LLMPathy: A Multi-Agent LLM Approach for Eliciting Inclusive Security Requirements

Shahid Waleed Bin, Naqvi Bilal, Afzal Hammad

This is a Author's accepted manuscript (AAM) version of a publication

published by IEEE

in 2025 IEEE 33rd International Requirements Engineering Conference Workshops (REW)

DOI: 10.1109/REW66121.2025.00037

Please cite the publication as follows:

Shahid, W. B., Naqvi, B., Afzal, H. (2025). LLMPathy: A Multi-Agent LLM Approach for Eliciting Inclusive Security Requirements. 2025 IEEE 33rd International Requirements Engineering Conference Workshops (REW), Valencia, Spain, 2025, pp. 243-248. DOI: 10.1109/REW66121.2025.00037

LLMPathy: A Multi-Agent LLM Approach for Eliciting Inclusive Security Requirements

Waleed Bin Shahid, Bilal Naqvi, and Hammad Afzal

***Abstract*—With a surge in cyber attacks targeted towards exploiting the human element of cybersecurity, it is vital to account for the diverse abilities and limitations of users. These limitations, whether temporary or permanent, physical or cognitive, acquired or congenital, vary in severity and deeply influence security related behaviour of users and give rise to a critical question: Should cognitively impaired users be held accountable if their actions result in a security breach? This paper presents a five-step approach that uses multi-agent large language models (LLMs) to identify inclusivity concerns and refine security requirements accordingly. The proposed approach (LLMPathy) is validated through a realistic use case, demonstrating its usability, effectiveness, and potential for adoption in real-world scenarios.**

***Index Terms*—cognitive disabilities, LLM, Inclusivity, security requirements, SRS**

I. Introduction

The nature of cyber attacks targeting human aspects of cybersecurity have seen a surge in recent years. The latest report by Verizon [1] revealed that 68% of analyzed security breaches in the last year alone involved human element including compromised credentials, stolen cards, phishing, among others. In many of these cases, the end user has some form of disability, which, according to the WHO, affects 16% of the world's population [2]. This makes it difficult for them to follow security best practices, increasing their risk of falling victim to cyber attacks. These disabilities can be related to cognition, mobility, hearing, vision, or other areas. Cognition-related disabilities, which include difficulties with memory, attention, problem-solving, or understanding information, are the most common [3].

Humans with cognitive disabilities struggle with memory, attention or understanding complex instructions which leads to difficulties in following cybersecurity practices like managing passwords, identifying phishing attempts or promptly responding to security alerts [4]. These challenges increase their vulnerability to cyber threats such as scams, malware attacks, and identity theft. A cybersecurity task that may seem hard but still manageable for an average neuro-typical user can become extremely challenging, or even impossible for some user with cognitive disability. This gives rise to an important question: as more services like healthcare, education, and banking move online, can we fairly place the blame on cognitively impaired users if a security breach happens because of their actions or lack of action? These cognitive disabilities can be divided into mild, moderate, severe, and profound levels, as each level affects how a person interacts with cybersecurity in different ways [5]. Table I describes these four levels of cognitive disability and how each level creates different challenges for cybersecurity.

TABLE I
Cognitive Disability Levels and their Impact on Cybersecurity

Level	**Description**	**Cybersecurity Challenges**
Mild	Some difficulty with memory, attention, or problem-solving	May forget complex passwords or struggle to understand security alerts and incidents
Moderate	Regular help required to perform daily tasks and understanding security instructions	Face challenges following security procedures like multi-factor authentication setup or software updates
Severe	Significant impairment in understanding and communication	Unable to identify suspicious activity, phishing emails and relies on others for safe internet use
Profound	Needs constant care due to very limited understanding and communication	Unable to manage any cybersecurity task and requires full supervision to avoid in security mishap

These challenges accentuate the need to address the needs of cognitively disabled people not just after digital solutions are deployed, but early in the software development life cycle. Unfortunately, there have been many gaps, and cognitive disabilities have not received enough attention during the design and development of cybersecurity features in everyday products and solutions. These gaps include a lack of clear policy directives that promote inclusive software development [6], a lack of awareness among security practitioners about the needs of cognitively impaired users, and an absence of cognitive requirement specifications from stakeholders [4]. This makes cognitively impaired users more vulnerable to cybersecurity attacks, as their needs and requirements are overlooked. This gives rise to a major question: how to gather the requirements for cognitively challenged users in the development of cybersecurity systems and services? The question becomes complicated when various levels of disabilities as mentioned in Table I are considered. For example, ADHD and Dyslexia are typically considered mild to moderate in severity and can cause challenges such as difficulty in managing passwords and understanding security instructions. Autism, which can range from mild to profound, may result in confusion when security messages and directives are unclear and increased stress or anxiety when security processes or functions change unexpectedly [7]. Another important question often considered by security practitioners is whether addressing all inclusivity requirements for cognitively impaired users might compromise the overall security of systems.

Based on the gaps and questions discussed, this paper proposes an approach using multi-agent Large Language Models (LLMs) to identify and refine requirements for users with different levels of cognitive disability, while still keeping systems, processes, and services secure from cyber attacks. The proposed approach first identifies inclusivity concerns in software requirements. Then it finds the right balance between security and inclusivity by eliciting trade-offs. In the next phase, it checks the quality of these inclusivity requirements using the ISO/IEC/IEEE 29148:2018 standard [8]. Finally, the requirements are updated to improve inclusivity without affecting security. To demonstrate the approach, the paper also presents a case study where a sample software requirement specification (SRS) document was fed to LLMPathy to identify inclusive concerns and update security requirements.

Research Contributions: The proposed work embeds neuro-diversity in software engineering practices through a novel multi-agent LLM approach that analyzes security requirements in the SRS, for cognitive inclusivity across varying severity levels of impairment, while ensuring human-in-the-loop design. Moreover, the approach holds implications for both research and practice by translating inclusive design principles into actionable, standards-aligned techniques applicable to real-world software systems.

The remainder of the paper is structured as follows. Section II presents the background and related work. Section III describes the approach for eliciting inclusive security requirements using LLMs. Section IV demonstrates the proposed approach using a realistic case study, and Section V concludes the paper and suggests future work.

II. Background and Related Work

Renaud, et al. [9] explored the role of accessibility in human-centered security and its impact on inclusive security technologies. They also identified the vulnerabilities caused by inaccessible security measures and also discuss the challenges in addressing them. Stelea, et al. [10] proposed the Holistic Security and Accessibility Layered Architecture (HSALA), a model that integrates accessibility and cybersecurity across web application development. With help of a theoretical case study, they demonstrated that HSALA outperforms traditional architectures in security, accessibility, scalability, and user experience. Santeri, [11] investigated the challenges users with disability face with software accessibility by recording user experiences and suggests improvement to the software development process to better accomodate disabled users. Aguiar, et al. [12] created AutismGuide which is a set of 69 recommendations for developing software applications tailored to individuals with Autism Spectrum Disorder (ASD).

Few research works have made use of Natural Language Processing (NLP) techniques to assess software requirements against various standards. Krishna, et al. [13] evaluated the ability of LLMs, specifically GPT-4 and CodeLlama, to automatically draft, validate, and improve Software Requirements Specification (SRS) documents and demonstrated that GPT-4 produces drafts which are more reliable as compared to CodeLlama. Marques, et al. [14] studied the integration of LLMs like ChatGPT-3.5 into software requirements engineering highlighting the ethical considerations along with benefits and challenges.

Villamizar, et al. [15] presented an approach that improves the security of web applications by using user stories and security details as input. It then applies NLP techniques to connect these stories to Open Web Application Security Project (OWASP) security requirements. Subahi, [16] presented a proof-of-concept approach using various machine learning techniques to automatically detect non-functional requirements and detect security related issues early in the software development process. With experiments he demonstrated that BERT-based models give promising results for improving software security and supporting sustainability. Aljedaani, et al. [17] presented an approach to help students learn how to make web applications more accessible to everyone. It used ChatGPT to assist students in addressing accessibility issues and found that using AI improves their understanding and ability to address accessibility issues. Tabarsi, et al. [18] found out that LLMs have significantly impacted the entire software development life cycle but also maintain that a careful review is still required due to occasional errors thereby urging the need for a human in the loop design.

The review of existing literature emphasizes the urgent need to address cognitive accessibility for a significant and under-served global population by involving both Large Language Models and a human-in-the-loop design. This aligns closely with the United Nations Sustainable Development Goals, specifically SDG 10 (Reduced Inequalities) and SDG 9 (Industry, Innovation, and Infrastructure).

III. Proposed Approach for Eliciting Inclusive Security Requirements

The proposed approach for eliciting security requirements for users with cognitive disabilities uses a five step approach as depicted in Fig. 1. It has been developed on the principles of Design Science Research [19] that focuses on development of IT systems with the intention of solving problems that serve human purpose.

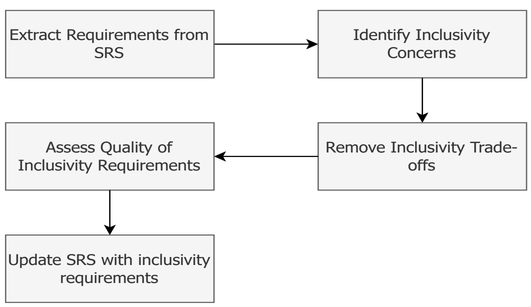

Fig. 1. Five step process of eliciting security requirements

The approach takes the Software Requirements Specification (SRS) document as input and extracts the relevant security requirements. These requirements are then forwarded to a multi-agent system powered by Large Language Models (LLMs), which independently identify inclusivity concerns for each of the four levels of cognitive disability. The key reason for addressing each level separately within the multi-agent system is that organizations may have varying needs and users with different levels of cognitive disability. The entire flow of LLMPathy is shown in Fig. 2.

A. Extracting Security Requirements from Software Requirements

LLMPathy starts by extracting security requirements $\boldsymbol{\mathcal{R}}_S$ from the SRS document or the product backlog, depending on whether traditional or agile methodologies are followed. Additional inclusivity requirements $\boldsymbol{\mathcal{R}}_D$ are gathered from dedicated surveys, interviews, and engagement with relevant stakeholders. These extracted requirements are collectively denoted as $\mathcal{R}^+$, where $\mathcal{R}^+$ = $\mathcal{R}_s$ + $\mathcal{R}_D$.

B. Generating Inclusivity Concerns via Multi-Agent LLMs

After extracting these requirements, a selected subset of LLMs within the multi-agent system, $\boldsymbol{\mathcal{M}}$, is activated by humans-in-the-loop $\boldsymbol{\mathcal{H}}$, based on the targeted cognitive disability categories. These agents analyze each requirement, $\mathcal{R}_j \in \mathcal{R}^+$, and generate associated inclusivity concerns, labeled by both the source requirement and the cognitive domain.

Let $\mathcal{R}^+ = \mathcal{R}_S \cup \mathcal{R}_D$ denote the complete set of requirements.

Let $\mathcal{I} \subseteq \{1, 2, 3, 4\}$ be the indices of LLM agents corresponding to different severity levels of cognitive disability (e.g., from mild to profound).

Then, the agent-specific concern generation is defined as:

$$\forall\, \mathcal{R}_j \in \mathcal{R}^+ \tag{1}$$

$$\forall i \in \mathcal{I}: \quad \mathcal{M}_i(\mathcal{R}_j) \xrightarrow{\mathcal{H}} \mathcal{C}_{j,i} \tag{2}$$

Each $\mathcal{C}_{j,i}$ represents an inclusivity concern raised by agent $\mathcal{M}_i$ for requirement $\mathcal{R}_j$.

The complete set of aggregated concerns is then:

$$\mathcal{C} = \left\{ \mathcal{C}_{j,i} \,\middle|\, \mathcal{R}_j \in \mathcal{R}^+,\, i \in \mathcal{I} \right\} \tag{3}$$

C. Elicitation of Inclusivity Tradeoffs

Once inclusivity concerns $\boldsymbol{\mathcal{C}}$, associated with the corresponding requirements $\mathcal{R}^+$ have been identified by the multi-agent LLM system, the next step involves contextually re-balancing these requirements to accommodate the identified inclusivity concerns. This stage is referred to as the elicitation of trade-offs, and is denoted by $\mathbb{T}$.

Algorithm 1 Eliciting Inclusivity Tradeoffs

procedure $\mathbb{T}(\mathcal{R}^+, \mathcal{C})$
  $\mathcal{R}^* \leftarrow [\,]$
  for all $\mathcal{R}_j \in \mathcal{R}^+$ **do**
    $\mathbb{T}_j \leftarrow \text{elicit}(\mathcal{R}_j, \mathcal{C}_j, \text{standards})$
    $\mathcal{R}^*_j \leftarrow \text{synthesize}(\mathcal{R}_j, \mathbb{T}_j)$
    $\mathcal{R}^*.\text{append}(\mathcal{R}^*_j)$
  end for
  return $\mathcal{R}^*$
end procedure

The process aims to synthesize revised requirements $\boldsymbol{\mathcal{R}}^* = \mathcal{R}^*_1, \mathcal{R}^*_2, \ldots, \mathcal{R}^*_n$ that incorporate inclusivity without compromising essential aspects of security and privacy. This is achieved through collaborative evaluation by requirement analysts, product owners, and security engineers. Furthermore, international standards and guidelines such as ISO 25010, WCAG, GDPR, and

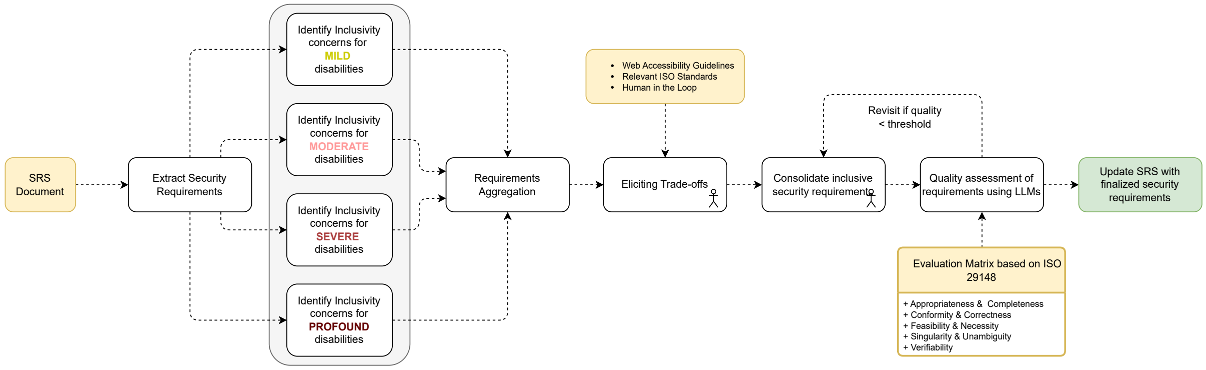

Fig. 2. Flow diagram of the proposed approach, LLMPathy

domain-specific DEI policies serve as reference points during trade-off analysis.

The elicitation mechanism ensures that the modified requirements are inclusive and secure, allowing responsible adoption of neurodiversity-aware design in practical settings. The algorithm 1 outlines the complete procedure for eliciting these inclusivity trade-offs.

D. *Quality Assessment of Updated Inclusivity Requirements*

This step assesses the updated requirements $\boldsymbol{\mathcal{R}}^*$ for quality, using the international industry standard ISO/IEC/IEEE 29148:2018, which defines best practices for requirements engineering in software and system development. This quality assessment is carried out using LLMs, which evaluate each requirement against the nine quality parameters $\boldsymbol{\mathcal{P}}$ mentioned in the standard, as listed in Fig. 2. Later, a quality score $\boldsymbol{\mathcal{Q}}_S$ is calculated and all the characteristics listed in ISO 29148:2018 are given the same weight in this calculation. If the quality score of any requirement is below the acceptable threshold $\boldsymbol{\mathcal{Q}}_{th}$, it is sent back to the previous stage for refinement and contextual realignment. The algorithm 2, shows the entire quality assessment process for these requirements and obtaining the finalised requirements $\boldsymbol{\mathcal{R}}^f$.

This step in the process mitigates any negative impact on the overall quality of these security requirements caused by accommodating inclusivity concerns introduced in Step-2. The final requirements after this stage are labelled $\boldsymbol{\mathcal{R}}^f$. Once the requirements are finalised, the SRS document is updated.

IV. Demonstrating the Approach: A Case Study

To demonstrate the proposed approach, LLMPathy was systematically applied to a realistic use case scenario, providing a complete walkthrough and demonstrating its applicability and effectiveness; the study was conducted by a group of six, including security practitioners and researchers.

Algorithm 2 Evaluating $\boldsymbol{\mathcal{R}}^*$ for Quality and Updating the SRS

procedure $\mathbb{Q}(\boldsymbol{\mathcal{R}}^*)$
    criteria $\leftarrow$ ISO-29148:2018($\mathcal{P}$)
    $\boldsymbol{\mathcal{R}}^f \leftarrow [\ ]$
    for all $\mathcal{R}_j \in \mathcal{R}^*$ **do**
        $\mathbb{Q}_S \leftarrow$ evaluate ($\boldsymbol{\mathcal{R}}_j$, criteria)
        if $\mathbb{Q}_S \geq \mathbb{Q}_{th}$ **then**
            $\mathcal{R}^f$.append($\mathcal{R}_j$)
        else
            $\mathcal{U} \leftarrow \text{refine}(\mathcal{R}_j, \mathbb{Q}_S)$
            $\mathcal{R}^f$.append($\mathcal{U}$)
        end if
    end for
    updated_SRS $\leftarrow$ update(SRS, $\mathcal{R}^f$)
    return updated_SRS
end procedure

A. *Use Case Scenario*

Consider a government initiative to deploy a mobile banking application designed for public sector pensioners. The mobile application facilitates the management of pensions, funds and tax documentation. The user base of this application comprises primarily of elderly individuals having potential cognitive impairments like memory decline, reduced attention span and language processing and comprehension difficulties.

B. *Step-1: Extraction of Security Requirements*

Security requirements extracted from the SRS document ($\boldsymbol{\mathcal{R}}_S$) are listed below:

$\boldsymbol{\mathcal{R}}_1$: User must re-authenticate after one minute of inactivity.

$\boldsymbol{\mathcal{R}}_2$: Account statements are available via secure download.
$\boldsymbol{\mathcal{R}}_3$: Two-factor authentication is required for all logins.

Additionally, stakeholder interviews with support staff and caregivers highlighted further requirements ($\boldsymbol{\mathcal{R}}_D$):

$\boldsymbol{\mathcal{R}}_4$: Users should be able to recover passwords with assistance.
$\boldsymbol{\mathcal{R}}_5$: Application should support simplified mode for cognitive impairments.

These five requirements form the combined set of inclusivity requirements $\mathcal{R}^+$.

C. Step-2: Analysis of Inclusivity Concerns by LLMs

For generating inclusivity concerns for $\mathcal{R}^+$, the human-in-the-loop analyst invokes LLMs for mild, moderate and severe cognitive impairment. The LLM for profound disabilities is skipped because such users do not independently use mobile applications. The three triggered LLMs have raised the following concerns which are mild, moderate and severe respectively.

$\boldsymbol{\mathcal{C}}_{R1,M1}$: Timeout in $\mathcal{R}_1$ may confuse forgetful users.
$\boldsymbol{\mathcal{C}}_{R2,M2}$: $\mathcal{R}_2$ assumes the ability to comprehend data in the account statement.
$\boldsymbol{\mathcal{C}}_{R3,M3}$: Two-factor authentication in $\mathcal{R}_3$ can be cognitively overwhelming.

D. Step-3: Elicitation of Tradeoffs

In this step, the security requirements are contextually rebalanced to accommodate the inclusivity concerns highlighted in the previous step. It recommends the addition of a biometric option for two-factor authentication in $\boldsymbol{\mathcal{R}}_3$ and the inclusion of a voice-based statement readout in $\boldsymbol{\mathcal{R}}_2$. The updated security requirements $\boldsymbol{\mathcal{R}}^*$ are listed below:

$\boldsymbol{\mathcal{R}}_1^*$: User must re-authenticate after three minute of inactivity.
$\boldsymbol{\mathcal{R}}_2^*$: Secure statements shall include audio narration as an accessibility feature.
$\boldsymbol{\mathcal{R}}_3^*$: Two-factor authentication shall support biometric fallback for users with cognitive impairments.

E. Step-4: Quality evaluation of $\boldsymbol{\mathcal{R}}^*$ with LLMs

Each updated requirement in $\mathcal{R}^*$ is assessed using the nine quality characteristics listed in ISO/IEC/IEEE 29148:2018 for the calculation of the quality score $\mathcal{Q}_S$. This quality assessment is conducted using LLM. Requirements are accepted only if their quality score $\mathcal{Q}_S$ is higher than the set threshold, which is 0.75 in this case. This value was chosen based on the criticality of the system, as 0.75 reflects reasonable compliance with the ISO 29148:2018 standard. For other use cases, this threshold can be adjusted depending on the criticality of the system. The calculated quality scores for the requirements are listed in the equation. 4, 5 and 6.

$$\mathcal{Q}_S(\mathcal{R}_1) = 0.89 > Q_{\text{th}} \tag{4}$$

$$\mathcal{Q}_S(\mathcal{R}_2) = 0.66 < Q_{\text{th}} \tag{5}$$

$$\mathcal{Q}_S(\mathcal{R}_3) = 0.82 > Q_{\text{th}} \tag{6}$$

Since $\mathcal{Q}_S$ of $\mathcal{R}_2$ is less than the threshold, the LLM suggests re-wording $\mathcal{R}_2$ for clarity and verifiability and feeds back to the previous stage for consolidation. Once all requirements have been updated, the finalised requirements $\boldsymbol{\mathcal{R}}^*$ are incorporated into the SRS document as shown in eq. 7.

$\boldsymbol{\mathcal{R}}_2^f$: Statements shall include an optional audio playback feature, with secure session logging.

$$\text{SRS}^{\text{updated}} = \text{Update}\Big(\text{SRS}, \bigcup_{f=1}^{5} \mathcal{R}^f\Big) \tag{7}$$

V. Conclusion and Future Work

Cognition plays an important role in enabling positive user-security interactions. Without considering cognitive users' limitations, we risk building complex security systems susceptible to human errors resulting in a breach. It will be unfair to blame cognitively challenged users in such a case. To contribute towards the goal of cognitive inclusivity in the design and development of cybersecurity systems and services, the paper presents an approach to identify concerns about inclusivity in security requirements. The approach utilises LLMs to identify these concerns, which then informs modification of the requirements to achieve in line with inclusion concerns. The approach presented in this paper has implications both for research and practice. More importantly, it considered a timely and required research problem, ensuring cognitively disabled users' right of equal access to technology.

Furthermore, from a future work perspective, it is intended to expose LLMPathy to the feedback of security practitioners and evaluate it from the perspective of industry readiness and applicability in real-world scenarios featuring development of the state-of-the-art security systems and services.

References

[1] A. P. C. D. Hylender, P. Langlois and S. Widup, "2024 data breach investigations report," 2024, pp. 55–60.
[2] G. World Health Organization, "Global report on health equity for persons with disabilities," 2022.

[3] C. for Disease Control, Prevention *et al.*, “Disability impacts all of us infographic,” *Disability and Health Data System (DHDS). Available online: https://www. cdc. gov/ncbddd/disabilityandhealth/pdf/disability-impacts-all-of-us-infographic. pdf (accessed on 22 August 2024)*, 2020.

[4] B. Naqvi, J. Kävrestad, and A. Islam, “Ensuring usable cybersecurity for all: Examining cognitive disabilities from research to industry,” *Available at SSRN 4609555*, 2024.

[5] V. Totsika, A. Liew, M. Absoud, C. Adnams, and E. Emerson, “Mental health problems in children with intellectual disability,” *The Lancet Child & Adolescent Health*, vol. 6, no. 6, pp. 432–444, 2022.

[6] P. D. Chowdhury and K. Renaud, “Advocating a policy push toward inclusive and secure “digital-first” societies,” *IEEE Security & Privacy*, 2024.

[7] B. K. Wiederhold, “Diverse minds, secure networks: Embracing neurodiversity in cybersecurity,” *Cyberpsychology, Behavior, and Social Networking*, vol. 27, no. 9, pp. 613–615, 2024.

[8] *Systems and software engineering — Life cycle processes — Requirements engineering*, International Organization for Standardization Standard ISO/IEC/IEEE 29 148:2018, 2018.

[9] K. Renaud and L. Coles-Kemp, “Accessible and inclusive cyber security: a nuanced and complex challenge,” *SN Computer Science*, vol. 3, no. 5, p. 346, 2022.

[10] G. A. Stelea, L. Sangeorzan, and N. Enache-David, “When cybersecurity meets accessibility: A holistic development architecture for inclusive cyber-secure web applications and websites,” *Future Internet*, vol. 17, no. 2, p. 67, 2025.

[11] S. Kivimäki, “A study into ideal software development for the disabled,” 2023.

[12] Y. P. C. Aguiar, E. Galy, A. Godde, M. Trémaud, and C. Tardif, “Autismguide: a usability guidelines to design software solutions for users with autism spectrum disorder,” *Behaviour & Information Technology*, vol. 41, no. 6, pp. 1132–1150, 2022.

[13] M. Krishna, B. Gaur, A. Verma, and P. Jalote, “Using llms in software requirements specifications: An empirical evaluation,” in *2024 IEEE 32nd International Requirements Engineering Conference (RE)*, 2024, pp. 475–483.

[14] N. Marques, R. R. Silva, and J. Bernardino, “Using chatgpt in software requirements engineering: A comprehensive review,” *Future Internet*, vol. 16, no. 6, 2024. [Online]. Available: https://www.mdpi.com/1999-5903/16/6/180

[15] H. Villamizar, M. Kalinowski, A. Garcia, and D. Mendez, “An efficient approach for reviewing security-related aspects in agile requirements specifications of web applications,” *Requirements engineering*, vol. 25, no. 4, pp. 439–468, 2020.

[16] A. F. Subahi, “Enhancing software sustainability: Leveraging large language models to evaluate security requirements fulfillment in requirements engineering,” *Systems*, vol. 13, no. 2, 2025. [Online]. Available: https://www.mdpi.com/2079-8954/13/2/114

[17] W. Aljedaani, M. M. Eler, and P. D. Parthasarathy, “Enhancing accessibility in software engineering projects with large language models (llms),” in *Proceedings of the 56th ACM Technical Symposium on Computer Science Education V. 1*, ser. SIGCSETS 2025. New York, NY, USA: Association for Computing Machinery, 2025, p. 25–31. [Online]. Available: https://doi.org/10.1145/3641554.3701841

[18] B. Tabarsi, H. Reichert, A. Limke, S. Kuttal, and T. Barnes, “Llms’ reshaping of people, processes, products, and society in software development: A comprehensive exploration with early adopters,” 2025. [Online]. Available: https://arxiv.org/abs/2503.05012

[19] K. Peffers, M. Rothenberger, T. Tuunanen, and R. Vaezi, “Design science research evaluation,” in *Design Science Research in Information Systems. Advances in Theory and Practice: 7th International Conference, DESRIST 2012, Las Vegas, NV, USA, May 14-15, 2012. Proceedings 7*. Springer, 2012, pp. 398–410.