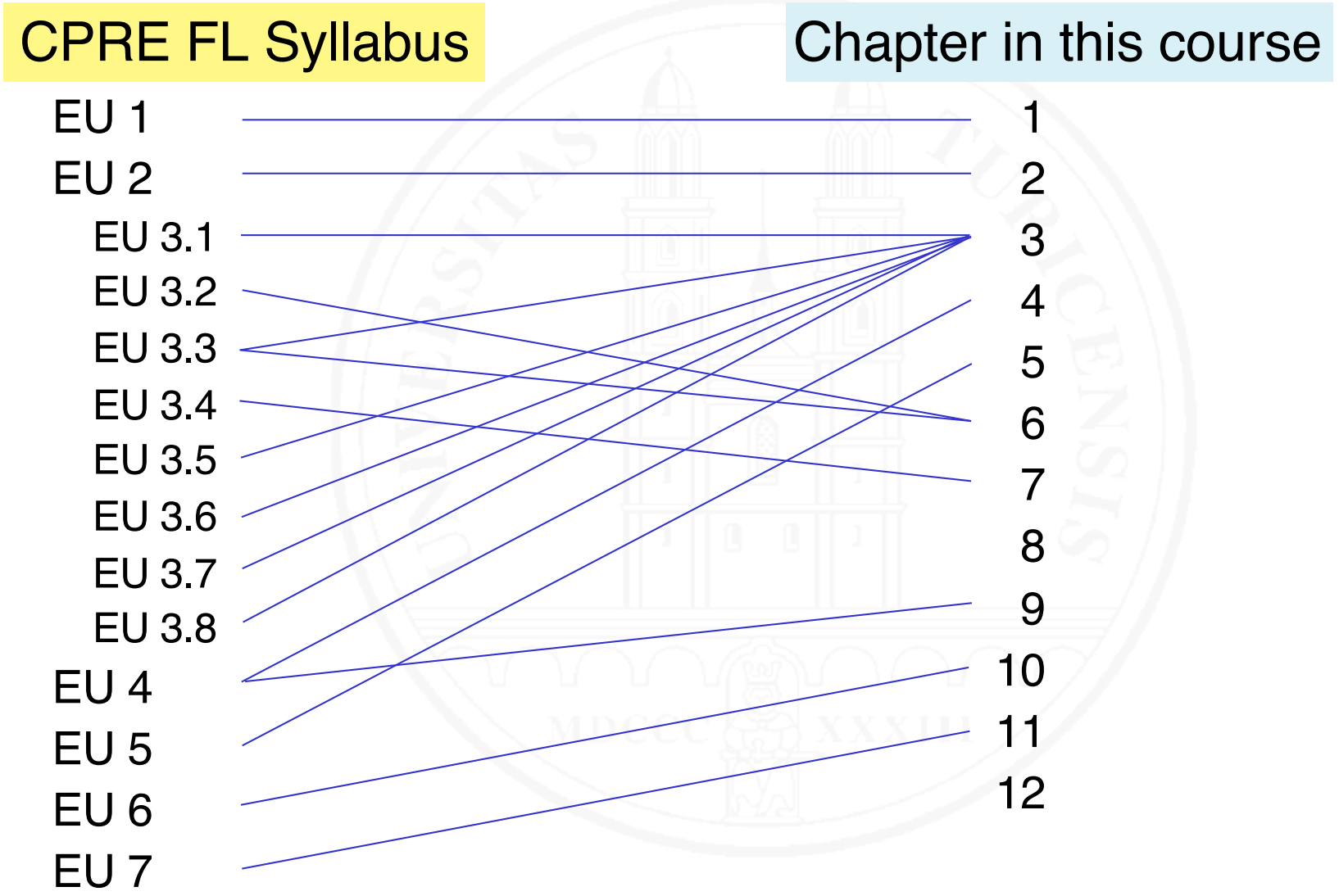
https://ireb.org/en

[IREB 2024]
[Glinz, van Loenhoud, Staal and Bühne 2024]
[Glinz 2024]

IREB CPRE vs. this course

- This course covers all topics of the IREB CPRE foundation level syllabus
- The terminology is the same
- Some topics of this course go beyond the CPRE FL, for example:
  - COTS-aware processes (Chapter 5)
  - Formal specification (Chapter 8)
  - Standards, AI for RE, RE Ethics (Chapter 11)

Synopsis of topics



11.4 AI for RE

What can AI do for RE?

[Dalpiaz and Niu 2020]
[Vogelsang 2024]
[Vogelsang and Fischbach 2025]

Primary means: Processing natural language text with machine learning (ML) and Large Language Model (LLM) technology

Main use cases in RE today (cf. Chapter 2.6):

- Understanding (i.e., finding, classifying, analyzing,...) tasks: AI-powered tools for use by humans
- Generation tasks: AI-powered agents act as assistants to a human requirements engineer

Understanding tasks

- Find and classify, for example
  - Identify potential requirements in large volumes of user feedback (app reviews, tweets)
  - Classify sentences in a document into requirements and informational statements
  - Extract glossary candidates from textual requirements
  - Find smells in requirements
- Analyze, for example
  - Automated impact analysis when requirements change

Understanding tasks – 2

- Recommend, for example
  - Recommend further stakeholders / stakeholder roles during stakeholder analysis
  - Provide advice for configuring requirements in a product line

Generation tasks

- Extract or summarize, for example
  - Extract requirements from interview transcripts
  - Generate epics from sets of user stories
- Complete, for example
  - Add acceptance criteria to a given user story
- Link, for example
  - Create trace links, e.g., from requirements to test cases
- Assist, for example
  - Perform a stakeholder analysis
  - Maintain a requirements engineer's to-do-list

Generation tasks – 2

- Transform, for example
  - Create process models from textual process descriptions
  - Generate a prototype from a set of requirements
  - Translate requirements into the official language of a project

Example 1: Stakeholder Analysis with GenAI

Context: Requirements Elicitation

Problem: Identify stakeholder roles and people in these roles

Task type: Generation task

Solution: Prompt a generative, pretrained AI system with

- Information about stakeholder roles and stakeholder classification
- Available information about the project
- Instructions how to present the results
- Maybe some examples

Example 2: Feedback analysis with ALERT.me

[Guzmán, Ibrahim, Glinz 2017]

Context: Large product or service providers continuously receive thousands of tweets about their product.

Problem: Some of these tweets contain user needs that are a source of requirements for evolving the product or service. Manually finding these tweets is tedious and expensive.

Illustration: Two tweets to Slack:

@SlackHQ At my company we share code snippets around a lot. There should be a quick way to copy a raw code snippet to your clipboard. User need

I always uwanted t-shirts, but I didn't know socks were an option. I've got the start with my @SlackHQ faves - gotta catch 'em all! Other stuff

Task type: Understanding task

Feedback analysis with ALERT.me – 2

Solution: Build an ML-based tool that extracts user needs and presents them in a convenient form to the requirements engineers

Three steps:

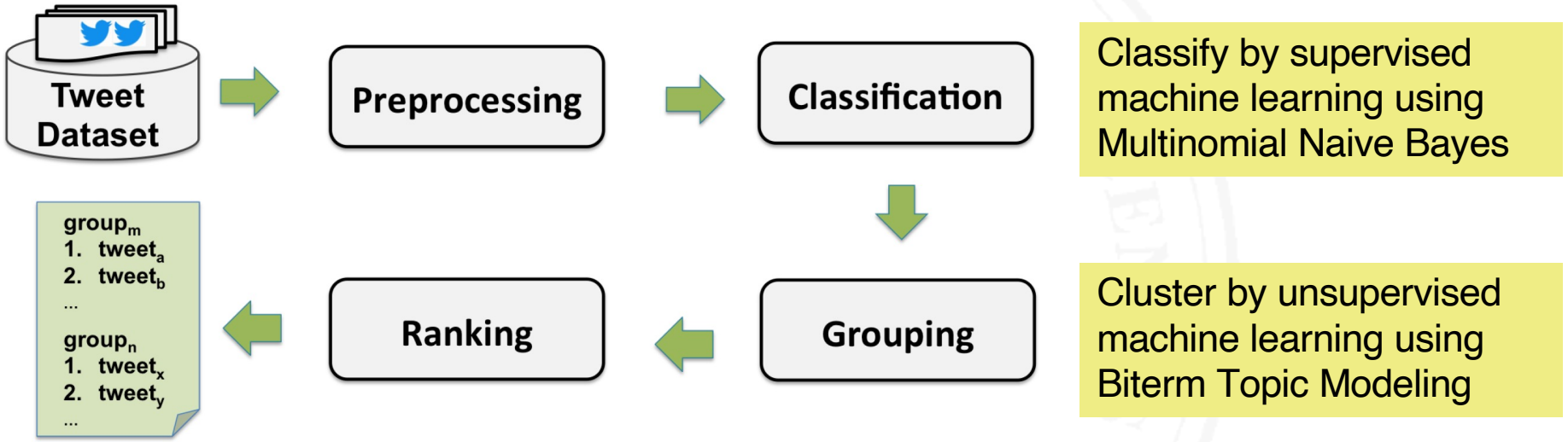
1 Classify tweets into improvement requests and other

2 Cluster improvement requests by grouping them into topics

3 Rank the grouped requests by their relevance

Feedback analysis with ALERT.me – 3

Architecture of ALERT.me



Rank with a weighting function considering factors such as likes, retweets or sentiment, using empirically determined weights (worked better than machine learning the weights)

Note: When we developed ALERT.me, GenAI systems did not yet exist. Today, one would probably solve the given problem by taking a GenAI system, fine-tune it for the given context and then prompt it to perform the three steps of classifying, clustering, and ranking.

The recall problem of AI-based RE tools

[Berry 2021]

- Automated classifiers make mistakes:
  - Not including relevant items in the result set (false negatives; recall < 100 %)
  - Including irrelevant items (false positives; precision < 100 %)
- A tool such as ALERT.me is still useful when recall is only about 80 %.
- In other contexts, a tool with 80% recall can be useless because the missed items have to be found manually

Mini-Exercise

Explain why an AI-based tool with a recall of 80% can be very useful in certain RE contexts and useless in other RE contexts.

Analytic vs. generative AI

[Vogelsang 2024]

Analytic AI tools for understanding tasks (such as ALERT.me)

- Configured for specific tasks
- Need to be trained
- Rather easy to handle and interpret once training is done

Generative AI (such as ChatGPT)

- General-purpose large language models
- Specific prompts required for RE problems: context, task, expected result(s), qualities, constraints,...
- Interpretation of results requires humans in the loop
- May need additional training on context and RE problems
- Proper prompting can become an RE task in itself

Outlook

In the future, we will probably see autonomous AI-powered agents that

- interact with stakeholders without guidance by a requirements engineer
- analyze the information received from stakeholders
- generate RE work products or even deployable solutions

Opportunities: for example, producing agreed requirements from interactions with thousands of stakeholders

Challenges: How to validate? How to avoid bias and discrimination? How to detect fakes and hallucinations?

11.5 Requirements Engineering Ethics

Three relevant ethical dimensions

- Ethics of profession
- Ethics of use
- Ethics of design

in RE

[Simon 2022]
[Barker&Ferguson 2022]
[Norman 2013]

Ethic principles

- No harm 
- For the good 
- Fairness 
- Autonomy 
- Transparency 

Ethics of profession in RE

- Consider how to act ethically as a requirements engineer
- Comply with the code of ethics
  - of your organization
  - of professional societies where you are a member

Some advice

- Refuse working on maleficent requirements (no harm)
- Assess benefits and risks of systems built according to the requirements (for the good)
- Treat equal stakeholders equal (fairness)
- Guide stakeholders, but do not force them (autonomy)
- Be able to explain what you are doing and why (transparency)

Ethics of use in RE

- Consider the impact of your requirements on the users of the system to be built
  - Can the system in use do harm (to people, the environment, the society,...)?
  - Does the system help its users doing things better than before?
  - Does or can the system discriminate certain users or or favor them over others without a valid reason?
  - Does the system help empower its users?
  - Does the system help users understand what the system does when they use it?

Ethics of design in RE

- Consider the impact of your requirements on the design of the system to be built
  - Are there requirements that prevent the system from doing harm (safety, security, reliability,...)
  - Do the requirements enable designing a system that provides benefit for, for example, its users, the environment or the society – and do this with controllable risks?
  - Do the requirements enable designing a system that is user-friendly and empowers its users?
  - Are there requirements asking for explainability of what the system does when in operation?