Fairness as a First-Class Requirement: A Fairness Hazard Analysis Approach to Socio-Technical Processes

Giovanna Broccia[1][0000−0002−4737−5761], Lucio Lelii[1], Roberto Cirillo[1], Dario Di Nucci[3], Samuel Fricker[4], Fabio Palomba[3], Giorgio O. Spagnolo[1], and Alessio Ferrari[1,2][0000−0002−0636−5663]

[1] ISTI-CNR, Pisa, Italy
{giovanna.broccia, lucio.lelii, roberto.cirillo, spagnolo}@isti.cnr.it
[2] University College Dublin, Dublin, Ireland, alessio.ferrari@ucd.ie
[3] University of Salerno, Salerno, Italy, {ddinucci, fpalomba}@unisa.it
[4] University of Applied Sciences and Arts Northwestern Switzerland, Windisch, Switzerland, samuel.fricker@fhnw.ch

Abstract. *Context and Motivation.* Fairness in socio-technical systems is increasingly recognised as a critical requirement, especially in processes involving human-AI interaction. Fairness hazards are situations or factors that threaten the fair treatment of individuals or groups. If left unaddressed, they can accumulate into systemic bias. Therefore, ensuring fairness must be treated as a first-class requirement during system design, rather than a post-hoc fix. *Question/Problem.* Systematic methods for identifying fairness hazards in socio-technical workflows and translating them into requirements-level mitigations are still missing. *Principal Ideas/Results.* We propose Fairness Hazard Analysis (FHA), an adaptation of hazard analysis methods from the safety-critical domain to analyse fairness in socio-technical processes. FHA is demonstrated through an AI-assisted hiring case and supported by *HumAInFlow*, a modelling and simulation platform. The approach is preliminarily evaluated through two focus groups. The feedback from participants highlights FHA's usefulness for structured fairness analysis, the importance of diverse expertise, and the potential for deeper integration within HumAInFlow. *Contribution.* This work offers a novel method for integrating fairness into requirements analysis of socio-technical workflows, and provides an LLM-based tool to automate the analysis, marking a shift from bias detection to bias prevention with *fairness-by-design*.

1 Introduction

As artificial intelligence (AI) technologies are increasingly deployed in everyday activities and mediate decisions that affect people's lives, the risks of biased or inequitable outcomes have become more visible and urgent. In socio-technical systems, where humans and AI agents interact, these risks can be amplified: biases may propagate across actors and processes, leading to compounding fairness issues over time [14].

Despite its relevance, fairness is often treated as a post-hoc evaluation concern rather than a requirement to be engineered from the outset [11]. Ensuring fairness, however, requires systematic attention comparable to safety and security, calling for requirements engineering (RE) methods that can identify and mitigate fairness risks early in the design process.

Recent work has introduced the notion of *fairness debt*, conceptualising fairness issues as liabilities that accumulate when unaddressed and become increasingly difficult and costly to resolve [24]. While research on algorithmic fairness has produced a wide range of metrics and mitigation techniques, most are typically applied at the AI model or dataset level. Consequently, there remains a lack of operational methods that requirements engineers can apply to analyse and address fairness in socio-technical workflows [23].

To address this gap, we propose Fairness Hazard Analysis (FHA), an adaptation of hazard analysis methods from safety engineering [9]. FHA treats fairness issues as hazard-like states that may emerge and propagate through human–AI workflows, enabling their systematic identification, analysis, and mitigation. This analogy is motivated by the conceptual parallel between safety and fairness: just as safety engineering aims to identify and control conditions that could lead to harm, fairness engineering can systematically anticipate and mitigate conditions that may cause inequitable outcomes. By treating fairness issues as hazard-like states that can propagate through human–AI workflows, FHA provides a structured way to trace and control the accumulation of fairness debt [24] before it leads to systemic bias.

To illustrate the approach, we applied FHA to an AI-assisted hiring process. We also introduce a tool named HumAInFlow for modelling, simulating and analysing socio-technical workflows and support FHA. The approach and tool have been qualitatively evaluated through two focus groups involving diverse experts, aimed at gathering feedback on its clarity and usefulness, and on the suitability of HumAInFlow as a supporting tool for conducting FHA. Major points of improvements are the need to frame fairness concepts within specific contexts, the enhancement of analytical rigour and simulation capabilities, and the strengthening of usability and interoperability of the tool.

It should be noted that both method and tool are at the *proof-of-concept* level, i.e., Technology Readiness Level (TRL) 3. This study is part of a larger design science [28] endeavour, where we have currently performed the phases of problem investigation, treatment design, and preliminary validation in a controlled environment (through the focus groups). These will be later followed by implementation, i.e., application in a real-world problem context, and evaluation, i.e., systematic assessment in practice.

Related work. Fairness in algorithmic and socio-technical systems has traditionally been addressed through metrics and mitigation techniques applied to data or models. While effective for local parity, these approaches often abstract away the organisational and human contexts in which decisions occur. Foundational critiques emphasise that fairness must be reasoned about at the system level: abstraction from social context can conceal structural inequities and reproduce

systemic harms [24,22,8]. Recent studies have begun to explore how fairness and broader human values can be operationalised throughout the system lifecycle. *Values@Runtime* proposes mechanisms to capture and adapt to stakeholder values during operation [3], while *ReFair* focuses on fairness-requirement elicitation in machine learning systems through a context-aware recommender system [12]. Empirical analyses further show that fairness is still treated as a secondary quality attribute: developers lack systematic, lifecycle-oriented methods to specify, trace, and maintain fairness requirements [20,27]. In parallel, safety and security engineering provide well-established hazard-analysis frameworks for early identification and mitigation of risks [16]. Recent work demonstrates that system-safety methods can also uncover social and ethical risks in machine-learning systems [21]. However, explicit translations of fairness risks into actionable, requirements-level controls across human–AI workflows remain scarce.

Our proposed analysis (i.e., FHA) extends these foundations by (i) systematically identifying fairness hazards using known sources of fairness debt, (ii) analysing their propagation through socio-technical workflows, and (iii) deriving requirements-level mitigations with explicit traceability to the unfair outcomes they are intended to prevent.

2 Background

2.1 Bias, Fairness, and Hazards

Bias, the systematic deviation from objective accuracy in judgment, often arises when data or decisions reflect unequal representation or pre-existing human prejudices [26]. In AI systems, bias frequently originates from the human-generated datasets used for training [13]. Fairness, in this context, involves mitigating such systematic errors to ensure that AI-supported outcomes do not perpetuate or amplify inequities across demographic or social groups [19].

However, recent evidence shows that fairness cannot be considered a static property of algorithms alone, as human–AI interactions can create feedback loops that dynamically shape and intensify biases in human cognition and behaviour [14]. These feedback effects represent significant hazards: risks that extend beyond technical malfunction to encompass psychological, social, and ethical consequences [6]. When biased AI systems influence human perception and judgment, they may not only distort individual decision-making but also reinforce societal disparities, making the identification and correction of these feedback-driven hazards a critical challenge for responsible design of AI systems [1].

2.2 Fairness Debt

Fairness debt was introduced to explain how fairness issues in software systems accumulate when they are not explicitly managed throughout the software lifecycle [24]. Aligned with the definitions of technical debt [2] and social debt [25],

fairness debt represents the latent socio-technical liabilities that result from fairness oversights, omissions, or trade-offs made during development and operation.

De Souza Santos et al. [24] identify several root causes of fairness debt across the software lifecycle: **(i) cognitive bias**, arising from developers' subjective assumptions; **(ii) requirements bias**, from incomplete or non-inclusive elicitation; **(iii) design bias**, introduced through architectural or interface choices; **(iv) historical bias**, stemming from legacy data that reproduces inequities; **(v) training bias**, due to unrepresentative datasets; **(vi) model bias**, produced by algorithmic simplifications or parameter settings; **(vii) testing bias**, when validation overlooks fairness metrics; and **(viii) societal bias**, reflecting broader structural inequalities in the system's context.

These causes are not isolated but interdependent, meaning that fairness issues can propagate across lifecycle stages: for example, an unaddressed requirements bias may evolve into design or testing bias downstream. Over time, the accumulation of such debts increases the risk of systemic inequities, reputational damage, and regulatory non-compliance. Although defined in the context of software development, several of these root causes — particularly cognitive, societal, and requirements bias — can also emerge within the human components of socio-technical systems. Human decision-makers interacting with software systems may, for instance, over-rely on algorithmic recommendations, apply subjective evaluation criteria, or reproduce social stereotypes. This socio-technical interpretation reinforces that fairness debt is not purely a software engineering concern but a property of the entire human–software ecosystem. Hence, it should be treated as a managed and traceable property of socio-technical systems, requiring continuous attention rather than post-hoc correction.

This work builds on this idea by using the identified root causes of fairness debt to structure the identification of fairness hazards in socio-technical workflows through the proposed FHA approach.

2.3 Hazard Analysis

Hazard analysis is a foundational concept in system safety engineering, aimed at identifying and mitigating conditions that could lead to undesired or unsafe system states [9]. A hazard is typically defined as a state or set of conditions that, together with certain triggers, can result in harm or loss [15]. The purpose of hazard analysis is to anticipate such conditions as early as possible, evaluate their causes and potential consequences, and design appropriate preventive or corrective controls [17].

Among the most commonly used hazard analysis techniques, the Preliminary Hazard Analysis (PHA) is a qualitative, top-down approach that provides an initial overview of potential hazards, even before detailed system design information is available [9]. Its objective is to capture early insights concerning potential risk sources, their likely causes and effects, and to propose preliminary mitigation strategies, which are documented in a hazard table. A PHA generally follows a structured sequence of activities. The process begins with system definition, followed by the identification of potential hazardous conditions, failures,

and actions. Each identified hazard is then examined to determine its possible causes and the severity and likelihood of its potential consequences. The combination of severity and likelihood provides a preliminary basis for assessing risk and prioritising hazards that require further attention. Finally, preventive or control measures are proposed to eliminate each hazard or reduce its associated risk to an acceptable level. The process is iterative: as the system design matures, new information can refine both the identified hazards and the proposed mitigations.

This work takes inspiration from PHA to conceptualise FHA—a socio-technical adaptation that treats fairness deficiencies as hazard-like conditions. FHA retains the PHA structure to identify, trace, and mitigate fairness hazards across socio-technical workflows.

3 Fairness Hazard Analysis

We adapt PHA to identify and mitigate fairness issues in FHA systematically. FHA treats fairness issues—e.g., biased decisions, unbalanced access to information, or unequal treatment of agents—as hazard-like conditions that can arise during socio-technical processes, enabling their structured analysis and mitigation at the requirements level.

Fig. 1: FHA process

Consistent with the structure of PHA, FHA follows the sequential process described below (cfr. Figure 1). Each step is carried out by a team of analysts with diverse expertise in RE, data science, ethics, software engineering, and the specific system domain. These experts are trained to recognise fairness issues and reason about their propagation across socio-technical processes, ensuring consistency in hazard identification, classification, and mitigation planning.

Step A. System Definition. As in PHA, the first step defines and models the socio-technical process, its actors, and their interactions.

Step B. Fairness Hazard Identification. Each actor within the process is examined by the analysts in terms of the fairness-debt root causes [24], systematically assessing whether and how any of these causes may give rise to fairness hazards within that part of the process. This step results in a fairness-hazard list documenting the potential fairness issues and where they could emerge.

Step C. Fairness Hazard Analysis. Each identified fairness hazard is analysed in terms of its consequences, propagation, impact (degree of unfair treatment), and likelihood (probability of occurrence) through a collaborative review process where analysts discuss and resolve differing judgments to reach consensus. Impact ranges from none (intentional or justified differentiation) to critical

(structural unfairness breaching ethical, legal, or organisational norms), while likelihood expresses how often a fairness issue may occur—from rare to systemic. Combining impact and likelihood allows analysts to prioritise fairness risks. FHA distinguishes between undesirable bias, which causes harm and requires mitigation, and justified or goal-aligned differentiation, which may be acceptable in context. The latter is recorded for transparency but assigned no impact or risk. This step produces a table reporting each hazard, its consequences, propagation, and qualitative risk classification.

Step D. Mitigation Planning. For each fairness hazard, FHA defines control actions at the requirements level that modify the socio-technical workflow to prevent, detect, or compensate for unfair outcomes. Mitigation is achieved by introducing or adjusting workflow nodes and by implementing specific controls at critical decision points. These controls may include procedural additions–such as inserting human review or consensus nodes for high-impact decisions–as well as technical interventions, for instance, refining or constraining AI behaviour through targeted prompt engineering, introducing fairness-aware scoring functions, or enforcing transparency and auditability checkpoints.

Step E. Iterative Refinement. As in traditional PHA, FHA is an iterative process. Once the socio-technical process evolves or new empirical evidence emerges, fairness hazards and mitigations are revisited.

4 Sample Case

AI-assisted hiring systems can enhance recruitment quality by improving efficiency and reducing the transactional workload of human personnel. Nevertheless, insufficiently investigated biases may lead to unfair practices and discriminatory outcomes based on factors such as gender, race, ethnicity, or personality traits [7]. We select the AI-assisted hiring process as a representative case because it involves complex and continuous human-AI interactions and decision-making steps that are particularly sensitive to fairness concerns, as largely demonstrated by previous literature on fairness engineering [10]. These characteristics make it an ideal context for illustrating how FHA can uncover and mitigate fairness hazards in socio-technical workflows. To conduct the analysis, a team of three analysts, who are authors of this paper, applied the FHA to the selected case.

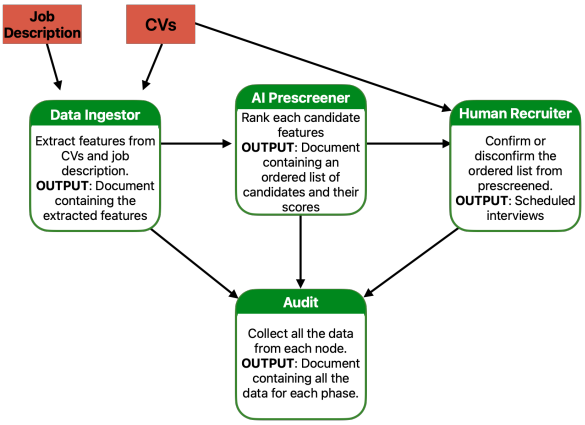

Fig. 2: AI-assisted hiring workflow

Step A. System Definition. The AI-assisted hiring process comprises four operational nodes (cf. Figure 2). The *Data Ingestor* receives the job description and candidates' curricula, and extracts structured features and job requirements. These are provided to the *AI Prescreener*, which produces a ranked shortlist of candidates based on feature–requirement matching. The *Human Recruiter* receives both the ranked shortlist and the original curricula, reviews them, and possibly overrides the AI's ranking to select interview candidates. All intermediate artefacts—the extracted features, AI shortlist, and recruiter decisions—are finally transmitted to the *Audit Node*, which consolidates the records into a persistent log for traceability.

Table 1: Identified Fairness Hazards (FH_i) across workflow nodes, with examples and detailed root causes.

ID	Node	Hazard Description	Example(s)	Root Cause(s)
FH_1	Data Ingestor	**Historical Bias.** CVs and job ads reflect past inequities (e.g., gender imbalance, unequal access to roles). The Ingestor encodes these disparities into structured data.	Career breaks due to maternity leave are interpreted as lower experience.	**Historical bias**; **Requirements bias**
FH_2	Data Ingestor	**Data Representativeness Deficit.** Parsing or extraction fails to capture diverse CV formats, languages, or trajectories, leading to incomplete representation.	Foreign degrees not recognised; Nonstandard CVs partially parsed.	**Training bias**; **Requirements bias**
FH_3	Data Ingestor	**Requirements Bias in Job Descriptions.** Subjective or exclusionary job terms are ingested without inclusivity checks.	"Young and dynamic" or "native English speaker" disadvantage certain groups.	**Requirements bias**; **Societal bias**
FH_4	AI Prescreener	**Training Data Imbalance.** The ranking model is trained on demographically skewed datasets, reinforcing dominant patterns.	Historical hires (mostly men) bias the model toward male-typical CVs.	**Training bias**; **Historical bias**
FH_5	AI Prescreener	**Proxy Feature Bias.** Certain features act as proxies for ranking.	Higher-ranked universities or institutions lead to higher candidate scores.	**Model bias**; **Training bias**
FH_6	AI Prescreener	**Transparency Deficit.** Lack of interpretability prevents auditors from identifying bias.	Recruiters receive rankings without explanations for scores.	**Design bias**; **Testing bias**
FH_7	Human Recruiter	**Confirmation Bias.** Recruiters overly rely on AI rankings, disregarding contradictory evidence.	Human recruiters interview only top-ranked candidates or highlight candidate weaknesses to justify a low score.	**Cognitive bias**; **Design bias**
FH_8	Human Recruiter	**Cognitive Bias.** Recruiters apply subjective heuristics or stereotypes during evaluation.	Foreign-sounding names rated as less suitable.	**Cognitive bias**; **Societal bias**

Step B. Fairness Hazard Identification. Each of the nodes defined in Step A is examined by the analysts for hazard identification. The identified fairness hazards (FH_i) are then jointly discussed and refined, resulting in the list provided in Table 1 and used in Step C.

Table 2: Fairness risk assessment for identified hazards.

ID	Potential consequences	Impact	Likelihood	Risk	Propagation
FH_1	Systemic exclusion of minority candidates; reputational damage.	High	Likely	High	Data Ingestor → AI Prescreener; Human Recruiter
FH_2	Misclassification/omission of nonstandard CVs; missed talent.	High	Likely	High	Data Ingestor → AI Prescreener; Human Recruiter
FH_3	Disadvantage for certain demographic/social groups.	Moderate	Possible	Medium – High	Data Ingestor → AI Prescreener; Human Recruiter; Audit
FH_4	Ranking bias; systematically lower scores for minorities.	High	Likely	High	AI Prescreener → Human Recruiter; Audit
FH_5	Higher-ranked institutions or companies receive higher scores, intentionally reflecting desired selection criteria.	None	Likely	None	AI Prescreener → Human Recruiter; Audit
FH_6	Inability to detect/contest biased rankings.	High	Possible	Medium – High	AI Prescreener → Human Recruiter; Audit
FH_7	Amplification of prescreening bias; reduced accountability.	High	Likely	High	Human Recruiter → Audit
FH_8	Inconsistent/unfair human assessments.	High	Likely	High	Human Recruiter → Audit

Step C. Fairness Hazard Analysis. The fairness hazards are analysed in terms of its consequences, propagation, impact, likelihood, and risk, as shown in Table 2. Some identified biases represent undesirable conditions that may lead to high-risk fairness issues and therefore require prompt mitigation (e.g., FH_1 or FH_4 can systematically disadvantage underrepresented candidates and propagate through multiple workflow nodes). Others, however, correspond to intended or contextually acceptable behaviours (e.g., preferential weighting of candidates from prestigious universities) and are consequently assigned no risk and not subject to mitigation.

Step D. Mitigation Planning. Table 3 summarises all the mitigation strategies defined to prevent, detect, or compensate for unfair outcomes, while Figure 3 shows the updated workflow. Some of the mitigations have a direct impact on the workflow by modifying or extending its execution (e.g., adding the *Feature-Validation* node to mitigate FH_2 or refining the prompt used in the *AI Prescreener* node to mitigate FH_1). Others act as informative or procedural controls, supporting awareness and organisational learning for the future (e.g., the detailed report on historical bias produced to mitigate FH_1, or the report on biased terms used in the job description issued by the *Requirements-Check* node to mitigate FH_3). As previously mentioned, some of the identified fairness hazards are not mitigated, as they represent expected behaviour (e.g., FH_5 is not mitigated because candidates from high-ranked universities are intentionally prioritised according to the desired selection criteria).

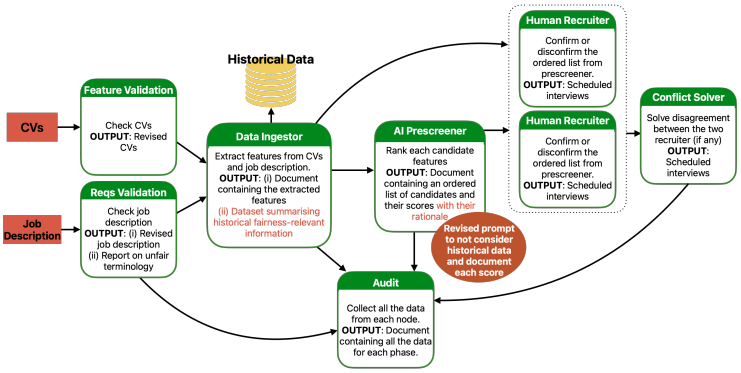

Fig. 3: AI-assisted hiring workflow with mitigation strategies

Table 3: Summary of mitigation actions for identified fairness hazards.

ID	Workflow modification / control	Expected effect
FH_1	Extend the *Data Ingestor Node* to produce an additional output dataset summarising historical fairness-relevant information.	Document historical bias for future hiring processes.
	Refine the prompt in the *AI Prescreener Node* to explicitly disregard historical imbalances during ranking.	Prevent replication of historical inequities in candidate scoring.
FH_2	Add a *Feature-Validation Node* between the user interface and the *Data Ingestor* to enforce a structured CV submission format.	Ensure completeness and comparability of candidate data; reduce representational bias due to unstructured or nonstandard CVs.
FH_3	Add a *Requirements-Check Node* at the input stage to automatically scan job descriptions for potentially biased or exclusionary terminology.	Detect and neutralise linguistic or cultural bias in job descriptions.
	The *Requirements-Check Node* generates a *Fairness Report* passed as input to the *Audit* Node to inform HR personnel about flagged terminology.	Support organisational awareness and long-term bias reduction in job descriptions.
FH_4	Retrain the *AI Prescreener* model using synthetic, demographically balanced data to counteract skewed patterns in the original training set.	Reduce model bias during candidate ranking by ensuring that historical or demographic imbalances do not influence learned representations.
FH_6	Implement prompt engineering rules that explicitly instruct the *AI Prescreener* to document the rationale behind each candidate ranking.	Improve transparency, interpretability, and contestability of AI decisions.
FH_7 and FH_8	Introduce a second *Human Recruiter Node* operating in parallel with the first, to independently review the AI-generated shortlist.	Mitigate overreliance on AI outputs and subjective judgments.
	Add a *Disagreement Discussion Node* to consolidate and compare the evaluations of the two recruiters, simulating a consensus phase when discrepancies occur.	Resolve divergences between human reviewers.

Step E. Iterative Refinement. Once all the mitigation actions have been performed, the new workflow is analysed from Step B.

5 HumAInFlow

HumAInFlow is a no-code, agentic platform, that we designed to model, simulate, and analyse socio-technical workflows where humans and software, including AI agents, co-exist and collaborate [5]. The platform is planned for

open-source release following completion of validation and testing phases. Unlike existing agentic AI frameworks (e.g., Langflow, Flowise AI, AutoGen Studio), HumAInFlow explicitly represents human roles as first-class nodes and allows their simulation through large language models (LLMs)—by instantiating personas through embedded prompts—making it suitable for studying processes that combine automated reasoning with human judgment.

We used HumAInFlow as a supporting tool for the AI-assisted hiring analysis, modelling and simulating all workflow nodes and mitigation strategies through LLMs (Figure 4 shows the original, non-mitigated, workflow).

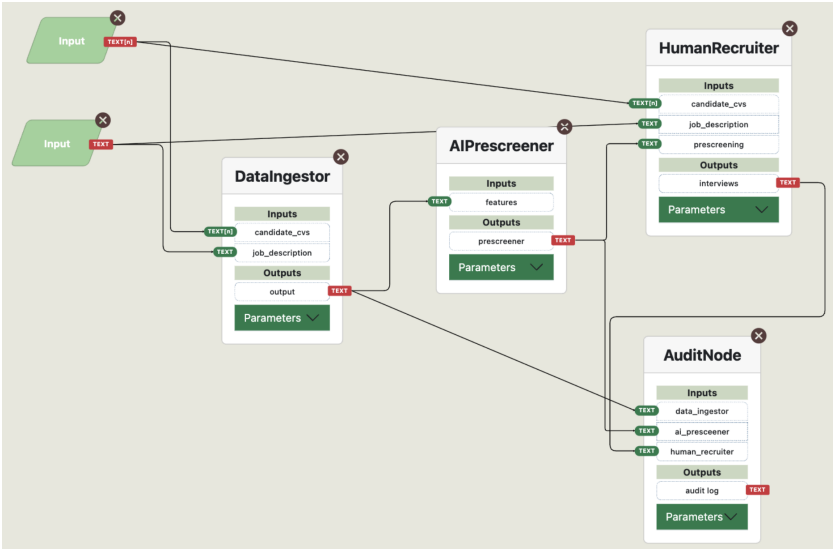

Fig. 4: AI-assisted hiring workflow modelled in HumAInFlow

This setup enables end-to-end analysis of socio-technical interactions under controlled and reproducible conditions. Within FHA, HumAInFlow supports both the identification and assessment of fairness hazards by allowing analysts to model workflow components—human and technical—as autonomous agents whose behaviour can be systematically varied. For instance, intentionally biased or imperfect nodes (e.g., a human recruiter affected by confirmation bias or an AI prescreener overvaluing specific features) can be introduced to observe how their behaviour influences downstream decisions. This allows analysts to systematically examine how fairness hazard can emerge and propagate. Furthermore, mitigation strategies can be modelled and simulated within the same environment to evaluate their effectiveness in reducing or eliminating bias.

For simulation purposes, the model provides a simplified abstraction of reality. For instance, human discussions or model retraining are represented by additional simulation nodes or prompt engineering mechanisms. To ensure a realistic representation of each actor in the AI-assisted hiring process, a careful prompt engineering phase was conducted to design prompts that could repro-

duce the expected behaviour of both human and technical entities. Standard prompting techniques—such as chain-of-thought reasoning and persona-based design—were applied in accordance with the OpenAI guidelines [18]. Since fairness hazards may also emerge from the wording or framing of prompts, three authors of this paper independently reviewed and iteratively refined all node prompts, selecting the formulations that best balanced realism and neutrality.

6 Preliminary Evaluation

At this stage, the FHA method and HumAInFlow are at the proof of concept level. Given their preliminary nature, we did not evaluate them across different scenarios, but we performed a first *treatment validation* with users in a controlled environment—following design science terminology and concepts—to identify relevant points of improvements before actual *implementation*, i.e., introduction of the artifact in practice. We validated the approach in two two-hour focus groups with 6 people each, involving twelve academics (33.4% Female, 66.7% Male) with different degrees of expertise in fairness (58% Basic or None, 42% Intermediate to Advanced). The focus groups were moderated by the first and last authors. They included a presentation of the approach and a video showing the capability of HumAInFlow to model and simulate socio-technical workflows to support FHA. During the presentation, the participants could ask questions and provide observations. At the end of the presentation, a set of eight questions was posed to participants to trigger further reflection on FHA and HumAInFlow, concerning ease of use, usefulness, and recommendations for improvement—questions reported in the replication package due to space limitations. The focus groups were recorded and automatically transcribed. Then, the last author conducted a thematic analysis to identify points of improvements. The results of the thematic analysis about the FHA method are in Table 4, while Table 5 reports the themes related to improvement recommendations for the HumAInFlow tool.

Overall, the participants were positive about the method and the tool, specifying that: "The approach feels structured and clear, especially for those familiar with requirements engineering."; "It helps identify fairness issues throughout the process...not only in the AI component."; and that "The tool nicely complements the method...it translates the analysis steps into something operational."

They also provided several recommendations for improvements. Concerning the method (Table 4), participants emphasised the need to clarify and contextualise the concept of fairness by explicitly defining it within each analysis context and incorporating ethical frameworks that distinguish between acceptable and unacceptable biases. They suggested strengthening the detection and representation of bias by improving tool support, clarifying the distinction between human and algorithmic sources, and enabling exploration of hidden or emergent biases. Enhancing the analytical rigour and usability of FHA was also highlighted, calling for clearer guidance on risk evaluation, the inclusion of domain-specific templates, and the provision of practical tutorials or worked examples. In terms of mitigation, participants recommended integrating automated suggestions for fairness interventions, supporting human–AI collaboration to balance oversight,

and ensuring transparent documentation of mitigation rationales. They also encouraged positioning fairness as an ongoing, reflective practice by embedding feedback loops into the method and reframing bias identification as a constructive opportunity for learning and ethical growth.

Concerning the tool (Table 5), participants highlighted the importance of improving its overall usability and visualization to better manage complex socio-technical workflows. They recommended introducing automated layouting, hierarchical representations, and semi-automatic abstractions to enhance clarity and reduce visual clutter. Strengthening model validation and knowledge reuse was also considered essential, suggesting pre-execution checks, breakpoints for debugging, and mechanisms to recall and warn about known fairness hazards or mitigation patterns when reusing existing components. The group particularly valued the tool's simulation and analytical potential, encouraging the ability to model biased human or AI agents, as well as feedback loops, to explore how bias propagates and evolves over time. In terms of extensibility and interoperability, participants proposed a modular plugin architecture and textual export options to facilitate integration with external models and analytical tools. Finally, they recommended broadening the analytical scope of the tool to assess the side effects of fairness interventions on other system qualities and to consider differentiated impacts across multiple stakeholder groups.

7 Conclusion

This paper introduced Fairness Hazard Analysis (FHA), a structured approach for identifying, analysing, and mitigating fairness risks in socio-technical workflows. By adapting hazard analysis principles from safety engineering, FHA enables fairness-by-design through early, requirements-level reasoning rather than post-hoc evaluation. The AI-assisted hiring case and preliminary focus group evaluation demonstrated the method's feasibility and its value in promoting interdisciplinary reflection on fairness. Future work will extend FHA to larger, practitioner-led case studies and integrate automated support for bias detection and mitigation within the HumAInFlow platform, advancing fairness as a first-class non-functional requirement in socio-technical system design.

Acknowledgments. Research supported by the EU Project CODECS GA 101060179. The authors acknowledge the use of ChatGPT to refine the text.

Data Availability. We made our supplementary material available in [4].

References

1. Afreen, J., Mohaghegh, M., Doborjeh, M.: Systematic literature review on bias mitigation in generative ai. AI and Ethics pp. 1–53 (2025)
2. Alves, N.S., et al.: Identification and management of technical debt: A systematic mapping study. Information and Software Technology **70**, 100–121 (2016)
3. Bennaceur, A., Hassett, D., et al.: Values@runtime: An adaptive framework for operationalising values. In: ICSE – SEIS. pp. 175–179. IEEE (2023)
4. Broccia, G., et al.: Fairness as a first-class requirement: A fairness hazard analysis approach to socio-technical processes - supplementary material (Oct 2025). `https://doi.org/10.5281/zenodo.17472752`

5. Broccia, G., et al.: Humainflow : a no-code platform for modelling and simulating human-ai workflows. Tech. Rep. 011, ISTI-CNR (2025)
6. Chen, C., et al.: Ethical perspective on ai hazards to humans: A review. Medicine **102**(48), e36163 (2023)
7. Chen, Z.: Ethics and discrimination in artificial intelligence-enabled recruitment practices. Humanities and social sciences communications **10**(1), 1–12 (2023)
8. Dolata, M., Schwabe, G., Schwabe, D.: Fairness as a sociotechnical concept in information systems. Information Systems Journal **33**(4), 970–995 (2023)
9. Ericson, C.A., et al.: Hazard analysis techniques for system safety. John Wiley & Sons (2015)
10. Fabris, A., Messina, S., Silvello, G., Susto, G.A.: Algorithmic fairness datasets: the story so far. Data Mining and Knowledge Discovery **36**(6), 2074–2152 (2022)
11. Farahani, A., et al.: On adaptive fairness in software systems. In: Proc. of SEAMS. pp. 97–103. IEEE (2021)
12. Ferrara, C., et al.: Refair: Toward a context-aware recommender for fairness requirements engineering. In: Proc. of ICSE. IEEE (2024)
13. Gichoya, J.W., et al.: Ai pitfalls and what not to do: mitigating bias in ai. The British Journal of Radiology **96**(1150), 20230023 (2023)
14. Glickman, M., Sharot, T.: How human–ai feedback loops alter human perceptual, emotional and social judgements. Nature Human Behaviour **9**(2), 345–359 (2025)
15. Leveson, N.G.: Safeware: system safety and computers. ACM (1995)
16. Leveson, N.G.: Engineering a Safer World: Systems Thinking Applied to Safety. MIT Press, Cambridge, MA (2011)
17. Lutz, R.R.: Analyzing software requirements errors in safety-critical, embedded systems. In: Proc. of RE. pp. 126–133. IEEE (1993)
18. OpenAI: Openai cookbook: Examples and guides for using the openai api. `https://github.com/openai/openai-cookbook` (2025), accessed: 2025-10-15
19. Pagano, T.P., et al.: Bias and unfairness in machine learning models: a systematic review on datasets, tools, fairness metrics, and identification and mitigation methods. Big data and cognitive computing **7**(1), 15 (2023)
20. Palomba, F., Ferrara, C., Sellitto, G., De Lucia, A., Ferrucci, F.: Fairness-aware machine learning engineering: How far are we? ESE **29**(1), 9 (2024)
21. Rismani, S., et al.: Applying system-theoretic process analysis (stpa) to identify ethical and social risks in machine learning systems. In: Proc. of FAccT. pp. 2540–2553. ACM (2023)
22. Selbst, A.D., Boyd, D., et al.: Fairness and abstraction in sociotechnical systems. In: Proc. of FAT*. pp. 59–68. ACM (2019)
23. Soremekun, E., Papadakis, M., Cordy, M., Le Traon, Y.: Software fairness: An analysis and survey. ACM Computing Surveys (2022)
24. de Souza Santos, R., et al.: Software fairness debt: Building a research agenda for addressing bias in ai systems. ACM TOSEM **34**(5), 1–21 (2025)
25. Tamburri, D.A., et al.: Social debt in software engineering: insights from industry. Journal of Internet Services and Applications **6**(1), 10 (2015)
26. Varona, D., Suárez, J.L.: Discrimination, bias, fairness, and trustworthy ai. Applied Sciences **12**(12), 5826 (2022)
27. Voria, G., et al.: Fairness-aware practices from developers' perspective: A survey. Information and Software Technology **182**, 107710 (2025)
28. Wieringa, R.: Design science methodology for information systems and software engineering. Springer (2014)

Table 4: Recommendations for improving the Fairness Hazard Analysis method

Improvement Area	Recommendation	Rationale / Description	Exemplary Quote (Participant)
1. Clarify and Contextualize Fairness Concepts	Define fairness explicitly for each analysis context	The concept of fairness is inherently subjective; the method should require explicit ethical framing and domain-specific definitions.	"For me, the word 'fairness' itself is tricky. What's fair depends on perspective — fairness is inherently biased."
	Distinguish between acceptable and unacceptable bias	The tool could allow users to tag certain biases as "intended" or "undesired" to reflect context-dependent ethics.	"There are desired and undesired biases — for instance, preferring candidates from high-ranking universities might be intentional."
	Include ethical principle templates	Offer pre-defined ethical or fairness frameworks (e.g., distributive justice, equal opportunity) to guide consistent analysis.	"Every system should state openly which ethical principles it follows, so users know what definition of fairness applies."
2. Strengthen Bias Detection and Representation	Enhance bias-identification support in the tool	Add structured prompts, examples, and checklists for detecting common human and algorithmic biases.	"AI systems often perpetuate existing inequalities, like paying men more than women."
	Model both human and algorithmic biases distinctly	The method should clearly separate bias types and provide visualization of how they interact.	"You should distinguish between human and machine biases — and possibly even combine their strengths to reduce weaknesses."
	Support exploration of hidden biases	Include sensitivity analysis or simulation tools to uncover biases not explicitly known by analysts.	"But how do we detect biases we don't know about?"
3. Improve Analytical Rigor and Usability of FHA	Provide clearer guidance on risk evaluation	Develop scales or calibration aids for judging likelihood and impact to reduce subjectivity.	"Judging likelihood and impact is subjective — calibration is needed."
	Offer domain-specific templates or libraries	Create FHA templates for common socio-technical domains (e.g., hiring, healthcare) to ease application.	"It's important to start with frequent, well-known recruitment cases — that's where this model can bring the most value."
	Provide interactive tutorials or example analyses	Tutorials can make the structured steps of FHA easier to apply and interpret.	"The approach feels structured and clear, especially for those familiar with requirements engineering."
4. Enhance Fairness Mitigation and Iteration Support	Integrate mitigation strategy suggestions	When a hazard is identified, the tool could suggest potential mitigation actions (e.g., retraining models, adding review nodes).	"If we know a bias exists — for example, gender bias in historical data — we can retrain models or balance datasets to mitigate it."
	Promote human–AI collaboration mechanisms	Explicitly model roles for human oversight, such as review checkpoints or multi-human consensus steps.	"Use two human recruiters and a discussion node to reduce over-reliance on AI."
	Support documentation of mitigation rationale	Encourage users to record why certain actions were chosen, increasing transparency and accountability.	"The tool allows process simulation to uncover unexpected biases through analysis of outputs."
6. Support Reflective and Ongoing Fairness Practice	Encourage iterative, dialogic reflection	Build feedback mechanisms for revisiting fairness assumptions as systems evolve.	"Fairness itself must be contextually defined."
	Frame bias as a learning opportunity	Treat the discovery of bias as a positive step toward ethical improvement, not merely a flaw.	"We're all biased about what counts as bias! Some biases might align with ethical values or goals."

Table 5: Recommendations for HumAInFlow

Improvement Area	Recommendation	Rationale / Description	Exemplary Quote (Participant)
1. Usability & Visualization	Provide auto-layout and clearer node arrangement	Reduce visual clutter in complex workflows; support automatic layouting so links and dependencies remain readable as models grow.	"The interactions between nodes are tangled; we need a clearer layout."
	Add hierarchical views / macro-nodes	Allow grouping nodes into higher-level "macro-nodes" and switching between levels of granularity to manage complexity.	"It would help to group single nodes into a macro-node and get a higher-level view."
	Semi-automatic high-level abstractions	Offer semi-automated clustering/abstraction of related nodes to generate higher-level visualizations without extra modeling burden.	"High-level views could be auto-generated to avoid modeling every abstraction level by hand."
2. Model Validation & Knowledge Reuse	Pre-execution validation & breakpoints	Add preflight checks (missing links, invalid connections) and debugging breakpoints to pinpoint execution failures early.	"Does the tool signal when something does not make sense?"
	Warnings on risky patterns / loops	Detect problematic loops or ill-formed connections and guide users to resolve non-termination or structural errors.	"We discussed adding breakpoints to understand where the problem arises."
	Memory of known hazards & reuse guidance	When importing nodes/models, surface past analyses (known fairness risks, typical mitigations) and suggest checks by node type.	"When importing something, the tool could run an analysis and warn: you should add a mitigation here."
3. Simulation & Analytical Capabilities	Simulate biased agents and propagation	Let users simulate biased humans/LLMs to observe how bias propagates through the socio-technical workflow and where mitigations help.	"I want to simulate a biased human or AI and see how the bias propagates and whether mitigation works."
	Temporal/feedback-loop simulation	Support time-evolving scenarios and feedback loops to evaluate whether mitigations hold over repeated interactions.	"Consider simulating feedback loops to verify if mitigations survive in the long term."
4. Extensibility & Interoperability	Plugin architecture for models/nodes	Enable adding local/remote models and custom nodes via plugins so organizations can integrate proprietary or fine-tuned components.	"It would be nice to add plug-ins or new nodes not initially foreseen by the system."
	Textual export/import (JSON/XML)	Provide an editable textual representation for complex models to support versioning, reviews, and interoperability with other tools.	"Having a textual representation of the diagram helps manage complex models."
5. Broader Analytical Scope	Assess side-effects on other NFRs	When planning fairness mitigations, analyze collateral impacts on other qualities (e.g., performance, usability, security).	"Mitigating fairness may affect other non-functional requirements; we should reason about side effects too."
	Multi-stakeholder impact weighting	Allow per-actor impact/risk weighting and trade-offs, since hazards may affect stakeholders differently.	"Risks can differ for recruiters vs. candidates; we should weight impacts and tailor mitigations."