Software Engineering Methodologies for Trustworthy Artificial Intelligence

ZAHOOR UL ISLAM, Umeå University, Sweden
Assignment for Software Engineering Course Module 2

August 2023

1 Introduction

The rapid development of Artificial Intelligence (AI) technology has led to significant economic and social benefits for many stakeholders, including companies, industries, and government organisations. New AI applications are being developed frequently across various domains, such as healthcare, automotive, finance, security, and entertainment. The increasing complexity and potential impact of AI applications requires a structured approach to requirements and system engineering to ensure that design and development activities are carried out in a responsible and trustworthy manner [1]. Ensuring robust, reliable and trustworthy system engineering is the foundation of most ethical guidelines and governance strategies published in the last few years and endorsed by governments and standard-making organisations worldwide [2, 3].

Trustworthy AI engineering involves designing and developing applications considering society's social, ethical, legal, and economic aspects, while involving several stakeholders throughout the AI development lifecycle. Practical approaches to operationalizing, integrating and incorporating ethical guidelines into the lifecycle of AI system engineering are essential. Different areas have proposed several solutions, with the SE area proposing a promising solution. SE provides a well-established set of development lifecycle activities across numerous domains with successful outcomes [4–7]. The AI lifecycle must consider ethical guidelines while following standard SE principles, such as *analysis, design, implementation, testing, deployment, and maintenance.* Additionally, the process of developing trustworthy AI must consider the maturity of various process areas. These areas include *configuration management, risk management, project planning, project monitoring and control, integrated project management, quantitative project management, and process and product quality assurance*. By focusing on SE areas, researchers can create AI systems that provide benefits and align with ethical values, promoting public trust and responsible innovation.

The European Commission's ethical framework outlines the dimensions that ensure the responsible development of AI systems, considering their impact on individuals, society, and the environment [8, 9]. These dimensions include Human Agency and Oversight, which ensure that individuals retain control and agency over AI systems. Technical Robustness and Safety ensure that AI systems are secure, reliable, and free from unintended harm. Privacy and Data Governance protects personal data and privacy through appropriate data collection, use, and management practices. Transparency provides clear and accessible information about AI systems' design, operation, and decision-making processes. Diversity, Non-discrimination, and Fairness promote diversity and ensure that AI systems do not discriminate against individuals or groups. Social and Environmental Well-being considers AI systems' broader social and environmental impacts. Accountability holds AI systems and their developers accountable for the outcomes and decisions these systems generate. I am working on creating a framework that incorporates ethical values into the entire development process of AI systems [10].

2 Two concepts/ideas from Robert's lecture:

2.1 Software Development lifecycle

Software development Life Cycle (SDLC) aims to provide an effective process for systematically developing software systems. It is crucial to develop AI software by breaking the task into smaller, more manageable activities, and then identifying which activities are relevant at each stage. This will help practitioners identify trustworthy requirements and systematically implement them.

The SDLC and Trustworthy Machine Learning (ML) engineering are different but interconnected concepts. The SDLC provides a structured method for developing software applications and systems, while Trustworthy ML focuses on developing reliable, explainable, fair, privacy-preserving, and robust ML models [11].

2.1.1 Here's how they might connect:

- Requirements Gathering and Analysis: At this stage of the SDLC, the objective is to understand the user's needs and system requirements. For a ML system, this means understanding what predictions the system should make, what data will be available, and what level of accuracy is expected. Trustworthy ML considerations include the ethical and legal use of data, fairness of outcomes, and the privacy of individuals whose data is used.

- Design: In this stage, system specifications are prepared, which help define the system architecture. In a ML context, this could involve selecting appropriate ML algorithms, planning how to preprocess the data, and deciding how to validate the models.

- Implementation: Here, the system is developed or coded. In a ML context, this could involve training and testing ML models. Trustworthiness considerations include ensuring that the model training process is transparent and reproducible.

- Testing: The software is tested to meet the specified requirements in this phase. In a ML context, this would involve evaluating the model's performance on a held-out test set. Trustworthiness considerations include checking that the model performs well for all population subgroups, not just on average, and has no unintended bias.

- Deployment: The software is released for users. In a ML context, the model is deployed to make predictions in a real-world setting. Trustworthiness considerations include monitoring the model's performance over time, as the performance can degrade if the distribution of the data changes.

- Maintenance: The software is updated as needed to ensure it continues to function correctly. In a ML context, models must be retrained if their performance degrades. Trustworthiness considerations include ensuring that updates to the model preserve the properties of fairness, privacy, robustness, and explainability.

2.2 Software Designing

Software designing also plays a crucial role in SDLC. Software designing can be related to abstraction, patterns, separation of data, modulating, functional independence, and refactoring. Several methodologies and processes are designed, developed and proposed for developing AI systems based on different approaches, mainly extending existing object-oriented, behaviour-based designs and knowledge engineering methodologies. AI design technologies have been significantly redesigned, changed and updated over the years. AI system engineering design decisions must fit within Socio-Technical Systems (STS) frameworks. The STS approach to system development leads to systems that are more acceptable to end users and deliver better value to stakeholders [12]. One of the reasons for this approach is that traditional methods might not consider the complete requirements for the complex structure of AI systems, where

the environment is multidimensional and heterogeneous [13]. Assessing the current design challenges and methods of AI systems is key to developing robust design methods for trustworthy AI systems [14].

2.3 Stakeholders

Another critical criterion in AI development methods is how well they support real-world users, user participation, and user involvement in the development life cycle [15, 16]. Methods to engineer trustworthy AI systems should influence people, groups and societies when making decisions. To clarify how AI development methods support the participation process, in which individuals, groups and organisations are consulted or actively participating in a project or activity program. Responsibility for compliance with ethical guidelines and the guarantee of accountable development of AI systems lies not only with system engineers, project and product managers, designers and developers, but also with regulators, auditors, owners and users of AI technology. They all need the tools, methods and processes to participate effectively in development activities. To maintain privacy and security, all relevant parties must adhere to trustworthiness guidelines set by various authorities, such as the European Union and the IEEE.

3 Two ideas from ONE of the guest lectures

3.1 AI for Software Engineering

AI in software engineering is revolutionising the way code is written, tested, and deployed, enhancing efficiency and reducing human error. By leveraging ML algorithms and data analytics, AI tools can automate routine tasks, code reviews, and even predictive maintenance, transforming the SDLC [17, 18].

- Requirement Analysis Gathering requirements is a critical step in the software development process, including projects that aim to develop trustworthy ML systems. ML can be used to support and automate parts of the requirements-gathering process, such as:
- Analysing Stakeholder Input: If there is a large volume of stakeholder input, such as user feedback, emails, forum posts, etc., text mining and natural language processing (NLP) techniques can be used to automatically analyse this data and extract common themes. This can help identify requirements related to trustworthiness, such as privacy, fairness, or transparency.
- Sentiment Analysis: Sentiment analysis, a type of NLP, can be used to gauge stakeholder sentiment about certain aspects of a system. This could help identify areas where trust is lacking and needs to be addressed in the requirements.
- Survey Analysis: If surveys are used to gather input from stakeholders, ML can be used to analyse the responses. For example, clustering algorithms could be used to identify groups of stakeholders with similar views on trustworthiness issues.
- Requirement Prioritization: ML techniques can be used to prioritise requirements based on various factors, such as the frequency with which stakeholders mention a requirement, or the predicted impact of the requirement on system trustworthiness.
- Anomaly Detection: ML algorithms can be used to identify anomalies or outliers in the data gathered during the requirements analysis process. This could include identifying stakeholders whose views on trustworthiness are significantly different from the majority, or identifying requirements that are contradictory or inconsistent.

Remember, while ML can help automate parts of the requirements gathering process, human involvement is still essential.

3.2 Documentation

Technical documentation is a critical part of any software development project, including those involving ML [19, 20]. Proper documentation allows users and other developers to understand what a piece of software does, how it works, and how to use or modify it. Trustworthy ML techniques can be incorporated into technical documentation in several ways to ensure the project is transparent, understandable, and reliable [21].

- Data Documentation: Explain what data was used to train the model. This should include the source of the data, how it was collected, any preprocessing steps, and any known limitations or potential biases in the data. Document the reasoning behind the selection of data features used in model training, and their relevance to the prediction task.
- Model Architecture and Training: Document the ML algorithms used, including a description of how they work and why they were chosen. Describe how the model was trained, including what metrics were used to evaluate its performance, and any techniques used to prevent overfitting. Also, document the hyper-parameters used in the model and any tuning process.
- Performance Metrics: Provide comprehensive reporting of the model's performance, including accuracy, precision, recall, F1 score, ROC AUC, etc. Make sure to report these metrics for different subgroups of the data to check for fairness and identify any disparate impacts.
- Interpretability and Explainability: If explainability techniques have been used (e.g., SHAP, LIME), document how these techniques work and what insights they provide about the model's predictions. If the model is interpretable (e.g., a decision tree), provide documentation explaining how to interpret its structure.
- Privacy and Security Measures: Document any measures taken to protect the privacy of individuals' data, such as differential privacy or federated learning. Also document any robustness measures taken to protect the model from adversarial attacks.
- Version Control and Experiment Tracking: Document all versions of the data, code, and model, including what changes were made in each version and why. If an experiment tracking tool was used, provide documentation on how to access and interpret this information.
- Instructions for Use and Maintenance: Provide clear instructions on how to use the model, including any software dependencies, how to input data, and what the model's outputs mean. Also, provide guidance on when and how the model should be retrained or updated.

Including these elements in your technical documentation will ensure that ML applications are transparent.

4 Two topics from the given list:

4.1 Requirement Engineering for Trustworthy Artificial Intelligence

Software requirements engineering defines and documents requirements in the design process. It's crucial to software development, as it fulfils user and stakeholder needs. A few key considerations need to be considered when integrating Trustworthy ML into this process [22].

- Identifying the Requirements: This involves understanding the user's needs and the problem the system is designed to solve. For Trustworthy ML, this should also include specifications about model performance (accuracy, precision, recall, etc.) as well as ethical requirements such as fairness, transparency, privacy, and robustness. For example, if the system uses personally identifiable information, privacy could be a requirement. Or, if the system makes decisions that affect people's lives, such as loan approval, fairness could be a requirement.

- Requirements Specification: The requirements identified are then documented in a clear, precise, and unambiguous manner. For Trustworthy ML, this could involve defining what fairness or privacy means in the context of the particular system, and specifying how these properties will be measured or validated.
- Requirements Validation: This involves checking that the requirements are consistent, complete, and resolvable. For Trustworthy ML, this could mean that ethical requirements don't conflict, all relevant ethical concerns have been considered, and the requirements can be met given available data and technology.
- Requirements Management: This is about handling changes to the requirements as the project progresses. For Trustworthy ML, this could involve updating requirements as the team learns more about the data or task, or as external conditions change, such as regulatory or societal norms around algorithmic fairness or privacy.

Integrating trustworthy ML principles into the Software Requirements Engineering process will lead to ethically aligned AI systems.

4.2 Regulation and Compliance

The European Union (EU) has been proactive in establishing guidelines and regulations for AI systems. Two of the most relevant regulations to Trustworthy AI are the General Data Protection Regulation (GDPR) and the proposed Artificial Intelligence Act [23, 24].

4.2.1 General Data Protection Regulation (GDPR):

This regulation came into force in 2018, and focuses primarily on data privacy and individuals' rights over their data [25]. It has a significant impact on the development and deployment of ML models in several ways:

- Right to explanation: Users have the right to know how decisions affect them. This means ML models must be interpretable or explainable.
- Data minimisation and purpose limitation: You should only collect necessary and relevant data for your task, which can impact the type and quantity of data used to train ML models.

Data subject's rights, including the right to access, rectification, and erasure, can impact ML models, especially when a user invokes the right to be forgotten, requiring their data to be removed from datasets used to train models.

4.2.2 Artificial Intelligence Act:

Proposed by the European Commission in 2021, this regulation aims to create a legal framework for safe and trustworthy AI (including ML) [26].

- High-risk AI systems: The Act proposes stricter regulations for high-risk AI systems, including many that use ML. These systems must undergo a conformity assessment before being put on the market.
- Transparency obligations: Users should be aware when interacting with an AI system, and have the right to know the logic, significance, and consequences of processing their data by the AI system.
- Quality of datasets: The proposal emphasises the need for high-quality datasets free from biases to train high-risk AI systems.

Both these regulations emphasise the principles of trustworthy AI, such as fairness, explainability, privacy, and robustness.

References

[1] V. Dignum, *Responsible artificial intelligence: how to develop and use AI in a responsible way*. Springer, 2019, vol. 2156.

[2] E. A. Design, "A vision for prioritizing human well-being with autonomous and intelligent systems," *Report (The IEEE Global Initiative for Ethical Considerations in Artificial Intelligence and Autonomous Systems, 2018)*, 2019.

[3] R. Chatila, K. Firth-Butterfield, and J. C. Havens, "Ethically aligned design: A vision for prioritizing human well-being with autonomous and intelligent systems version 2," *University of southern California Los Angeles*, 2018.

[4] R. S. Pressman, *Software engineering: a practitioner's approach*. Palgrave macmillan, 2005.

[5] Y. Sedelmaier and D. Landes, "Software engineering body of skills (swebos)," in *2014 IEEE Global Engineering Education Conference (EDUCON)*. IEEE, 2014, pp. 395–401.

[6] A. Abran, J. W. Moore, P. Bourque, R. Dupuis, and L. Tripp, "Software engineering body of knowledge," *IEEE Computer Society, Angela Burgess*, vol. 25, 2004.

[7] J. McManus, "A stakeholder perspective within software engineering projects," in *2004 IEEE International Engineering Management Conference (IEEE Cat. No. 04CH37574)*, vol. 2. IEEE, 2004, pp. 880–884.

[8] G. Kilian, "White paper on artificial intelligence-a european approach to excellence and trust," 2020.

[9] A. HLEG, "Assessment list for trustworthy artificial intelligence (altai) for self-assessment," *High Level Expert Group on Artificial Intelligence. B-1049 Brussels*, 2020.

[10] Z. U. Islam, "Software engineering methods for responsible artificial intelligence," in *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems*, 2021, pp. 1814–1815.

[11] S. Martínez-Fernández, X. Franch, A. Jedlitschka, M. Oriol, and A. Trendowicz, "Developing and operating artificial intelligence models in trustworthy autonomous systems," in *International Conference on Research Challenges in Information Science*. Springer, 2021, pp. 221–229.

[12] G. Baxter and I. Sommerville, "Socio-technical systems: From design methods to systems engineering," *Interacting with computers*, vol. 23, no. 1, pp. 4–17, 2011.

[13] H. Aldewereld, V. Dignum, and Y.-h. Tan, "Design for values in software development," *Handbook of Ethics, Values, and Technological Design: Sources, Theory, Values and Application Domains*, pp. 831–845, 2015.

[14] S. Umbrello and I. Van de Poel, "Mapping value sensitive design onto ai for social good principles," *AI and Ethics*, vol. 1, no. 3, pp. 283–296, 2021.

[15] B. Li, P. Qi, B. Liu, S. Di, J. Liu, J. Pei, J. Yi, and B. Zhou, "Trustworthy ai: From principles to practices," *ACM Computing Surveys*, vol. 55, no. 9, pp. 1–46, 2023.

[16] C. González-Gonzalo, E. F. Thee, C. C. Klaver, A. Y. Lee, R. O. Schlingemann, A. Tufail, F. Verbraak, and C. I. Sánchez, "Trustworthy ai: closing the gap between development and integration of ai systems in ophthalmic practice," *Progress in retinal and eye research*, vol. 90, p. 101034, 2022.

[17] M. Barenkamp, J. Rebstadt, and O. Thomas, "Applications of ai in classical software engineering," *AI Perspectives*, vol. 2, no. 1, p. 1, 2020.

[18] R. Feldt, F. G. de Oliveira Neto, and R. Torkar, "Ways of applying artificial intelligence in software engineering," in *Proceedings of the 6th International Workshop on Realizing Artificial Intelligence Synergies in Software Engineering*, 2018, pp. 35–41.

[19] J. Chang and C. Custis, "Understanding implementation challenges in machine learning documentation," in *Equity and Access in Algorithms, Mechanisms, and Optimization*, 2022, pp. 1–8.

[20] F. Königstorfer and S. Thalmann, "Software documentation is not enough! requirements for the documentation of ai," *Digital Policy, Regulation and Governance*, vol. 23, no. 5, pp. 475–488, 2021.

[21] D. Piorkowski, D. González, J. Richards, and S. Houde, "Towards evaluating and eliciting high-quality documentation for intelligent systems. arxiv [cs. se]," 2020.

[22] A. Serban, K. van der Blom, H. Hoos, and J. Visser, "Practices for engineering trustworthy machine learning applications," in *2021 IEEE/ACM 1st Workshop on AI engineering-software engineering for AI (WAIN)*. IEEE, 2021, pp. 97–100.

[23] M. S. Gal and O. Aviv, "The competitive effects of the gdpr," *Journal of Competition Law & Economics*, vol. 16, no. 3, pp. 349–391, 2020.

[24] M. Veale and F. Zuiderveen Borgesius, "Demystifying the draft eu artificial intelligence act—analysing the good, the bad, and the unclear elements of the proposed approach," *Computer Law Review International*, vol. 22, no. 4, pp. 97–112, 2021.

[25] F. Sovrano, F. Vitali, and M. Palmirani, "Modelling gdpr-compliant explanations for trustworthy ai," in *Electronic Government and the Information Systems Perspective: 9th International Conference, EGOVIS 2020, Bratislava, Slovakia, September 14–17, 2020, Proceedings 9*. Springer, 2020, pp. 219–233.

[26] H. AI, "High-level expert group on artificial intelligence," p. 6, 2019.