CHALMERS
UNIVERSITY OF TECHNOLOGY

UNIVERSITY OF
GOTHENBURG

Requirements Engineering for Agentic AI Systems

Swedsoft
Expert Forum: Agentic AI

Jennifer Horkoff
Assoc. Prof./Docent

jenho@chalmers.se jennifer.horkoff@gu.se

UNIVERSITY OF GOTHENBURG
CHALMERS UNIVERSITY OF TECHNOLOGY
AI Engineering Lab

About Me

- Researcher in requirements engineering
- Now also working in AI Engineering
  - Co-founded/run Chalmers and GU AI Engineering Lab
  - Co-PC Chair of CAIN'26 (5th International Conference on AI Engineering – Software Engineering for AI), general chair of CAIN'27
  - Co-area chair of SE for AI for ICSE 2027



Software Engineering in the Rise of AI

- Most people think of AI for SE
  - Code generation, e.g., Claude Code, ChatGPT Codex
  - Using an LLM as a helper (or doer) in Software Development
- But there is also SE for AI
  - Development and engineering of systems with AI components, or built around AI
- Here, I will talk a bit about both, but more so SE for AI
  - Specifically, RE for AI and Agentic systems
- Even if your current focus is on agentic AI for code generation, I believe over time that “the system” will less about the code generated and more about the agentic system generating (and testing) the code
  - or that “the system” will contain agents
  - **Requirements for agentic AI systems is/will be important**

CHALMERS UNIVERSITY OF TECHNOLOGY | UNIVERSITY OF GOTHENBURG

Overview

- (briefly) RE
- (briefly) AI for RE
- AI Systems
- RE for AI Systems
- Agentic Systems
- RE for Agentic Systems

CHALMERS UNIVERSITY OF TECHNOLOGY | UNIVERSITY OF GOTHENBURG

Requirements Engineering (RE)

CHALMERS UNIVERSITY OF TECHNOLOGY | UNIVERSITY OF GOTHENBURG

Requirements Engineering – More than just a Spec

Agile RE

Human and behavioral issues

Modeling

Quality

Legal and regulations

Representations and Analysis

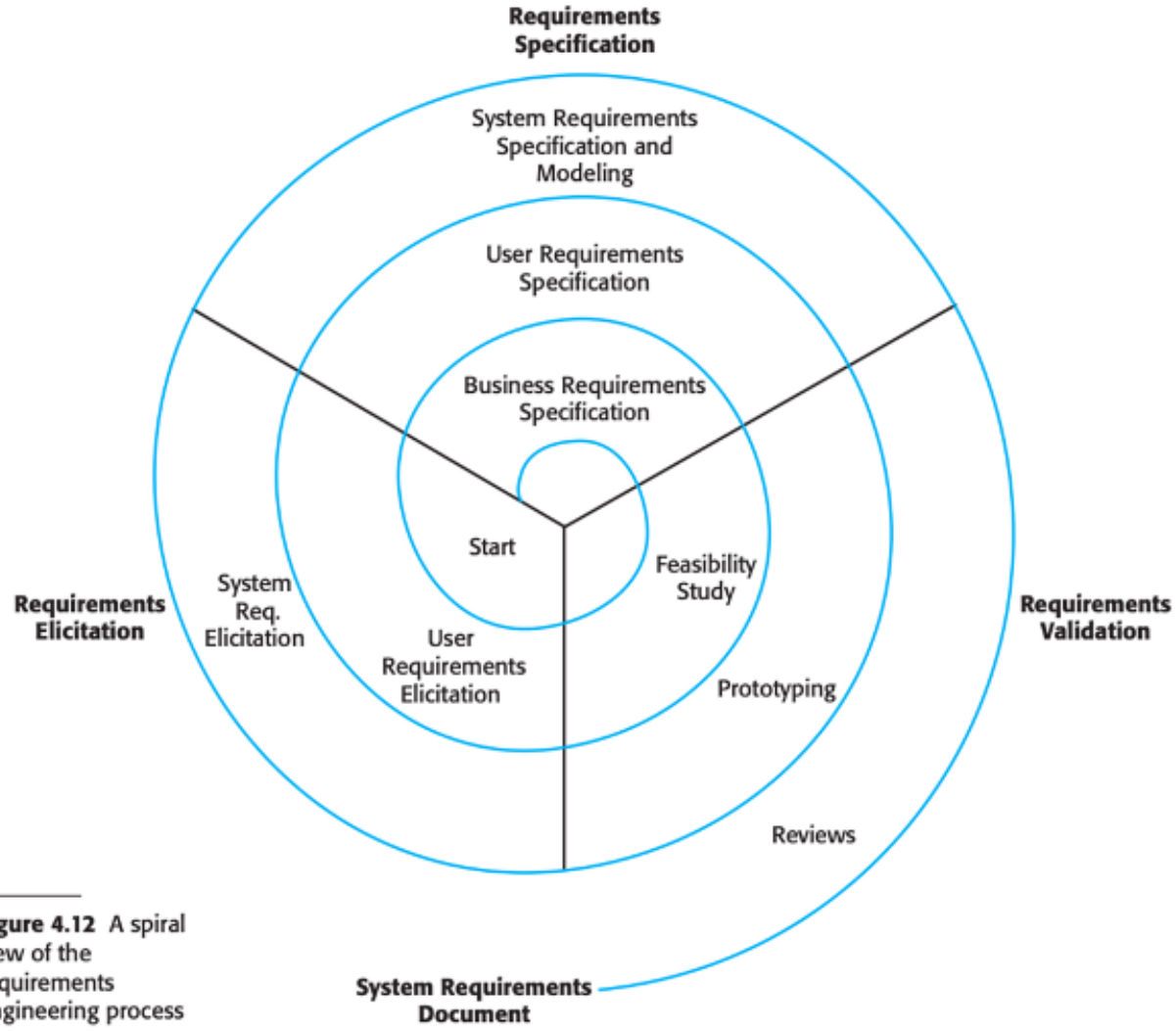
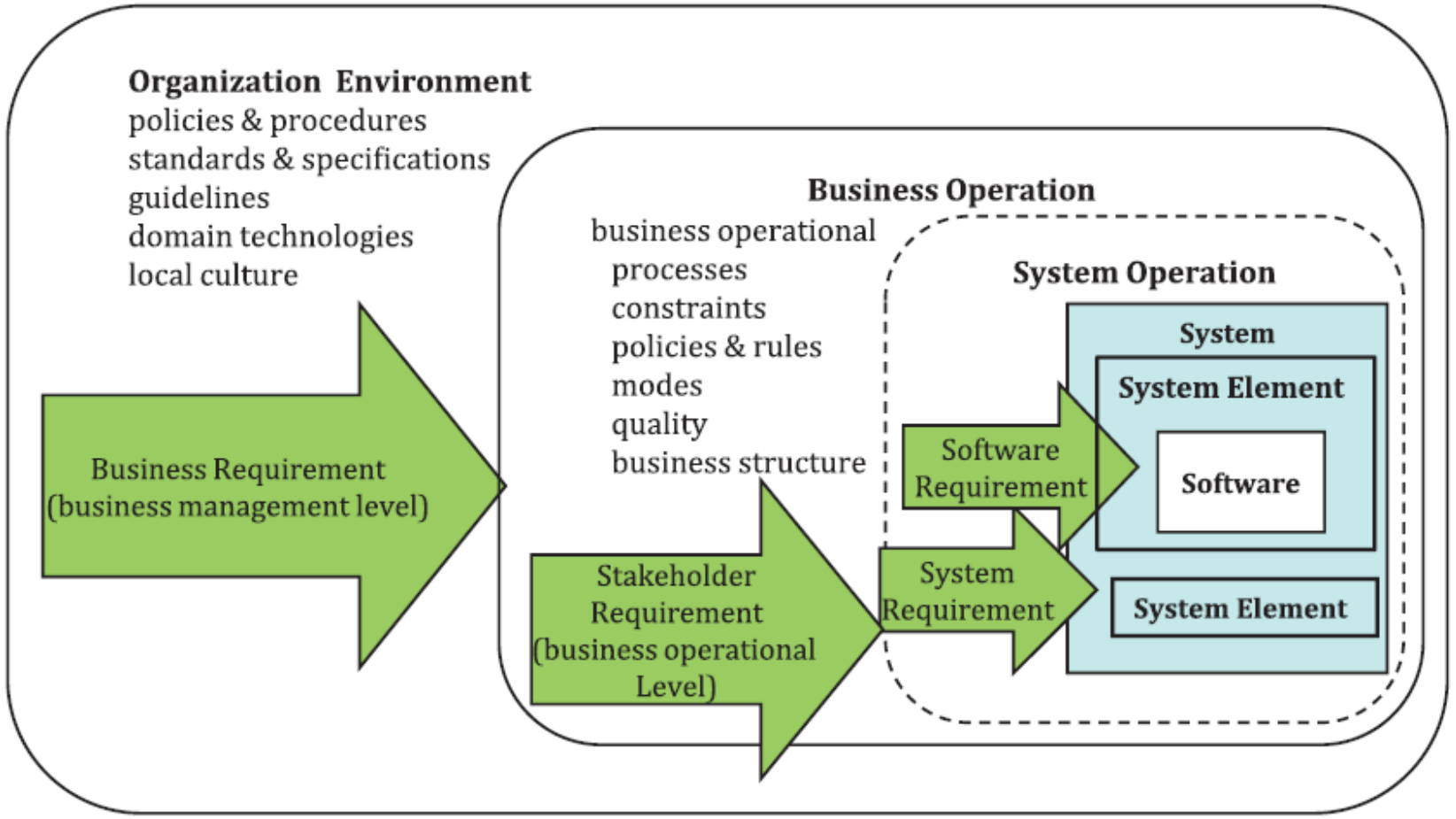


Figure 4.12 A spiral view of the requirements engineering process

Software Engineering, 10th edition, Ian Sommerville

CHALMERS UNIVERSITY OF TECHNOLOGY | UNIVERSITY OF GOTHENBURG

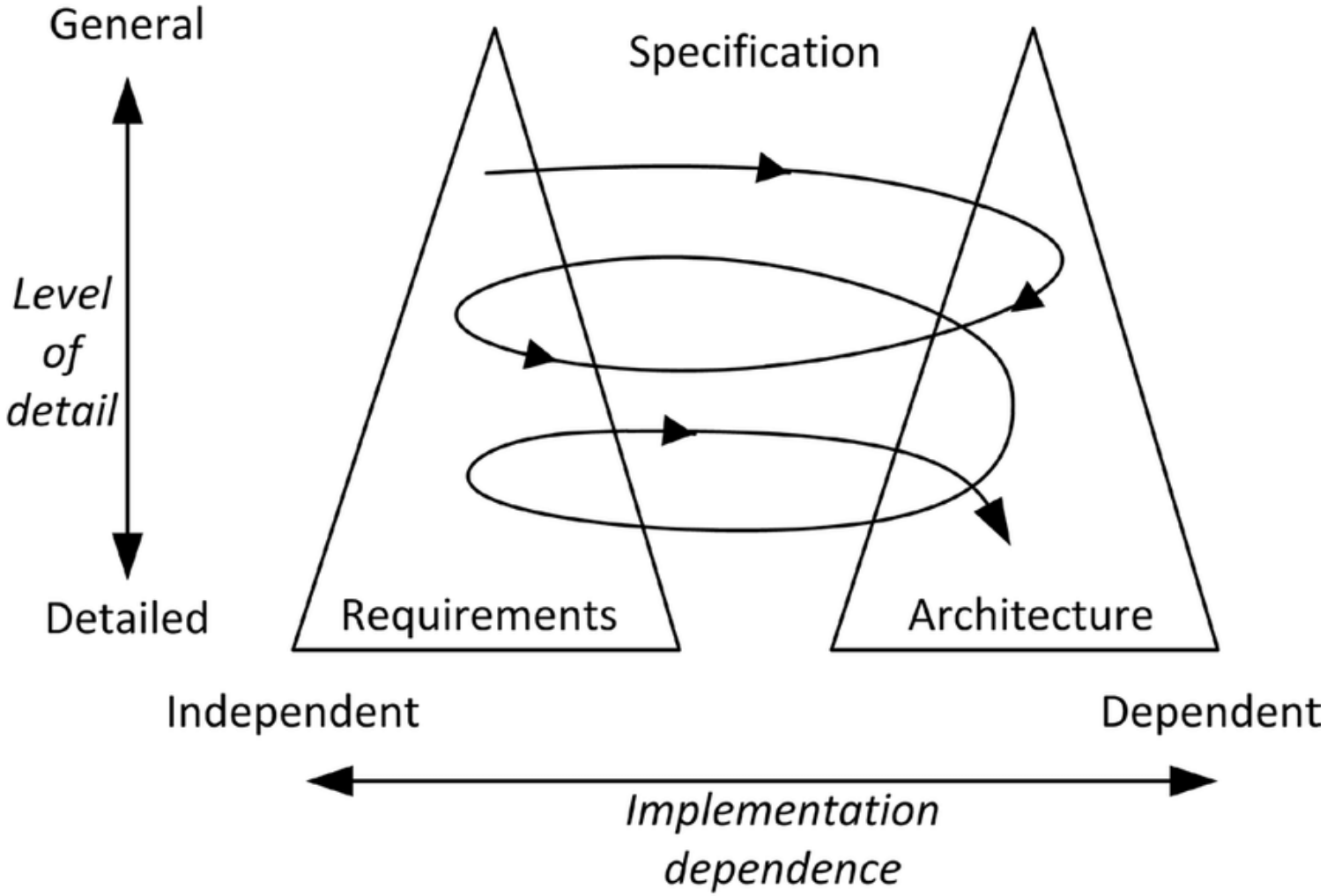
RE Background: Requirements Scopes



EEE/ISO/IEC 29148-2018
ISO/IEC/IEEE International Standard - Systems and software engineering -- Life cycle processes -- Requirements engineering

CHALMERS UNIVERSITY OF TECHNOLOGY | UNIVERSITY OF GOTHENBURG

RE Background: Twin Peaks Model



Nuseibeh, Bashar. "Weaving together requirements and architectures." *Computer* 34.3 (2001): 115-119.

(Briefly) AI for RE

CHALMERS UNIVERSITY OF TECHNOLOGY | UNIVERSITY OF GOTHENBURG

AI for RE

Extracting requirements from regulations

Extracting models from requirements

Generating Requirements

Classifying requirements

Generating Creativity Triggers

Evaluating quality of requirements

Generating Interview Questions

Finding traceability links

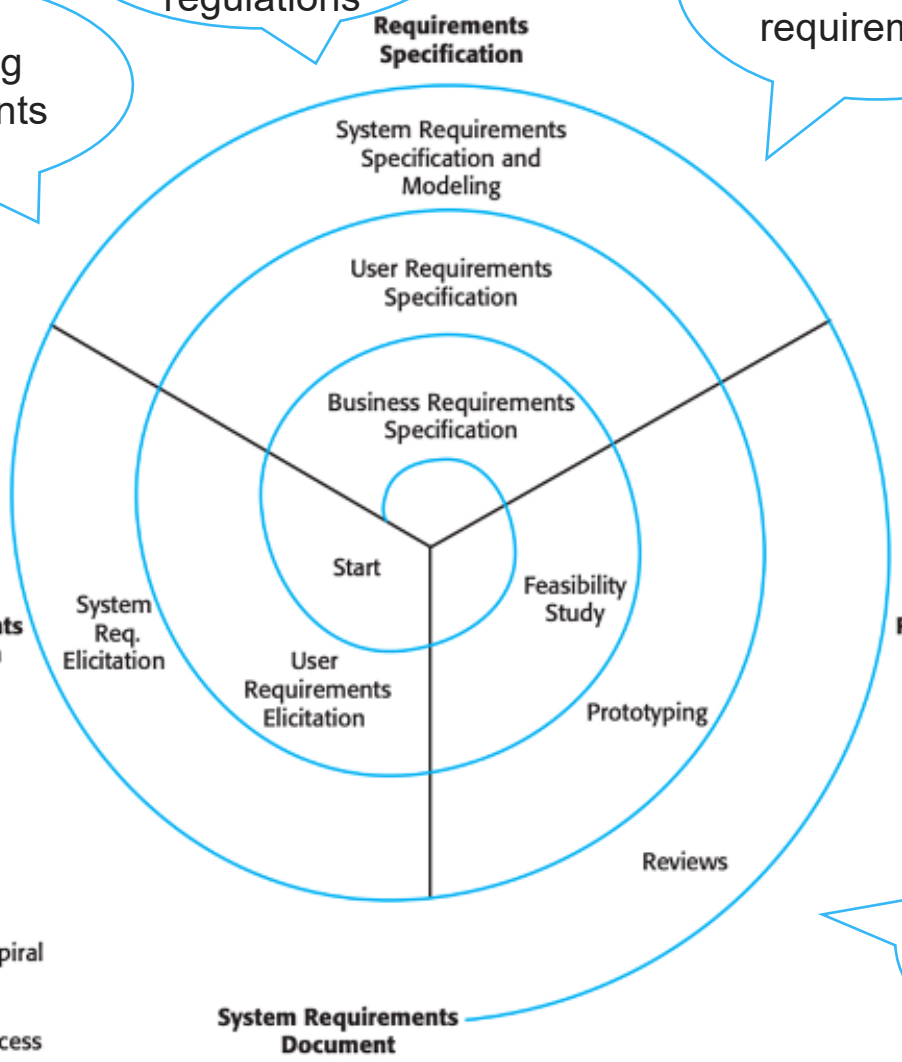


Figure 4.12 A spiral view of the requirements engineering process

Vasudevan, Poonkuzhali, and Sandeep Reddivari. "The role of generative ai models in requirements engineering: A systematic literature review." *Proceedings of the 2025 ACM Southeast Conference*. 2025.

CHALMERS UNIVERSITY OF TECHNOLOGY | UNIVERSITY OF GOTHENBURG

Gen AI for RE (Example)

- What prompt patterns perform best for what RE tasks? (GPT-3.5 turbo API)

Pattern	Classification	Tracing
Cognitive verifier	Classify the given list of requirements into functional (labelled as F) and non-functional requirements (labelled as NF). Ask me questions if needed to break the given task into smaller subtasks. All the outputs to the smaller subtasks must be combined before you generate the final output	List the IDs of requirements that are related to the [deprecated] feature in the requirements specification document below. Ask me questions if needed to break down the given task into smaller subtasks. All the outputs to the smaller subtasks must be combined before you generate the final output
Context manager	Classify the given list of requirements into functional (labelled as F) and non-functional requirements (labelled as NF). When you provide an answer, please explain the reasoning and assumptions behind your response. If possible, address any potential ambiguities or limitations in your answer, in order to provide a more complete and accurate response	List the IDs of requirements that are related to the [deprecated] feature from the requirements specification document below. When you provide an answer, please explain the reasoning and assumptions behind your response. If possible, address any potential ambiguities or limitations in your answer in order to provide a more complete and accurate response
Persona	Act as a requirements engineering domain expert, and classify the given list of requirements into functional (labelled as F) and non-functional requirements (labelled as NF)	Act as a requirements engineering domain expert, and list the IDs of requirements that are dependent on the [deprecated] feature in the following requirements specification document
Question refinement	Classify the given list of requirements into functional (labelled as F) and non-functional requirements (labelled as NF). If needed, suggest a better version of the question to use that incorporates information specific to this task, and ask me if I would like to use your question instead	List the IDs of requirements that are related to the [deprecated] feature from the requirements specification document below. If needed, suggest a better version of the question to use that incorporates information specific to this task, and ask me if I would like to use your question instead
Template	Read the following list of requirements, and return the IDs of non-functional requirements only. Write the result as a list like (ID=X) (ID=Y) (ID=Z), where X, Y, and Z are IDs of non-functional requirements	List the IDs of requirements that are related to the [deprecated] feature in the requirements specification document below. Follow the provided template when generating the output: ID list: X.X.X.X; X.X.X; X.X.X.X; etc.

Ronanki, K., Cabrero-Daniel, B., Horkoff, J., Berger, C. (2024). Requirements Engineering Using Generative AI: Prompts and Prompting Patterns. In: Nguyen-Duc, A., Abrahamsson, P., Khomh, F. (eds) Generative AI for Effective Software Development. Springer, Cham.

CHALMERS UNIVERSITY OF TECHNOLOGY | UNIVERSITY OF GOTHENBURG

FR vs NFR

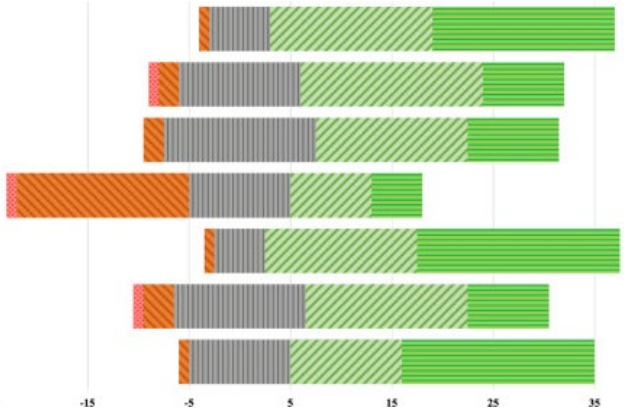
Prompt Pattern	Precision	Recall	F-Score	Accuracy
Cognitive Verifier	76,7% 76,7% 73,3%	87,1% 83,3% 84,0%	81,6% 79,9% 78,3%	78,0% 74,8% 75,6%
Context Manager	80,6% 79,1% 86,2%	73,0% 72,4% 49,3%	76,6% 75,6% 62,8%	73,6% 71,6% 75,6%
Persona	71,9% 79,9% 82,1%	81,0% 80,9% 81,0%	76,2% 80,4% 81,6%	71,2% 76,0% 77,6%
Question Refinement	84,6% 80,8% 83,7%	71,7% 76,6% 73,0%	77,6% 78,7% 78,0%	76,0% 74,4% 75,6%
Template	75,0% 66,0% 65,6%	91,0% 89,9% 91,3%	82,2% 76,1% 76,4%	77,2% 68,8% 68,8%

Traceability

Prompt Pattern	Precision	Recall	F-Score	Accuracy
Cognitive Verifier	35,2% 37,2% 42,3%	61,5% 63,2% 63,2%	44,8% 46,8% 50,7%	88,9% 89,5% 91,0%
Context Manager	26,5% 25,9% 30,5%	59,8% 51,3% 52,9%	36,7% 34,5% 38,7%	85,0% 85,8% 87,8%
Persona	41,3% 38,5% 33,8%	76,9% 76,1% 71,8%	53,7% 51,1% 46,0%	90,4% 89,4% 87,7%
Question Refinement	43,2% 45,4% 45,2%	62,4% 64,0% 60,7%	51,0% 53,2% 51,8%	91,3% 91,8% 91,8%
Template	38,3% 30,7% 33,2%	61,5% 62,6% 56,4%	47,2% 41,3% 41,7%	89,9% 86,9% 88,6%

Recommendations for Efficient and Responsible SE LLM Adoption

- Multi-case study with three organisations that use LLMs within their SE activities
- Complementary online survey with software practitioners from various industries (43 answers)
  - R1: It is beneficial to use the LLMs as assistants with human oversight rather than try and automate the task using the LLMs.
  - R2: When using LLMs as AI assistants, it is better to prioritise the usefulness of the LLMs' outputs rather than output accuracy.
  - R3: The relevant stakeholders' satisfaction should be used as part of the LLM-generated outputs evaluation criteria.
  - R4: It is not beneficial to use any task-specific fine-tuned LLMs like GitHub Copilot for tasks they were not designed for.
  - R5: It is crucial to develop and implement human oversight and validation mechanisms while using LLMs' assistance for any task.
  - R6: There is a need to create room for incorporating LLMs into business processes by restructuring (parts of) them.
  - R7: Knowledge-sharing activities like seminars and specialised training are crucial for the users of the LLMs.



Ronanki, K., Cabrero-Daniel, B., Herda, T., Sitkovich, S., Horkoff, J., & Berger, C. (2026). Recommendations for efficient and responsible LLM adoption within industrial software development. Information and Software Technology, 108171.

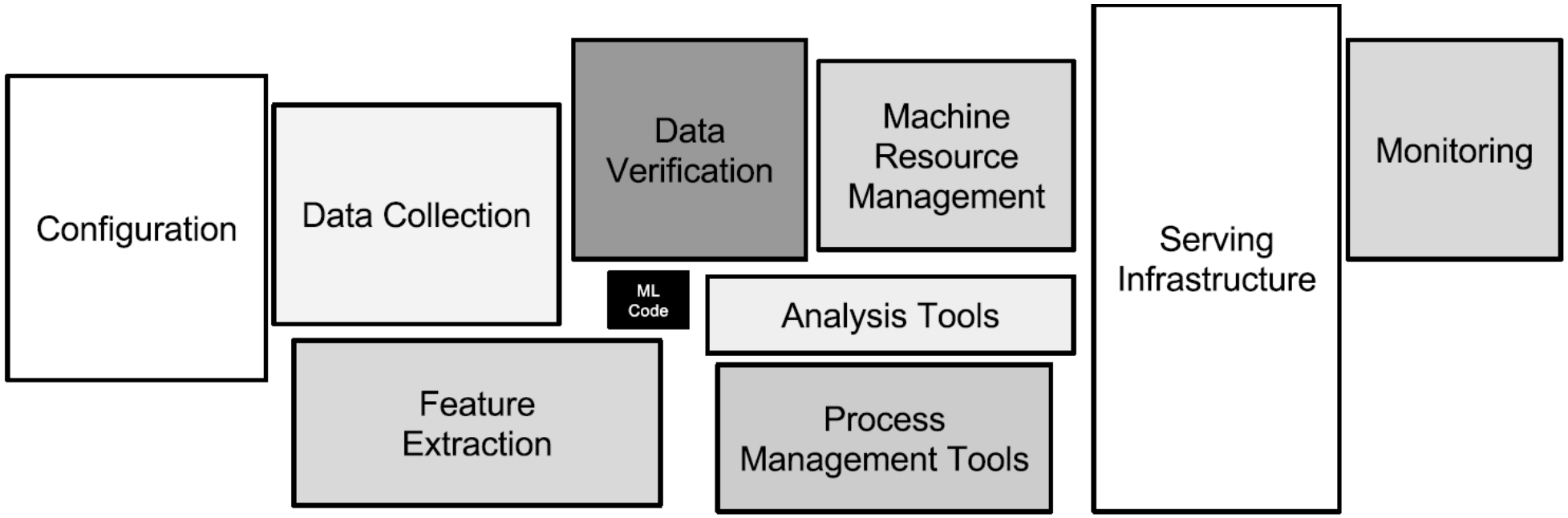
CHALMERS UNIVERSITY OF TECHNOLOGY | UNIVERSITY OF GOTHENBURG

AI Systems

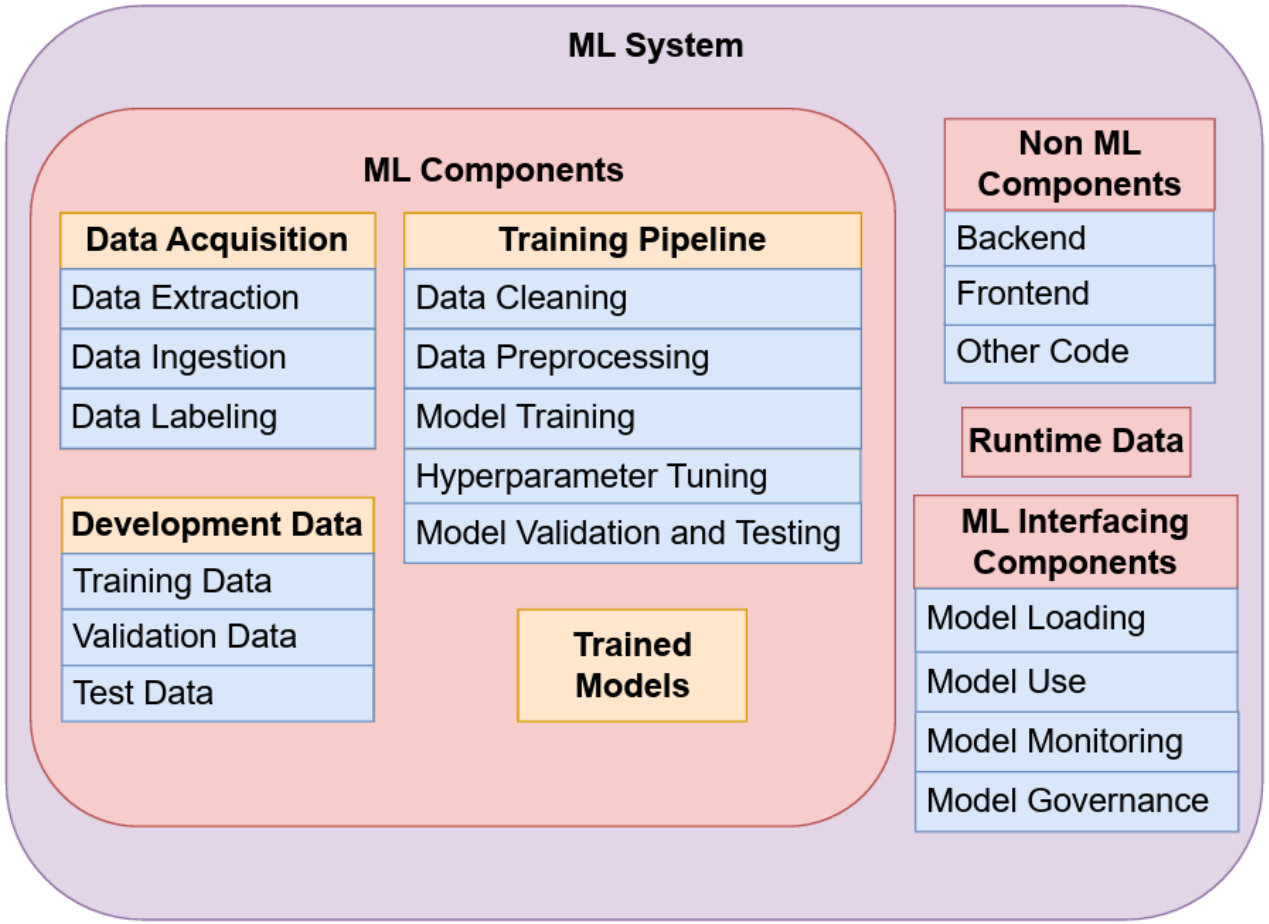
CHALMERS UNIVERSITY OF TECHNOLOGY | UNIVERSITY OF GOTHENBURG

AI Systems – Systems with AI Components

Even in 2015: Sculley et al., 2015, NIPS



AI Systems – Updated View



Habibullah, K.M., Gay, G. and Horkoff, J., 2024, September. A Framework for Managing Quality Requirements for Machine Learning-Based Software Systems. In International Conference on the Quality of Information and Communications Technology (pp. 3-20). Cham: Springer Nature Switzerland.

CHALMERS UNIVERSITY OF TECHNOLOGY | UNIVERSITY OF GOTHENBURG

Requirements Engineering (RE) for AI Systems

CHALMERS UNIVERSITY OF TECHNOLOGY | UNIVERSITY OF GOTHENBURG

RE for AI/ML Systems

- Data is a first-class component

  → Data requirements
- Different principles

  → Not possible to specify complete requirements upfront
- Different emphasis on qualities

  → Accuracy, transparency, reproducibility…
- Different challenges

  → Unintended bias, non-determinism, opacity

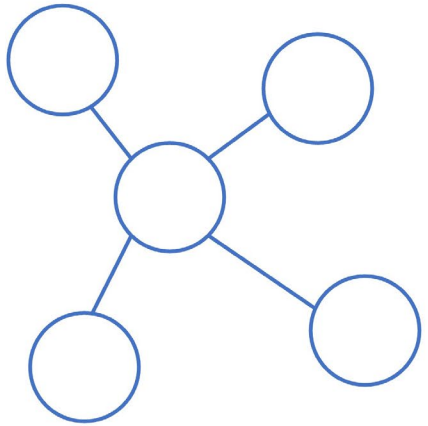
CHALMERS UNIVERSITY OF TECHNOLOGY | UNIVERSITY OF GOTHENBURG

RE for AI: Big Space of Work

SLRs and Mappings:

- Belani, Hrvoje, Marin Vukovic, and Željka Car. "**Requirements engineering challenges in building AI-based complex systems**." *2019 IEEE 27th international requirements engineering conference workshops (REW)*. IEEE, 2019. (166 citations)
- Villamizar, Hugo, Tatiana Escovedo, and Marcos Kalinowski. "**Requirements engineering for machine learning: A systematic mapping study**." 2021 47th Euromicro conference on software engineering and advanced applications (SEAA). IEEE, 2021. (97 citations)
- Ahmad, Khlood, et al. "**Requirements engineering for artificial intelligence systems: A systematic mapping study**." Information and Software Technology 158 (2023): 107176. (214 citations)
- Habiba, Umm-E., et al. "**How mature is requirements engineering for AI-based systems? A systematic mapping study on practices, challenges, and future research directions**." Requirements Engineering 29.4 (2024): 567-600. (57 citations)

Case Studies – Perception Systems in Autonomous Driving



FAMER

Facilitating Multi-Party Engineering of Requirements

UNIVERSITY OF GOTHENBURG

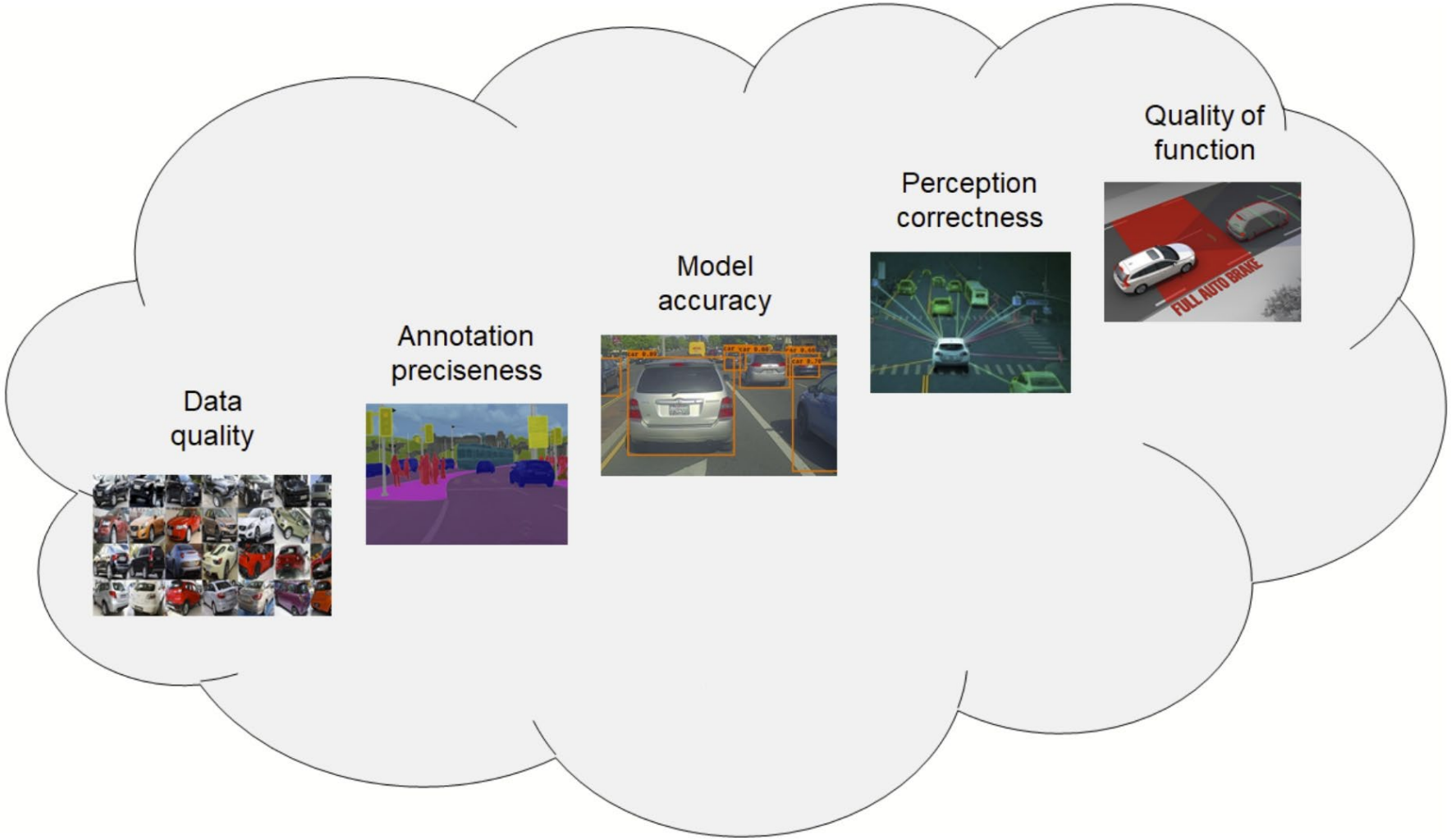
RI. SE

zenseact

KOGNIC

VOLVO

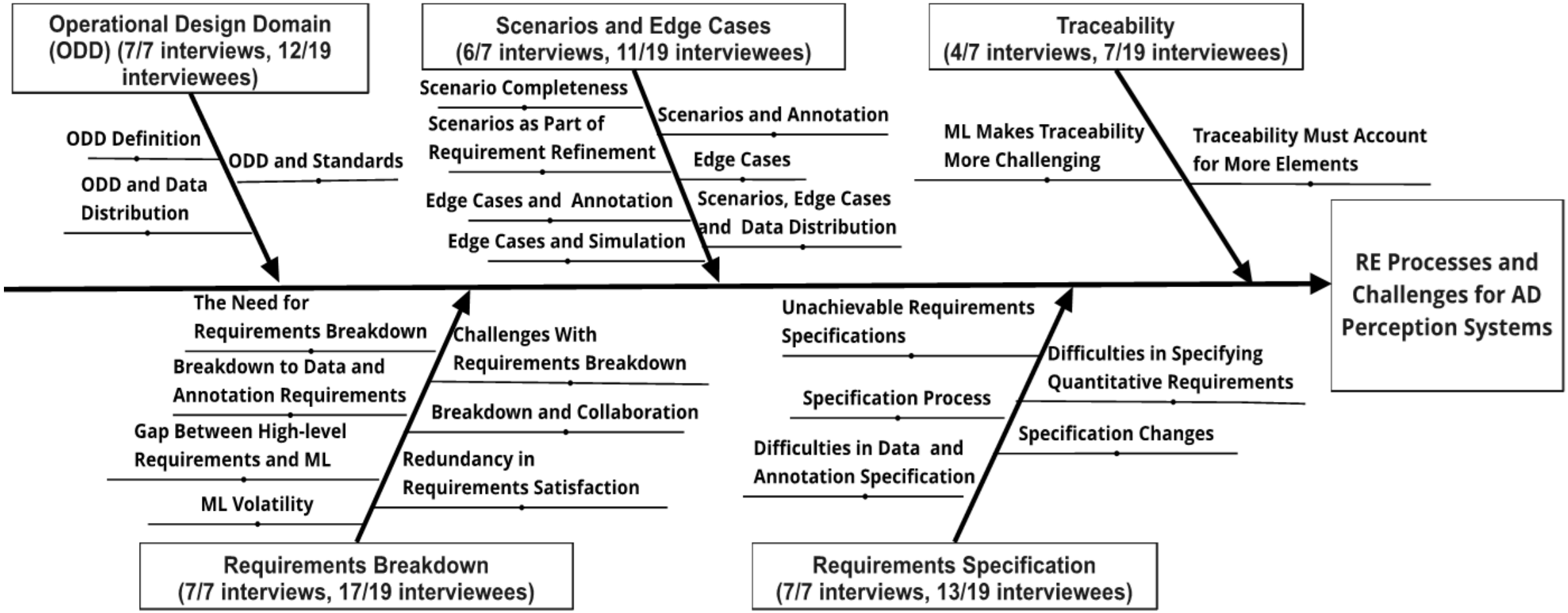
Automotive Perception System



Habibullah, K.M., Gay, G. and Horkoff, J., 2024, September. A Framework for Managing Quality Requirements for Machine Learning-Based Software Systems. In International Conference on the Quality of Information and Communications Technology (pp. 3-20). Cham: Springer Nature Switzerland.

CHALMERS UNIVERSITY OF TECHNOLOGY | UNIVERSITY OF GOTHENBURG

General RE Topics And Challenges For ML Systems (DAS)



K. M. Habibullah, H. -M. Heyn, G. Gay, J. Horkoff, E. Knauss, M. Borg, A. Knauss, H. Sivencrona, P. Li. Jing "Requirements engineering for automotive perception systems" 29th International Working Conference on Requirement Engineering: Foundation for Software Quality, Springer, 2023.

Case Study – Key Findings thus Far

- No clear links between quality of features, models, data, annotation.
  - They are linked, but no formulas or way to quantitatively predict quality
- Traditional RE methods are only followed to a certain point
  - Boundary of features to ML involves more negotiation, experimentation, trickle
- Safety and standard are key challenging issues
- Requirements as data
- Requirements over data, annotations
- Requirements as scenarios
- Requirements as requirements – high-level, to sensors, to components, to external parties
  - Abandon notion of complete and correct requirements specification for ML sub-parts
  - But requirements can and should still play a role

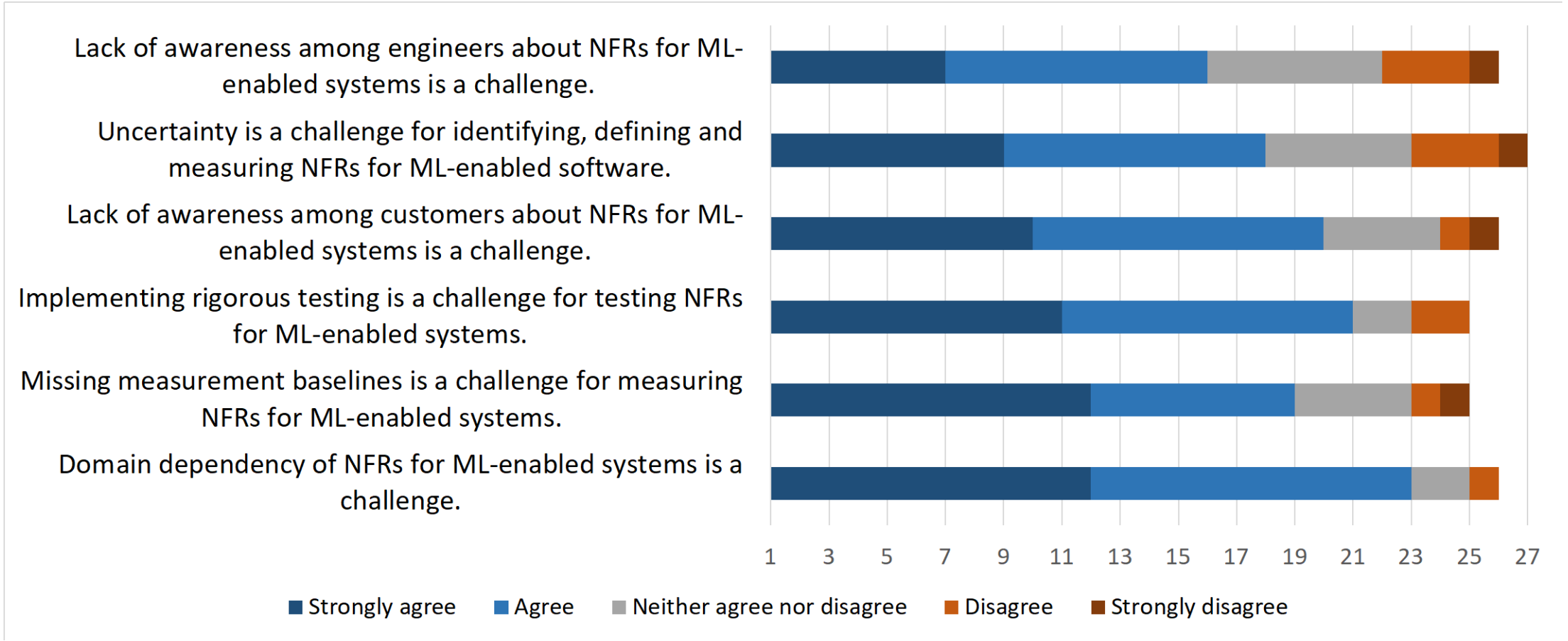
Qualities of ML Solutions

PhD: Khan Mohammad Habibullah

- Accuracy & Performance (Correctness)
- Fairness
- Transparency
- Security & Privacy
- Testability
- Reliability
- What else?
  - Trainability?
  - Maintainability?
  - Sustainability?

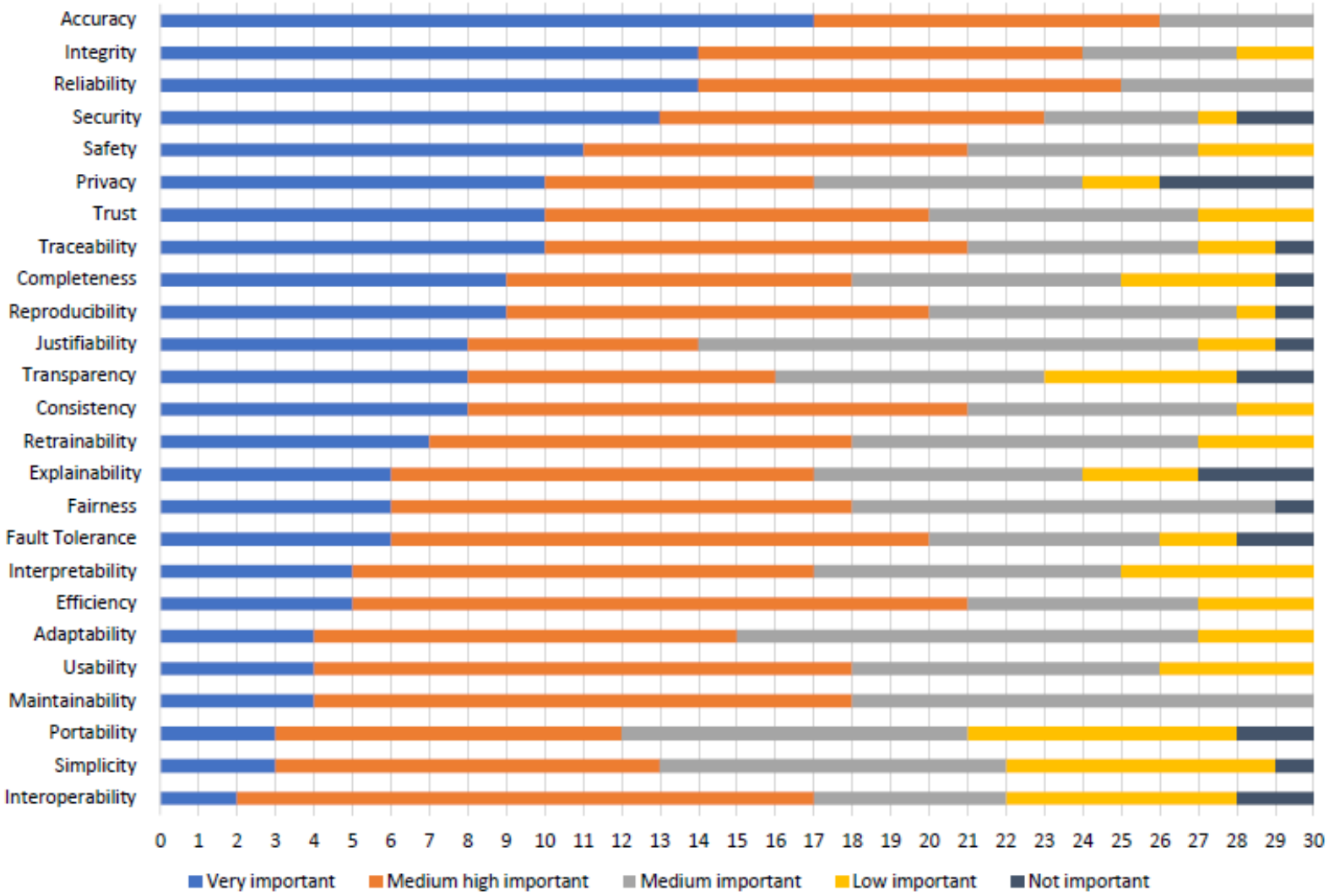
[Wan et al., 2020] “Instead of functional requirements in non-ML software systems, quantitative measures comprise the majority of requirements for ML systems.”

NFR and NFR Measurement–Related Challenges for ML Systems



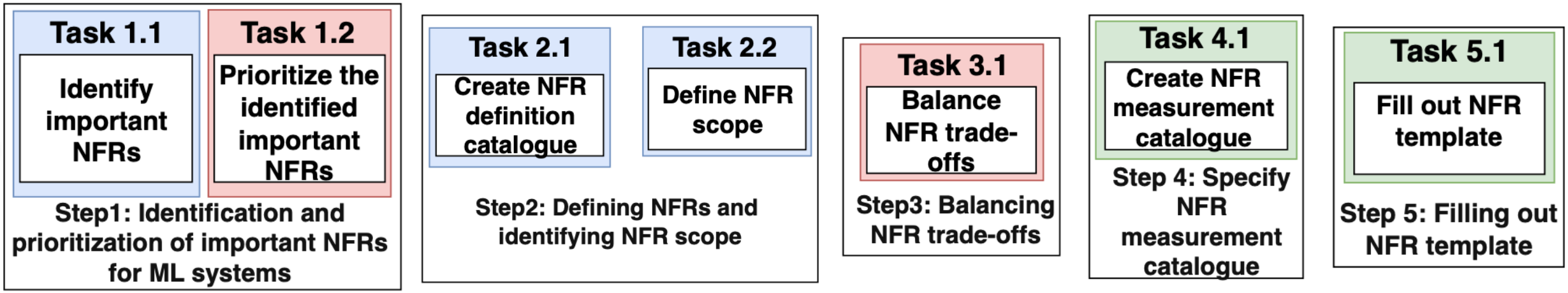
Habibullah, K.M., Gay, G. and Horkoff, J., 2023. Non-functional requirements for machine learning: Understanding current use and Challenges among practitioners. *Requirements Engineering*, pp.1-34.

Survey Results – NFR for ML Ranking



Habibullah, K.M., Gay, G. and Horkoff, J., 2023. Non-functional requirements for machine learning: Understanding current use and Challenges among practitioners. Requirements Engineering, pp.1-34.

A Preliminary Framework for Managing Quality Requirements for ML Systems



Habibullah, K.M., Gay, G. and Horkoff, J., 2024, September. A Framework for Managing Quality Requirements for Machine Learning-Based Software Systems. In International Conference on the Quality of Information and Communications Technology (pp. 3-20). Cham: Springer Nature Switzerland.

Systems with ML: Scope

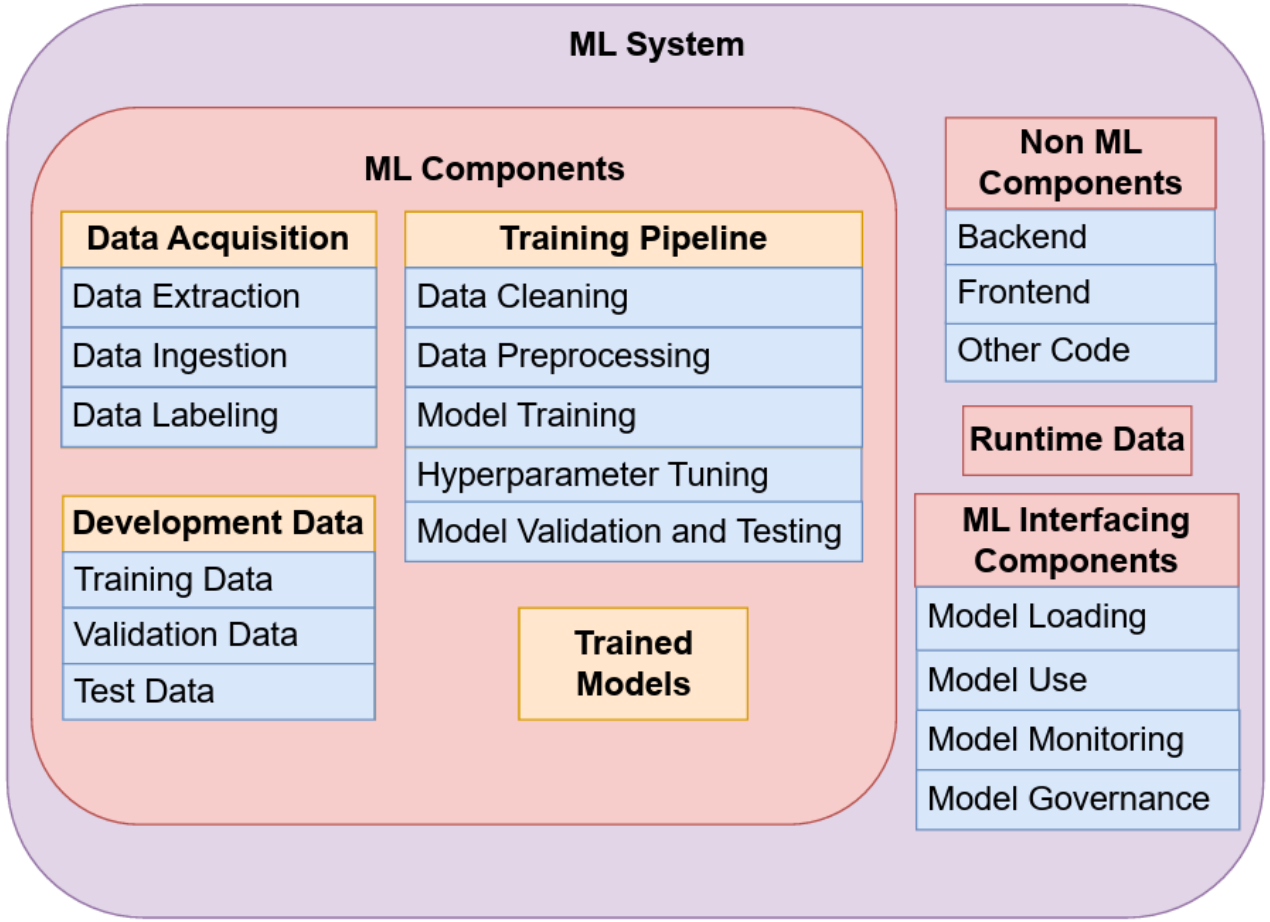


Table 3. Definitions of important NFR Types for autonomous perception system.

NFRs	Definition
Accuracy	The ability to precisely and correctly identify objects defined in the Operational Design Domain
Reliability	The system consistently identify objects correctly and without sudden errors, ensuring the system can react appropriately to the environment
Safety	Objects are accurately identified and located—preventing collisions, accidents, and other potential hazards—even in the presence of poor visibility, occlusions, and unexpected object behaviors

Table 4. Examples of NFR scoping for an APS.

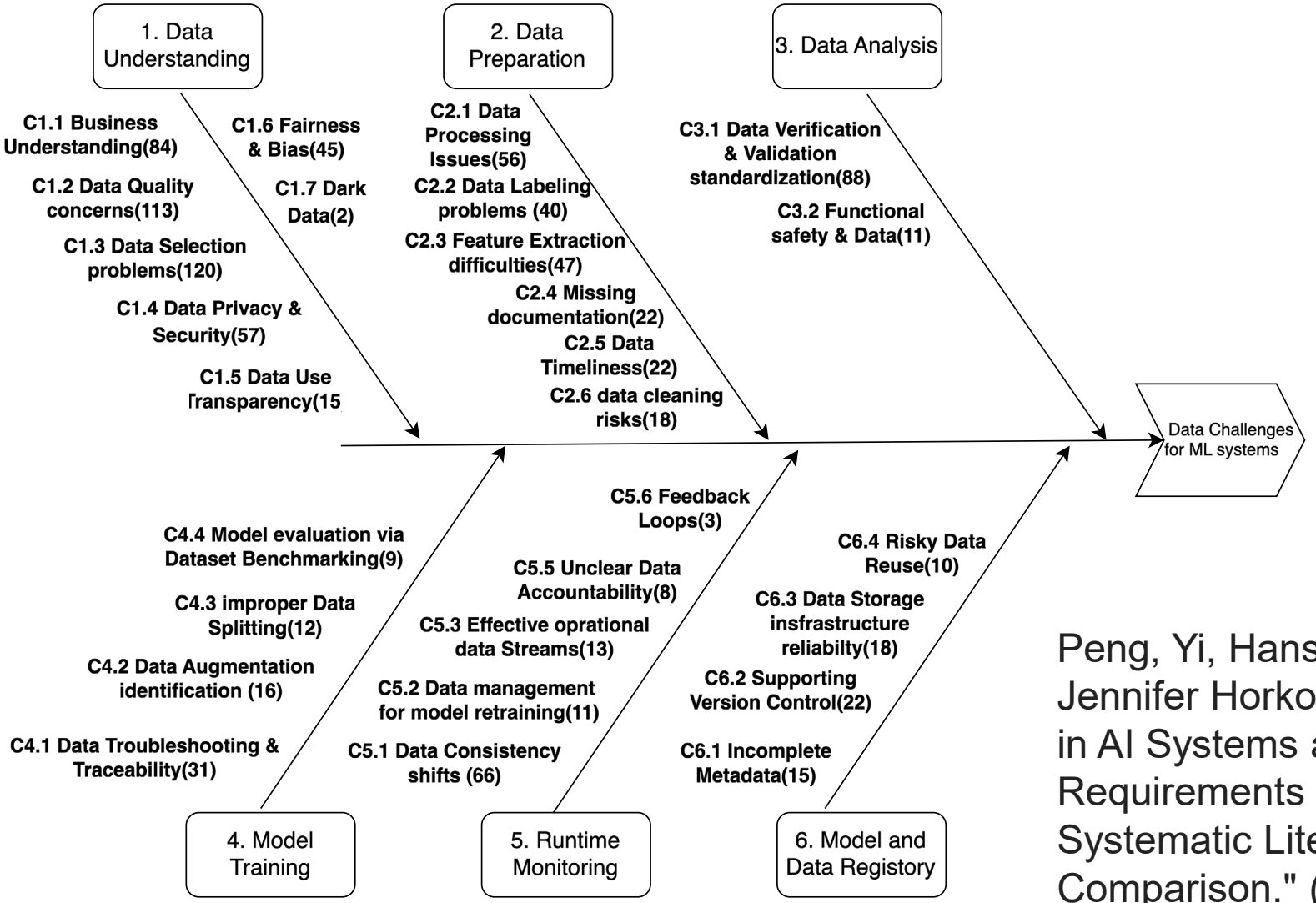
NFR	System Element the NFR Type Can be Defined Over				
	Training Data	ML Pipeline	ML Model	Results	Whole System
Accuracy	✖	✔	✔	✔	✔
Explainability	✖	✔	✔	✔	✔
Integrity	✔	✔	✔	✔	✔
Performance	✔	✔	✔	✔	✔
Reliability	✔	✔	✔	✔	✔
Safety	✔	✔	✔	✔	✔

Habibullah, K.M., Gay, G. and Horkoff, J., 2024, September. A Framework for Managing Quality Requirements for Machine Learning-Based Software Systems. In International Conference on the Quality of Information and Communications Technology (pp. 3-20).

CHALMERS UNIVERSITY OF TECHNOLOGY | UNIVERSITY OF GOTHENBURG

Data-related challenges for ML systems

PhD: Yi Peng

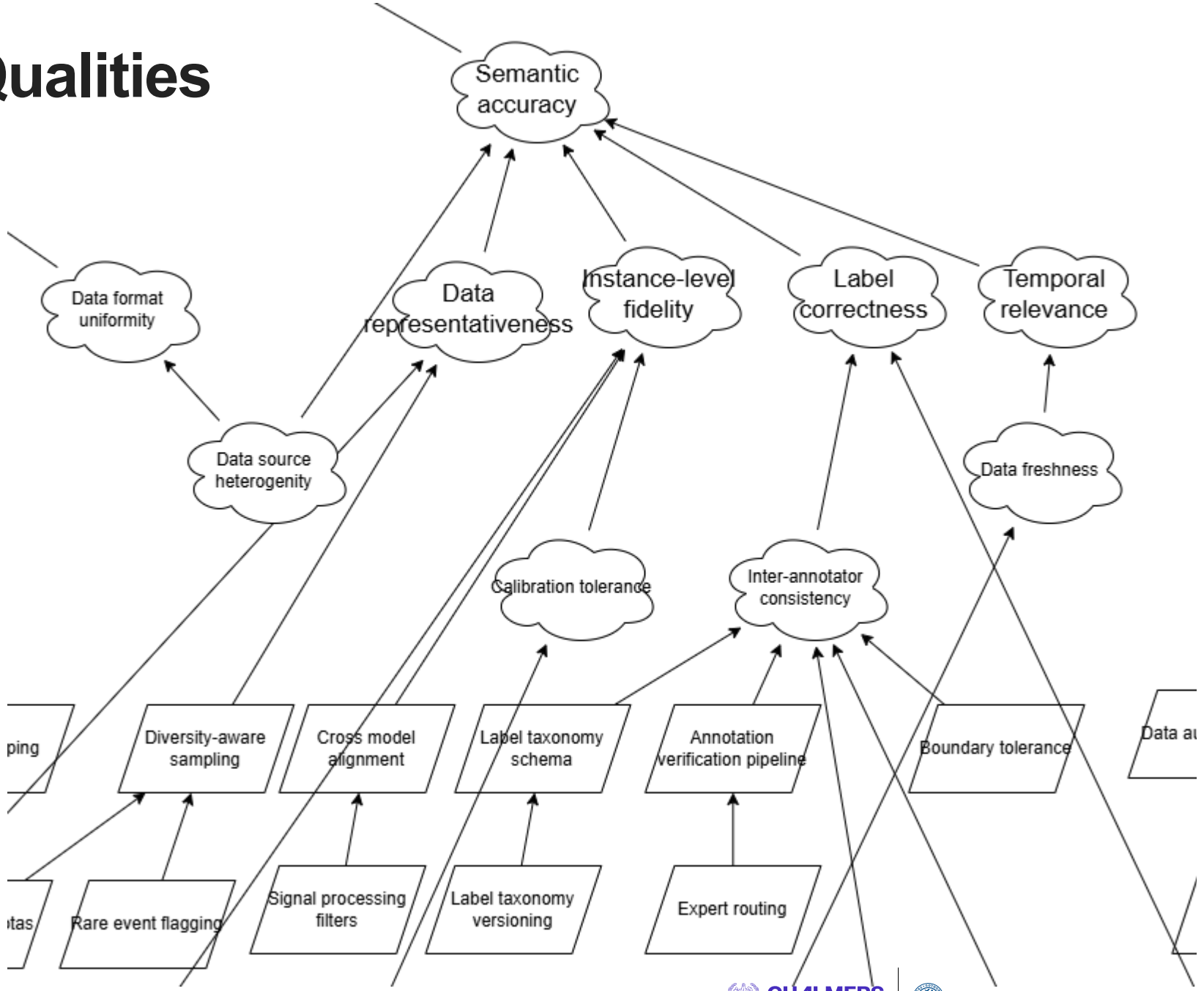


Peng, Yi, Hans-Martin Heyn, and Jennifer Horkoff. "Data Challenges in AI Systems and their Solutions: A Requirements and AI Engineering Systematic Literature Review and Comparison." (2026).

Data Requirements Qualities

Yi Peng, Hans-Martin Heyn, Jennifer Horkoff: Data Challenges in AI Systems and their Solutions: A Requirements and AI Engineering Systematic Literature Review and Comparison

Peng, Yi, Hans-Martin Heyn, Mathieu Laurent, and Jennifer Horkoff. Data Quality… Work in progress



Data Annotation Requirements Examples

Req ID: Accuracy-1
Scenario type: Standard
Vertical trace: C-4 Taxonomy (Pedestrian)
Req description: When <a person is on foot (moving, stationary) or pushing a vehicle (bike, scooter)>, in < an urban environment>, ***the annotator*** shall <label as pedestrian and draw a 2D bounding box>, <the box includes the person and pushed vehicle (if applicable) as a single entity>.
Rationale: Pushing a vehicle has the velocity profile of a moving pedestrian. The prediction module treats this cluster as a single low-speed entity.
Acceptance criteria: Verified via random sampling in quality assurance audit. The bounding box must have a intersection over union > 0.95 against auditor's ground truth. Mis-classification counts as critical error.

Standard Scenario:

Default scenarios featuring no exceptional or error conditions.

Req ID: Robustness-1
Scenario type: Edge-case
Vertical trace: C-2 Representation plan
Req description: When <a object appears on a reflective surface without a corresponding physical object in the 3D space>, in < an urban environment>, ***the annotator*** shall <ignore this object>.
Rationale: Reflections create ghost targets for the perception system, leading to false braking.
Acceptance criteria: Verified via random sampling in quality assurance audit. No false positives on reflective surfaces.

Edge-case Scenario:

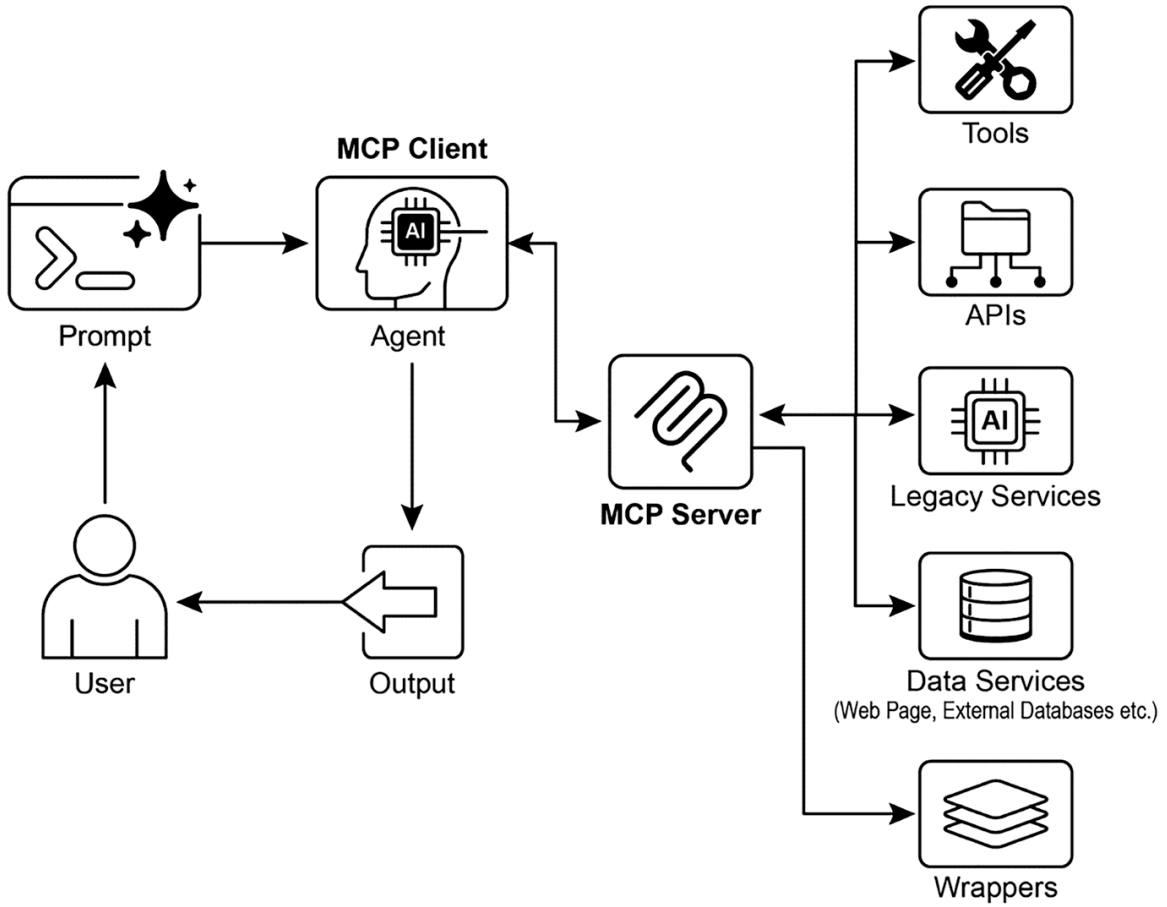
Rules for rare, atypical, or complex situations falling at the boundary of expected behavior.

Peng, Yi, et al. "Data Annotation: A Requirements Engineering for Machine Learning Systems Perspective." 2025 IEEE 33rd International Requirements Engineering Conference (RE). IEEE, 2025.

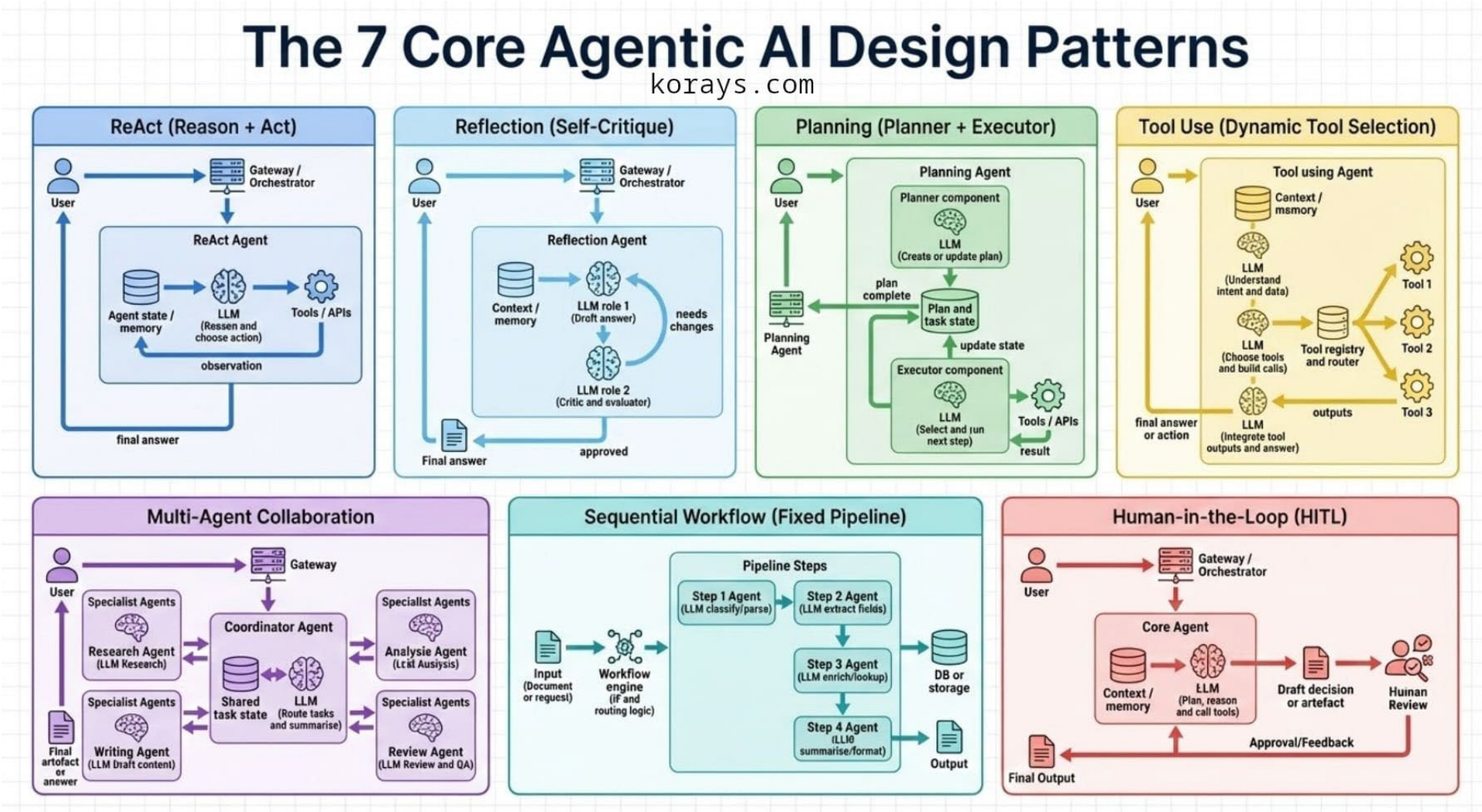
Agentic AI Systems

Agentic systems – Systems with AI Agents as Components

From: Gullí, Antonio. *Agentic design patterns: a hands-on guide to building intelligent systems*. Springer Nature, 2025.



Agentic systems

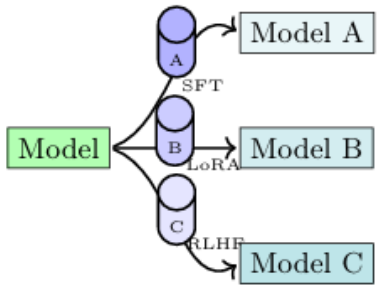
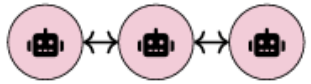
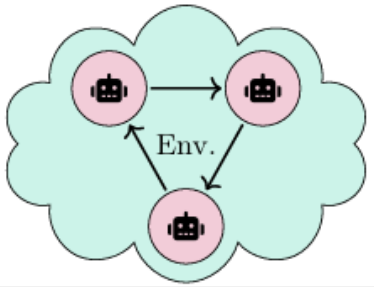


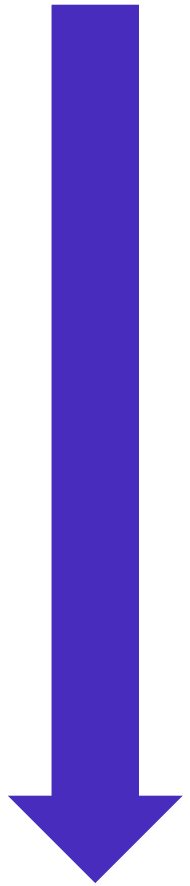
https://korays.com/a-brief-look-into-agentic-design-patterns/

A Comprehensive Survey of Self-Evolving AI Agents

A New Paradigm Bridging Foundation Models and Lifelong Agentic Systems

Jinyuan Fang[*1], Yanwen Peng[*2], Xi Zhang[*1], Yingxu Wang[3], Xinhao Yi[1], Guibin Zhang[4], Yi Xu[5], Bin Wu[6], Siwei Liu[7], Zihao Li[1], Zhaochun Ren[8], Nikos Aletras[2], Xi Wang[2], Han Zhou[5], Zaiqiao Meng[1] ✉

Paradigm	Interaction & Feedback	Key Techniques	Diagram
Model Offline Pretraining (MOP)	Model ⇔ Static data (loss/backprop)	• Transformer Pretraining (Causal LM, Masked LM, NSP) • BPE / SentencePiece • MoE & Pipeline Parallelism	
Model Online Adaptation (MOA)	Model ⇔ Supervision (labels/scores/rewards)	• Task Fine-tuning • Instruction Tuning • LoRA / Adapters / Prefix-Tuning • RLHF (RLAIF, DPO, PPO) • Multi-Modal Alignment • Human Alignment	
Multi-Agent Orchestration (MAO)	$Agent_1$ ⇔ $Agent_2$ (message exchange)	• Multi-Agent Systems • Self-Reflection • Multi-Agent Debate • Chain-of-Thought Ensemble • Function / Tool Calling / MCP	
Multi-Agent Self-Evolving (MASE)	Agents ⇔ Environment (signals from env.)	• Behaviour Optimisation • Prompt Optimisation • Memory Optimisation • Tool Optimisation • Agentic Workflow Optimisation	



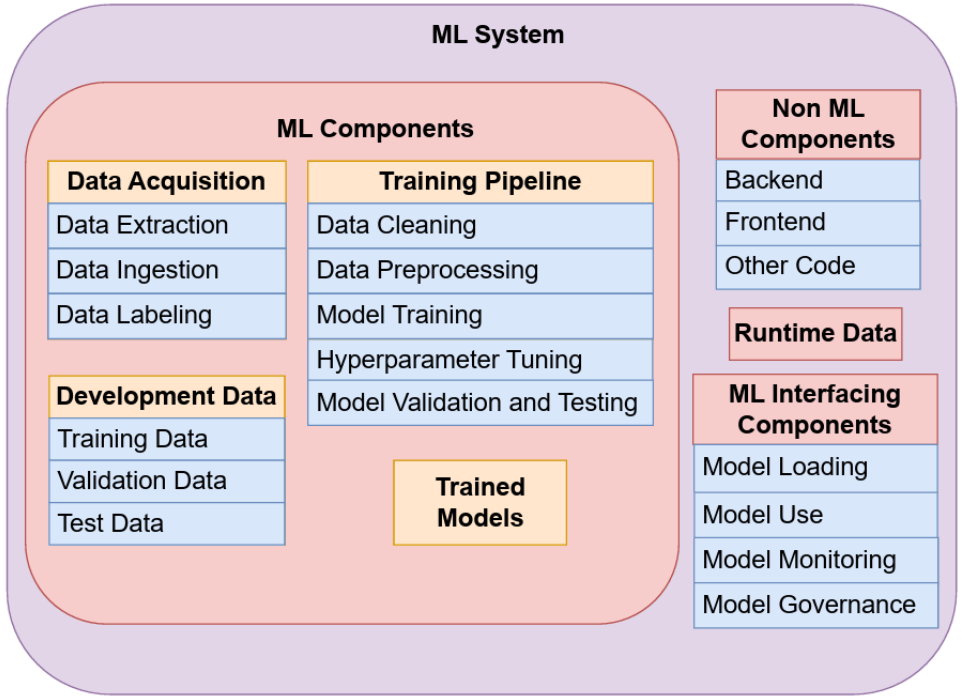
We are here?

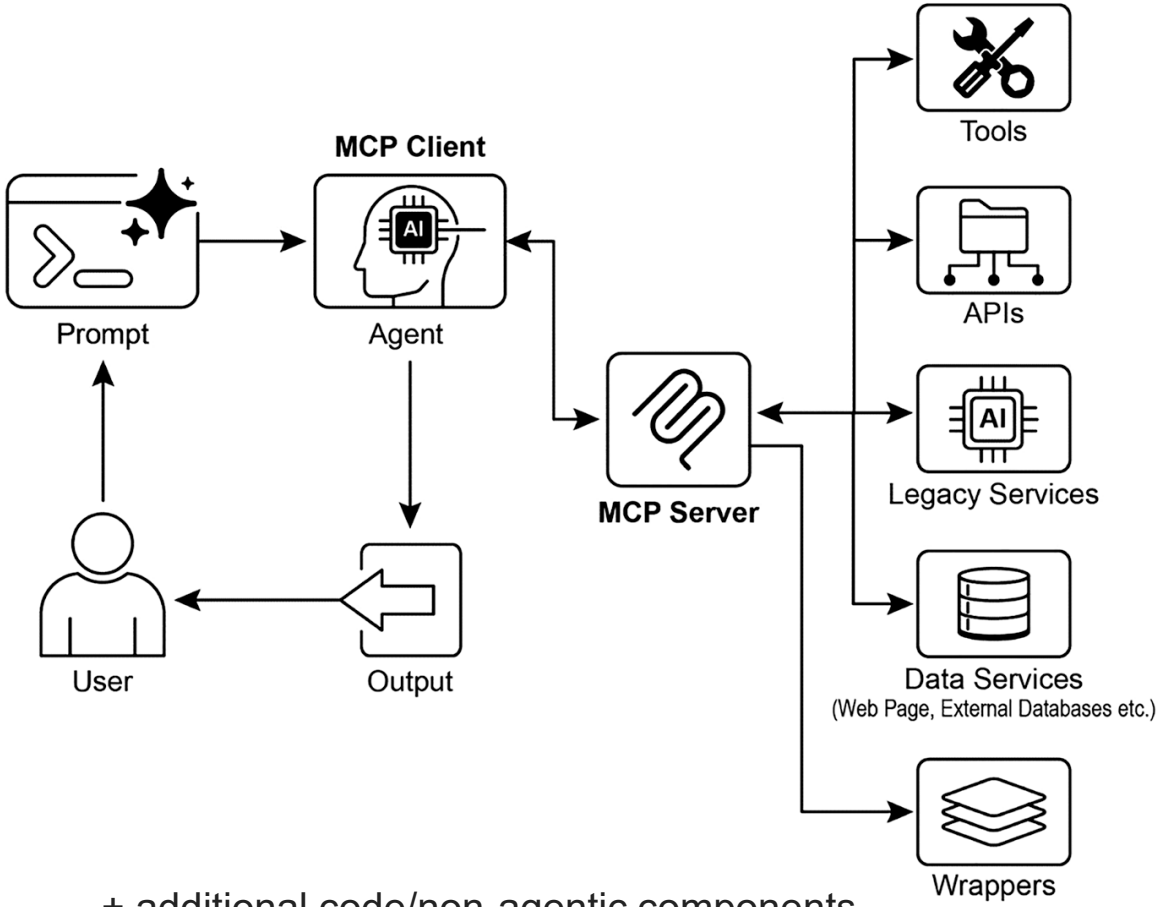
RE for Agentic Systems (forward looking)

CHALMERS UNIVERSITY OF TECHNOLOGY | UNIVERSITY OF GOTHENBURG

Different Components/Architectures

From: Gullí, Antonio. *Agentic design patterns: a hands-on guide to building intelligent systems*. Springer Nature, 2025.





+ additional code/non-agentic components
+ ?

CHALMERS UNIVERSITY OF TECHNOLOGY | UNIVERSITY OF GOTHENBURG

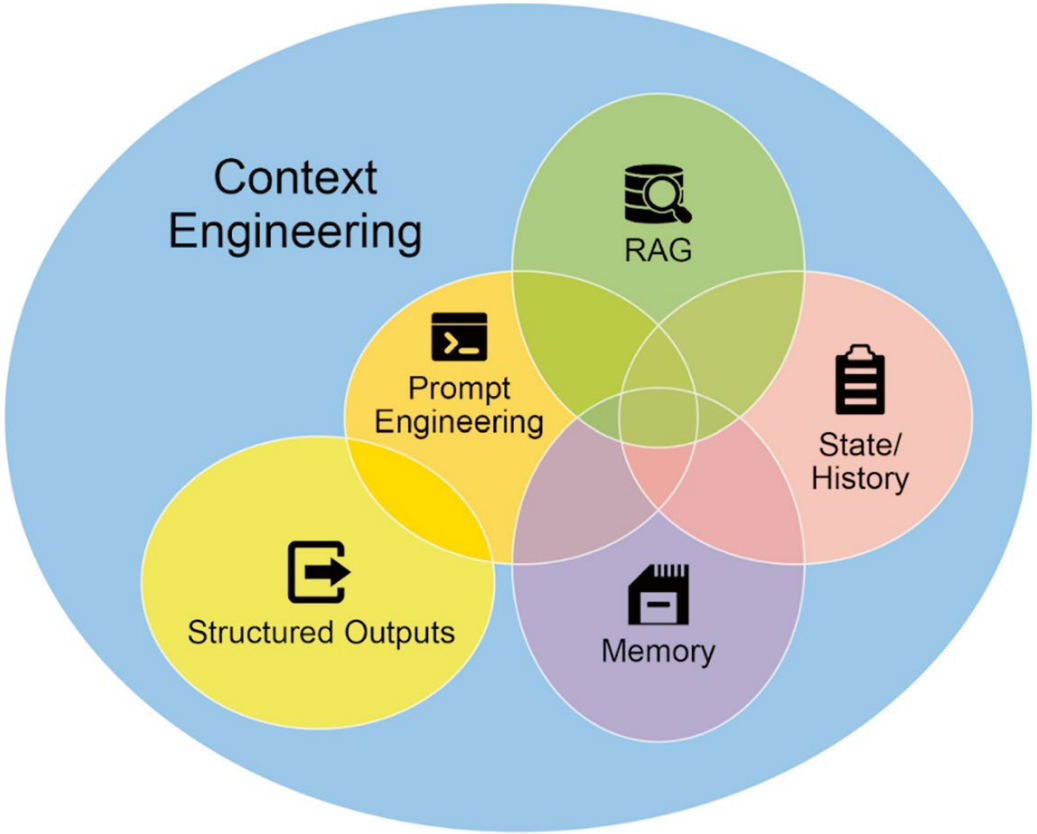
Different Data Sources/Types

Data Acquisition
Data Extraction
Data Ingestion
Data Labeling

Development Data
Training Data
Validation Data
Test Data

Habibullah, K.M., Gay, G. and Horkoff, J., 2024, September. A Framework for Managing Quality Requirements for Machine Learning-Based Software Systems. In International Conference on the Quality of Information and Communications Technology (pp. 3-20).

From: Gullí, Antonio. *Agentic design patterns: a hands-on guide to building intelligent systems*. Springer Nature, 2025.



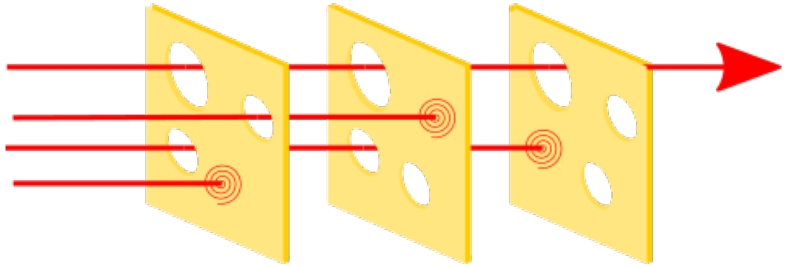
+ inter-agent communication (e.g., A2A)
+ all the data we had before

CHALMERS UNIVERSITY OF TECHNOLOGY | UNIVERSITY OF GOTHENBURG

Unwanted Behavior in Multi-Agent System

- Each individual agent may have improved predictability/reduced bias
- Better performance through chains of agents/models

https://en.wikipedia.org/wiki/Swiss_cheese_model



- Now uncertainty at the system level
- New emphasis on Guardrails
- Traditionally requirements about what to do, not what not to do
  - ISO IEEE 29148 2018: "Use positive statements and avoid negative requirements such as 'shall not'."
- ML Systems: not possible to elicit upfront everything the system should do
- Agentic Systems: also not possible to elicit upfront all the things the system should not do
  - Have to rely on iteration and human in the loop

CHALMERS UNIVERSITY OF TECHNOLOGY | UNIVERSITY OF GOTHENBURG

Rapid Evolution/Prototyping

A Comprehensive Survey of Self-Evolving AI Agents

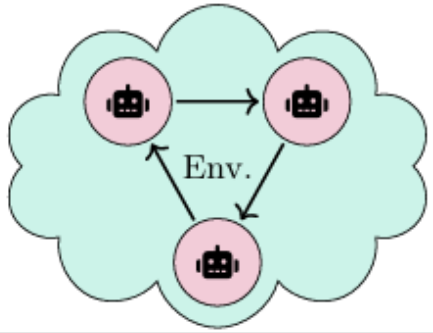
A New Paradigm Bridging Foundation Models and Lifelong Agentic Systems

Jinyuan Fang[*1], Yanwen Peng[*2], Xi Zhang[*1], Yingxu Wang[3], Xinhao Yi[1], Guibin Zhang[4], Yi Xu[5], Bin Wu[6], Siwei Liu[7], Zihao Li[1], Zhaochun Ren[8], Nikos Aletras[2], Xi Wang[2], Han Zhou[5], Zaiqiao Meng[1]✉

Multi-Agent Self-Evolving (MASE)

Agents ⇔ Environment
(signals from env.)

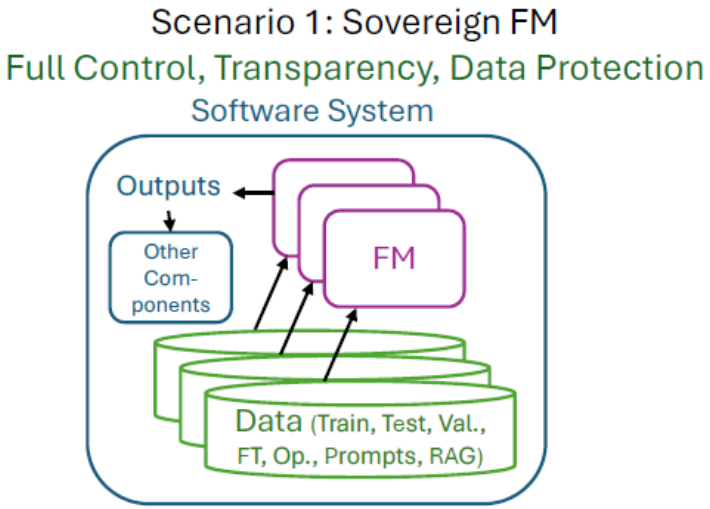
- Behaviour Optimisation
- Prompt Optimisation
- Memory Optimisation
- Tool Optimisation
- Agentic Workflow Optimisation



- Requirements (captured in prompts, guardrails, memory, etc.) are potentially static inputs to dynamic systems
- Requirements are part of the system
- Drive and constrain evolution
- System can dynamically try to improve its own requirements/purpose but there must be a human need/foundation for the system
  - Suggestions only – human in the loop

CHALMERS UNIVERSITY OF TECHNOLOGY | UNIVERSITY OF GOTHENBURG

New/More important Qualities?



- Control
- Transparency (of sources/training)
- Data protection (privacy)
- Sovereignty
- Cost (tokens, other resources)
- Sustainability (again)

CHALMERS UNIVERSITY OF TECHNOLOGY | UNIVERSITY OF GOTHENBURG

Capturing Intentions

- Increased focus on “intention engineering”
- It is important to capture the intention behind requirements rather than the requirements itself
- The agents should know “why”, will do a better job, make better decisions

- Meanwhile, take a trip back in time with me…

Iterative, Interactive Analysis of Agent-Goal Models for Early Requirements Engineering

by

Jennifer Marie Horkoff

A thesis submitted in conformity with the requirements for the degree of Doctor of Philosophy

Department of Computer Science
University of Toronto

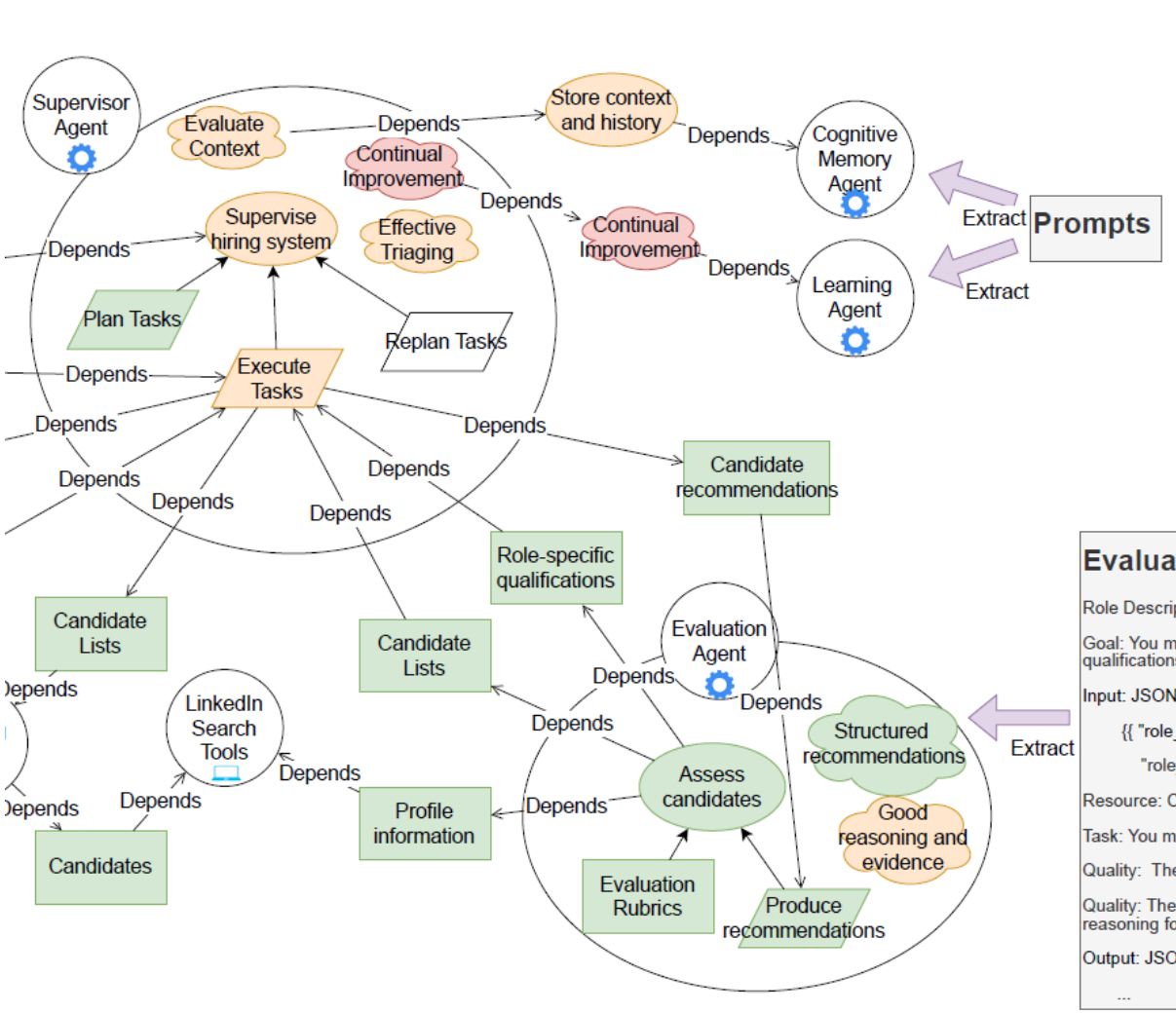
© Copyright by Jennifer Horkoff 2012

Opacity/Explainability: Monitoring and observation of Agentic AI Systems

Use requirements models to capture AI agents, their goals, and status

Example for hiring system:

Could lead to model-driven engineering for GenAI-based agentic systems



RE for Agentic AI – The way forward

RE for Agentic System Challenges	RE for Agentic Systems Solutions/Directions
New components/levels	➡ Examine new Agentic architecture patterns, identify components
Further different types of data	➡ Identify new data types and associated challenges
Multi-Agent Systems bring Mitigations and Challenges	➡ Guardrails, ways to specify more complete guardrails upfront, iterative specification
Continuous evolution	➡ Requirements as protected static artifact – human in the loop
Further new/more important NFRs	➡ Control, Transparency, Data Protection, Sovereignty, Cost
Capturing requirements and intentions	➡ Intentions and requirements as part of agentic system design
Observing and monitoring of GenAI-based agentic system	➡ Agentic Modeling Techniques, model analysis, goal modeling

More?

UNIVERSITY OF GOTHENBURG
CHALMERS UNIVERSITY OF TECHNOLOGY
AI Engineering Lab



Aim: discuss and collaborate on topics and issues related to the use of AI in systems development

Topics: Software Engineering for AI, AI for Software Engineering, AI and Robotic Applications, AI for Automotive, AI for Healthcare, Generative AI in Practice, AI Regulations and Guidelines, Human Aspects in AI, AI Safety Engineering, AI and Sustainability, etc.

Want to give a talk? Collaborate? Please reach out!

https://www.chalmers.se/en/projects/ai-engineering-lab/

ai-engineering-lab.cse@chalmers.se

CAIN 2027

6TH INTERNATIONAL CONFERENCE ON AI ENGINEERING

Software Engineering for AI-Enabled Systems

April 25–26, 2027

Dublin, Ireland

Co-located with ICSE 2027

Important Dates

30 Oct 2026
Research Track Paper Submission

6 Nov 2026
Doctoral Symposium, Posters, and Journal-First Track

Date TBD
Industry Track

Discussion Questions: Agentic Systems in or for Software

- Do you have agents (or LLMs/VLMs) as a component in your systems?
- If not, you will likely be using agentic systems to generate all or part of your system
- For the agentic systems you use or will use:
  - How do you know what this agentic system should do? Not do?
  - Where is this information stored? What format?
  - Do you capture intentions? Are they universally known?
  - What qualities are important for these systems?
  - What data is needed as input? Storage? Context? Memory?
  - How are the agentic systems updated/evolving?
    - Do they self-update? With humans-in-the-loop?
    - What kind of feedback loops do you envision?