Empowering Diversity by Building Inclusive Software Engineering Projects with Large Language Models

Clara Maathuis[1], Greg Alpar[2] and Stefano Bromuri[1]
[1]Open University of the Netherlands, Heerlen, The Netherlands
[2]Radboud University, Nijmegen, The Netherlands

clara.maathuis@ou.nl
greg.alpar@ru.nl
stefano.bromuri@ou.nl

Abstract: The integration of emerging AI technologies, particularly Large Language Models (LLMs), is fundamentally reshaping the landscape of software engineering. LLMs offer a wide array of capabilities that can enhance various aspects of software development, including design assistance, automated code analysis and synthesis, and testing. Consequently, their integration into software engineering practices holds the potential to greatly improve efficiency and code quality, marking a notable paradigm shift towards the creation of more intelligent and adaptive systems. This allows the definition of an extended perspective when establishing requirements for building software engineering solutions by incorporating evidence from literature resources and practice. By synthesizing requirements and evidence from scholarly and practitioner sources using LLMs, software engineers can ensure that their solutions are not only technically sound but also align with best practices in fostering social norms and values like inclusivity. Adopting such an approach supports the creation of responsible and inclusive educational software that caters to diverse learning needs and promotes equitable access to educational resources. Moreover, by using LLMs to inform decision-making throughout the software development lifecycle, software engineers can iteratively refine and enhance their solutions based on emerging research findings, thereby ensuring continuous improvement in fostering inclusive educational environments. Hence, this research aims to develop a novel evidence-based software engineering method informed by insights from scientific literature. As a use case, we design and implement a dyslexia-oriented educational software application that supports children in learning to read, guided by this new methodological approach.

Keywords: Evidence-based software engineering, AI, LLMs, Diversity, Neurodiversity, Dyslexia

1. Introduction

"Make reading a habit, and it will become a source of pleasure and knowledge." (Isaac Asimov)

The integration of LLMs in software engineering is transforming the software development process by enhancing various activities, including design assistance, automated code synthesis, and testing. This process is fostering a paradigm shift towards the development of more intelligent and adaptive systems, necessitating the exploration of novel methods and techniques that redefine traditional software development practices. At the same time, it is essential to prioritize societal well-being and balance ethical values like diversity, inclusivity, and social responsibility to ensure that software systems are beneficial, equitable, and sustainable. In educational settings, addressing diversity in software engineering projects through LLMs requires a nuanced and multidisciplinary approach to understand and tackle the multifaceted nature of this context. By focusing on developing inclusive software engineering processes with LLMs, it is possible to capture and implement the essential elements necessary for effective software development, leading to the creation of software systems that are not only efficient, but also socially responsible and sustainable (Huang et al., 2024; Bach et al,, 2022; Amatriain, 2024; Shah, 2024).

For children and individuals with dyslexia, the struggle with reading is often a daily and frustrating experience. When faced with a written text, their brains have difficulty processing the visual information, making it challenging to decode letters, to recognize words, and comprehend the meaning of the text. This can lead to feelings of anxiety, embarrassment, and low self-esteem. Children with dyslexia may experience difficulty sounding out words, remembering letter sequences, and understanding sentence structure, making reading a laborious and exhausting task (Mather, White & Youman, 2020; Akyurek et al., 2024). Further, impact can be observed on their academic performance, social interactions, and overall confidence, making it essential to provide targeted support to help them overcome their reading challenges. While various systems consider the integration of dyslexia-friendly supportive methods or are directly dedicated to dyslectic users (Akyurek et al., 2024), still, a limited number of studies and existing solutions are built using advanced AI technologies. To address this, we build a novel dyslexia-oriented software application prototype using LLMs, prompt engineering, and RAG (Retrieval Augmented Generation) based on extensive multidisciplinary literature review in an evidence-based software engineering methodological approach. To the best of our knowledge, this represents the first attempt in the literature in this direction and is based on an unique approach that can benefit from the development of state-of-the-art AI technologies applied in this domain. Building software engineering systems with Large Language Models (LLMs) in the context of dyslexia is important because it enables the creation of tailored, adaptive, and user-friendly tools that can assist children with dyslexia in overcoming their reading challenges, improve literacy skills, and increase their confidence. Hence, by leveraging LLMs, software engineers can design inclusive educational tools that support diverse learning needs, exemplifying the challenges and goals of inclusive software engineering.

The outline of the article is structured as follows. Section 2 discusses the technical dimension of the research background. Section 3 presents related studies that address diversity in the software engineering domain using advanced AI technologies such as LLMs. Section 4 presents the methodological approach adopted in this research. Section 5 introduces the design and development of the prototype proposed. At the end, concluding remarks and future research perspectives are provided.

2. Research Background

Software engineering encompasses the process of developing software to fulfil specific customer requirements. Initially, the customer or software developer identifies the problem and outlines the desired characteristics and features. Over time, changes to these requirements may be suggested and implemented. With a history that can be traced back to the 1960s when the advent of integrated circuits revolutionised the computer market, software engineering involves a broad range of activities that collectively constitute the software development process. These activities encompass requirements gathering, system design, software development, testing, deployment, maintenance, documentation, project management, collaboration and communication, and quality assurance (Royce, 1970; Boehm, 1979). Particularly in Agile development methodologies, these steps are often iterative and may overlap (Abrahamsson et al., 2017). The ultimate objective is to deliver a functional and reliable software system that effectively caters to the needs of its users. The advancement of software technology played a critical role in driving the progress of software engineering. Numerous methods and techniques were developed to facilitate efficient and dependable software production, and various programming environments provide integrated tools like compilers, debuggers, and syntax-oriented editors to support programming tasks. In the realm of modern software engineering, the continuous progress of software technology continues to shape the field, introducing novel methods and technologies that enhance the efficiency and reliability of software production. This includes practices like Agile development methodologies (Abrahamsson et al., 2017), DevOps, cloud computing, and AI, which revolutionised the entire software development, deployment, and maintenance processes (Bourque & Fairley, 2014).

An emerging paradigm that aims to integrate research findings and empirical evidence into software development practices aiming to support various stakeholder needs is named evidence-based software engineering (EBSE). This involves systematically collecting and analysing empirical data to support decision-making processes throughout the software development lifecycle, and further allow software engineers to make informed choices, mitigate risks, and improve the overall quality and efficiency of software projects (Kitchenham, Dyba & Jorgensen, 2004). This approach draws inspiration from the evidence-based practices established in other disciplines like psychology, medicine, and manufacturing while recognising the importance of basing software engineering decisions on empirical evidence rather than relying solely on expert knowledge and experience (Kitchenham, Budgen & Brereton, 2015). This facilitates the translation of research findings into practical design requirements and guidelines that can be applied in real-world contexts (Huang et al., 2024). Given the recent developments in the AI domain that include LLMs, a new avenue is opened for EBSE systems given the intrinsic ability of LLMs to process and generate human-like text that can be applied for various EBSE tasks like literature review, data extraction, and synthesis of requirements and guidelines formulation based on various data sources. LLMs can be fine-tuned on domain-specific datasets using prompt engineering techniques (Hadi et al., 2023; Minaee et al., 2024) to improve their performance in identifying and extracting key findings and empirical results from sources such as research articles and field standards and documentation. Prompt engineering involves providing the input prompts that guide the LLM towards the desired task, making use of its understanding capabilities (Bach et al,, 2022; Amatriain, 2024; Shah, 2024). Furthermore, RAG (Retrieval Augmented Generation) can be employed to augment the LLM's knowledge with external information retrieval from relevant field sources and provide more accurate and comprehensive evidence synthesis (Lewis et al., 2020; Feng et al., 2024). Hence, the integration of prompt engineering and RAG techniques into EBSE development has the potential to enhance the quality and reliability of the evidence extraction and synthesis process. In this way features and patterns are identified and translated to design requirements that are further accounted for implementation in the EBSE process. This mechanism can be applied following two strategies. First, extracting and summarizing the evidence that is provided to the LLM as a short set of requirements. And second, by downloading relevant evidence sources (e.g., research articles, technical and domain documentation) and implementing RAG to provide requirements through prompt engineering. In this research, both strategies are applied.

3. Related Studies

To sustain the ongoing progress of society in a fair, just, and sustainable manner, the diversity, equity, and inclusivity principles and values need to be acknowledged, respected, implemented, and promoted at local and global scale as well as at individual and collective levels, when necessary, through dedicated and proactive strategies (Gill et al., 2018). While diversity implies the recognition, appreciation, and celebration of the whole spectrum of human differences existing in society, equity relates to providing equal opportunities, resources, and access in the same way to all individuals. This is especially important for those who have historically experienced marginalisation. Inclusion refers to building and maintaining an environment where all individuals are respected, valued, and included, regardless of their particular characteristics or backgrounds (UN, 2019; ISC2, 2022; Russen and Dawson, 2023). These principles are strongly connected and contribute to addressing global challenges and to achieving sustainable goals related to the quality of education, gender equality, promotion of decent work and economic growth, reducing inequalities, and promoting peace, justice, and strong institutions (UN, 2015; Schwarz, 2023). Particularly, diversity, equity, and inclusion are essential in both education and work contexts for promoting the achievement of educational learning goals and work-related objectives, student or individual well-being while building and maintaining a positive social and educational development (Claeys-Kulik et al., 2023; Barnett, 2020).

In an educational context, assuring a diverse, equitable, and inclusive learning environment is crucial. Such an environment fosters exposure to diverse perspectives and experiences, enriching students ‘learning journey. Promoting interaction among students and with teachers cultivates empathy and respect within the educational community. Moreover, possible human and educational disparities that could occur based on aspects related to race, social status, or disabilities, are reduced while building and sustaining a positive and welcoming educational environment for students and staff through a general sense of community, acceptance, value, and respect that can only enhance the overall learning environment (UNESCO, 2017; Gretter et al., 2019; Washington, 2020). Concurrently, as part of the learning environment are the systems and tools used for supporting learning activities, processes, and overall communication. Designing educational systems with consideration for diversity, equity, and inclusion is essential for creating effective, efficient, and responsible educational programs that cater to the diverse needs and backgrounds of learners. (Fraser and Mancl, 2023). This underscores the value of inclusivity in promoting learning outcomes and preparing students for a diverse world Furthermore, embedding and respecting these principles also results to software that is relevant and usable to a wider range of users. This implies an improved user experience, more informed and effective decision-making benefiting from a broader range of viewpoints and insights, and an increased user engagement, creativity, and innovation (Jurado de Los Santos et al., 2020; Albusays et al., 2021; UN, 2022a; UN, 2022b).’

In the area of computer programming, Biswas (2023) investigates the role and capabilities of ChatGPT to provide users assistance, explanations, examples, and guidance for understanding programming concepts and techniques, finding useful resources, and solve specific technical problems. Accordingly, this study crosses a broad range of examples in Python and illustrates how ChatGPT could assist in solving problems in an effective way. In these lines, Biswas (2023) analyses the use of ChatGPT in Data Science projects to automate various project aspects such as data cleaning and pre-processing, model training, and interpretation of results. In higher education and professional development domains, Hassani and Silva (2023) conducts a study making use of GPT-3 to generate content on the study material and to provide a series of examples and prompts for tasks including automated essay scoring, research assistance, language translation, syllabus creation, quizzes, and exams for teaching purposes, and to generate summaries, reports, and other research-related instruments. In the context of using ChatGPT for learning and assessment of computer science curriculum at undergraduate level, Qureshi (2023) conducts an experimental study with two groups of students with a total number of 20 students whose performance was measured with and without making use of ChatGPT. The students participating in this study had to solve diverse tasks related to data structures and algorithms; for instance, sorting arrays, graph topological sorting, and Dijkstra’s algorithm. The results suggest that while ChatGPT is able to produce coherent and well-structured code, it generates promising outputs and comprehensive explanations. However, there is a need for a step-by-step approach to being guided by it in order for students to benefit from its guidance and to receive useful and well-explained results.

Wang et al., (2024) review LLMs comprehensively in the education domain to address various applications and benefits compared to traditional methods. The study categorizes educational applications based on user roles and scenarios, highlighting the transformative role of LLMs in education. The survey outlines datasets and benchmarks used for evaluating LLMs in educational settings, including question construction and automatic grading tasks. The authors further emphasize the importance of diversity in educational settings and stresses the challenges faced by certain demographic groups due to unequal access to educational models of comparable quality levels, highlighting the need for inclusivity. This directly relates to issues like social and language biases, fairness, and transparency of the models used (Maathuis & Chockalingam, 2024). These models need to be designed inclusively to accurately reflect the diversity of the student population and assure equitable learning opportunities for all students. In line with this, Dongre et al., (2024) explore ways to enhance empathy in LLMs by integrating psychological data and building the Empathic LLM (EmLLM) chatbot for stress monitoring and control in order to provide human-like responses and assessment. For neurodivergent individuals, Choi et al., (2024) emphasize the benefits of integrating conversational agents into their lives as they would be able to provide tailored responses to their needs, further stressing the risk of over-reliance on such solutions. From here, the authors recommend fostering autonomy and agency while respecting the unique needs of users. In a practical setting, Jang et al., (2024) simulate interactions between autistic individuals and LLMs revealing that many participants show a strong preference for LLM-generated over human advice, highlighting the potential for increased agency in navigating social situations. Nevertheless, the authors stress their concerns regarding the LLMs' tendency to predict best-case scenarios, potentially leaving individuals unprepared for unexpected interactions, and they highlight the importance of adopting an assistive framing in developing such technologies that truly serve the needs and autonomy of autistic individuals.

In the context of dyslexia, Akyurek et al., (2024) conduct a systematic review on current methods that improve cognitive skills such as reading skills of children, and recall promising methods like Sports-Vision Exercises, attention training, phonology reading writing programs, auditory and visual timing interventions, reading with masking interventions, working memory training, virtual reality neurorehabilitation treatments, and software programs like MAEVA(c) and RapDys(c). These methods target various aspects of reading including word accuracy, comprehension, and phonemic perception. Dyslexia is primarily caused by an imbalance in the functioning of the left brain hemisphere, and its core features can be classified as follows (Shaywitz, 1996; Egan & Tainturier, 2011; Savill & Thierry, 2011; Roitsch & Watson, 2019; Mather, White & Youman, 2020; Ahire et al., 2023):

- Reading and writing challenges: difficulty involving challenges in reading, word-spelling, writing, reasoning, word decoding, and other neurological traits regardless of intelligence level expressed through slow reading and writing skills, confusion among words, difficulty in reading single words in isolation without context, spelling errors, struggles with tasks, and incorrect perception of alphabets;
- Cognitive and processing difficulties: complications in various areas such as phonological, orthographic, working memory, brain systems asynchrony, poor executive function skills, and rapid naming processing;
- Language specific variations: variations per language as the linguistic demands vary and in languages like Chinese and Arabic, memory, orthographic knowledge, phonological awareness, and rapid naming play crucial roles in reading accuracy and fluency;
- Development and age-related differences: variations presents as individuals age, with young children struggling with sound-letter associations and older individuals facing issues with reading fluency, spelling, and written expression, which persist into adulthood.

As our review shows, various software engineering and LLMs-based solutions have been proposed for tackling various diversity aspects. Regarding dyslexia, to the best of our knowledge, while relevant solutions have been proposed for providing educational support in activities such as reading while reflecting on the potential impact produced, there exists a knowledge gap related to the incorporation of LLMs-based approaches for building an assistive intelligent solution for children with dyslexia.

4. Research Methodology

This research aims to build a software application with LLMs for reading assistance for children with dyslexia. On this behalf, the following research questions (RQ) are formulated:

RQ1: *What are the main features that characterize the reading process of children with dyslexia?*

RQ2: *What are the design requirements considered in the Prompt Engineering process?*

RQ3: *How can a software application be built by means of LLMs for reading assistance for children with dyslexia?*

To achieve this objective, an evidence-based software engineering approach is considered and implemented (Kitchenham, Budgen & Brereton, 2015; Huang et al., 2024) as depicted in Figure 1 below.

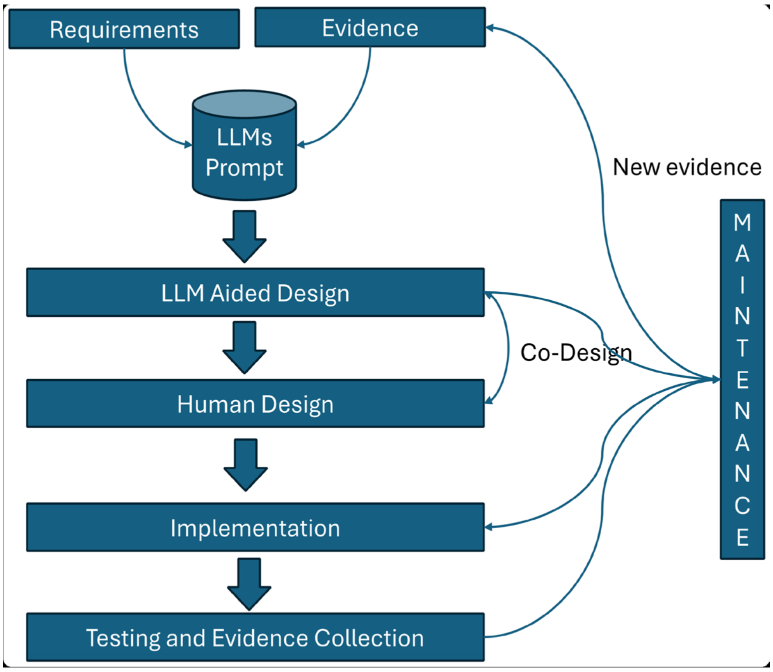

Figure 1: Methodological approach

In this process, the evidence is gathered from scientific sources and prompts are created to guide the LLM in the initial phase. Further, a collaborative co-design takes place, in which the human developers and the LLM iteratively refine and integrate responses from the LLM to develop the application in a unique human-AI teaming setting. At the end, the human developers take the lead by building the application integrating LLM-generated components and conducting initial tests to assess its functionality. After deployment, further refinement and improvements are possible based on the operation and usage of the application. In the next section, these phases are elaborated.

5. Solution

In **Phase 1 (Literature review)**, given the multidisciplinary nature of this research, an extensive literature review was conducted in the AI, software engineering, education, medicine, psychology, and diversity domains. In this process, combinations of keywords like *diversity, dyslexia, evidence-based software engineering, prompt engineering, LLMs, and ethics* were used to query scientific databases like IEEE Xplore, ACM Digital Library, Springer, Wiley, and Google Scholar.

In **Phase 2 (Experimentation with prompt engineering),** the collected studies were analysed and the translation of research findings into design requirements and implementation is executed through the following mechanisms:

- **Zero-shot prompting** where instructions are provided to ChatGPT 3.5 without examples or demonstration. In this way, the model generates its response based on its training data and task understanding. The prompt provided is as follows:

act as a software engineer and AI expert, write a software program in Python that is teaching reading.

The results provide a general program together with corresponding explanations that point to the fact that further functionalities like user progress tracking or a word database could be added.

- **Few-shot learning** where a small number of examples are provided to the model as part of the prompt to guide the model towards generating the desired output. In this case, the dyslexia context was given to the model using the prompt:

act as a software engineer and AI expert, write a software program in Python that is teaching reading taking into consideration dyslexia.

The results provide a program that mentions the incorporation of dyslexia-friendly font or style for displaying words.

- **Chain-of-thought prompting** where examples that demonstrate the step-by-step reasoning process leading to the solution are provided in the prompt. Here, features gathered synthetizing the findings obtained from the extensive literature review were provided to the model. The features contain textual, screen, sound, and educational elements, as follows:

act as a software engineer and AI expert, write a software program in Python that is teaching reading taking into consideration dyslexia. For this consider as features of the program the following ones: couple reading with audio, reading is preceded by phonological training, offer breaks to the users, consider adjustable timing, or even omit timing when reading, consider that the context of a word plays an important role, and distinguish between diagnosis and intervention.

The results provide a program that takes into consideration a part of these features and further reflecting on each of the category of features considered while pointing out on the possibility of including educational experts, speech therapists, and psychologists in such as project.

- **RAG** where the model is requested to retrieve relevant information from the research knowledge sources provided complementary to its internal knowledge. This retrieved information is then combined with the prompts provided by human developers to generate more informed and contextually relevant responses with respect to the design and development of the solution. Further, examples of prompts used are provided:

describe the relationship between dyslexia and reading.

describe tests to identify dyslexia.

describe effective approaches to train someone with dyslexia to read.

describe an application to learn how to read with dyslexia.

The results provide valuable information about the features considered and a series of reflective point for each category of features. Nevertheless, based on the human assessment conducted, while no significant points were found, the results are proportional to the level of evidence provided. This means that with a more multidisciplinary or transdisciplinary systematically analyzed evidence, valuable differences could potentially be seen.

In **Phase 3 (Solution development),** a prototype web application is built to assist children with dyslexia to read based on the evidence found and the LLM support provided. It is important to note that in this process, the use of RAG did not provide additional findings when compared to the shot-based approach. Accordingly, a preview of the website is provided in Figure 2.

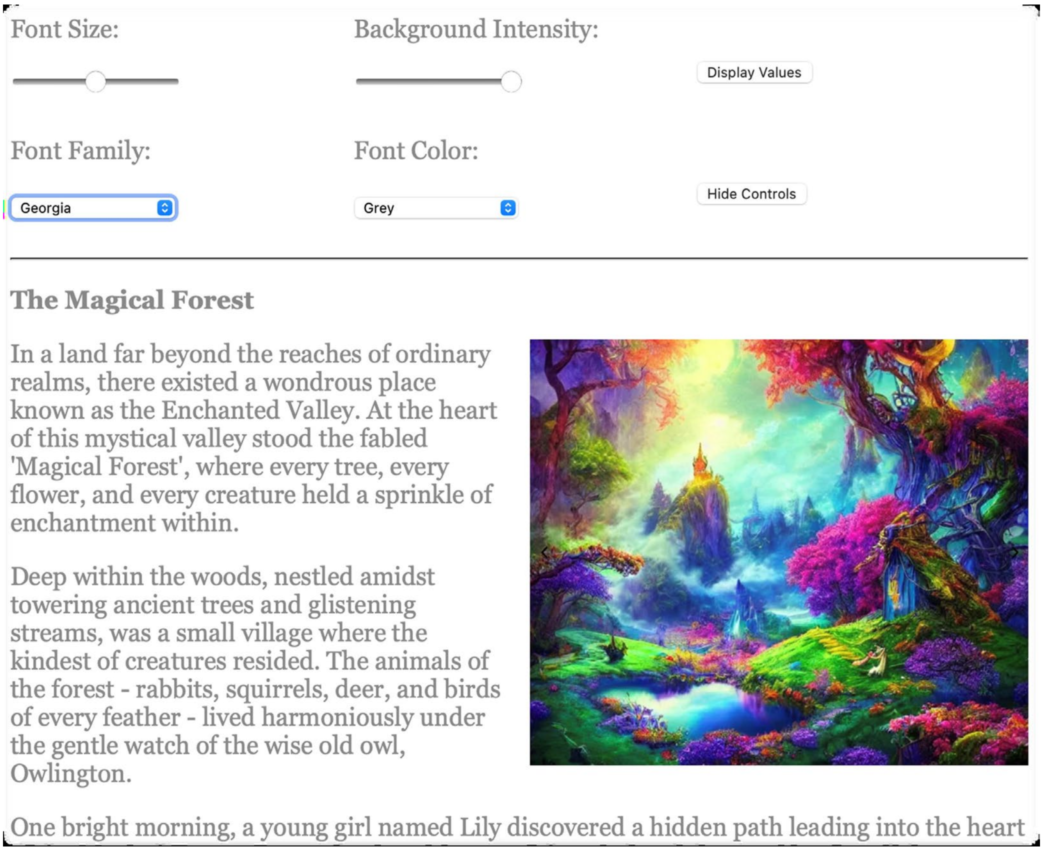

Figure 2: Prototype web application

The web application presents a fictional story and integrates a series of font size, type, and colour as well as background intensity controls that allow users to experiment and assess which combination is helpful in their reading process and assure an inclusive reading experience. As this represents a prototype, the intention is to evaluate it with experts and children. Based on the feedback received, the solution will be further adapted accordingly in future research.

6. Conclusions

This research introduces a novel evidence-based software engineering methodology designed to address the complexities of developing inclusive educational tools. By applying this new method, we aim to create innovative software solutions that cater to diverse learning needs, particularly for children (and adults) with dyslexia who have historically faced challenges in accessing educational resources. The methodology integrates scientific insights and practical knowledge while maintaining a user-centric perspective that prioritizes the specific needs of this audience. Our approach includes identifying the key features that characterize the reading process for children with dyslexia, establishing design requirements for effective prompt engineering, and ultimately developing a software application that provides tailored reading assistance to enhance their learning experience.

As future research perspectives, this research continues by conducting the evaluation of the proposed prototype with experts in dyslexia and children with dyslexia for refining and improving purposes. This evaluation process would not only ensure the application's effectiveness and usability, but would also provide valuable insights for future enhancements and research directions. Incorporating feedback from the target users is essential in creating an inclusive user-centric and impactful solution. Further developments aim to evolve the solution into a comprehensive and adaptive platform that addresses the diverse needs of children and dyslexia across different languages, cultures, and learning preferences. The evaluation will ultimately foster more inclusive and equitable educational landscape.

References

Abrahamsson, P., Salo, O., Ronkainen, J., & Warsta, J. (2017). Agile software development methods: Review and analysis. *arXiv preprint arXiv:1709.08439*.

Ahire, N., Awale, R. N, Patnaik, S. & Wagh, A. (2022). A comprehensive review of machine learning approaches for dyslexia diagnosis. *Multimedia Tools and Applications*, 82.

Albusays, K., Bjorn, P., Dabbish, L., Ford, D., Murphy-Hill, E., Serebrenik, A., & Storey, M. A. (2021). The diversity crisis in software development. *IEEE Software*, *38*(2), 19-25.

Amatriain, X. (2024). Prompt design and engineering: Introduction and advanced methods. *arXiv preprint arXiv:2401.14423*.

Akyurek, G., Koca, R. B., & Gunal Gunser, R. (2024). Systematic review of current approaches to cognitive skills in children with dyslexia. *Current Psychology*, 1-17.

Bach, S. H., Sanh, V., Yong, Z. X., Webson, A., Raffel, C., Nayak, N. V., ... & Rush, A. M. (2022). Promptsource: An integrated development environment and repository for natural language prompts. *arXiv preprint arXiv:2202.01279*.

Barnett, R. (2020). Leading with meaning: Why diversity, equity, and inclusion matters in US higher education. *Perspectives in Education*, *38*(2), 20-35.

Biswas, S. (2023). Role of ChatGPT in Computer Programming.: ChatGPT in Computer Programming. *Mesopotamian Journal of Computer Science*, *2023*, 8-16.

Boehm, B. W. (1979). Software Engineering-as it is. IEEE ICSE 1979, pp. 11-21.

Bourque, P. & Fairley, R. E. (2014). Guide to Software Engineering Body of Knowledge version 3.0. IEEE Computer Society.

Claeys-Kulik, A. L., Jørgensen, T. E., & Stöber, H. (2019). Diversity, equity and inclusion in European higher education institutions. *Results from the INVITED Project. Brussel: European University Association Asil, 51p*.

Choi, D., Lee, S., Kim, S. I., Lee, K., Yoo, H. J., Lee, S., & Hong, H. (2024, May). Unlock Life with a Chat (GPT): Integrating Conversational AI with Large Language Models into Everyday Lives of Autistic Individuals. In *Proceedings of the CHI Conference on Human Factors in Computing Systems* (pp. 1-17).

Dongre, P., Behravan, M., Gupta, K., Billinghurst, M., & Gračanin, D. (2024). Integrating Physiological Data with Large Language Models for Empathic Human-AI Interaction. *arXiv preprint arXiv:2404.15351*.

Egan J. & Tainturier M. J. (2011) Inflectional spelling deficits in developmental dyslexia. *Cortex*, 47(10):1179-96

Feng, Z., Feng, X., Zhao, D., Yang, M., & Qin, B. (2024). Retrieval-generation synergy augmented large language models. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 11661-11665). IEEE.

Fraser, S., & Mancl, D. (2023). Dimensions of Diversity, Equity, and Inclusion. *ACM SIGSOFT Software Engineering Notes*, *48*(2), 18-21.

Gill, G. K., McNally, M. J., & Berman, V. (2018, September). Effective diversity, equity, and inclusion practices. In *Healthcare management forum* (Vol. 31, No. 5, pp. 196-199). Sage CA: Los Angeles, CA: SAGE Publications.

Gretter, S., Yadav, A., Sands, P., & Hambrusch, S. (2019). Equitable learning environments in K-12 computing: Teachers' views on barriers to diversity. *ACM Transactions on Computing Education (TOCE)*, *19*(3), 1-16.

Hadi, M. U., Qureshi, R., Shah, A., Irfan, M., Zafar, A., Shaikh, M. B., ... & Mirjalili, S. (2023). A survey on large language models: Applications, challenges, limitations, and practical usage. *Authorea Preprints*.

Hassani, H., & Silva, E. S. (2023). The role of ChatGPT in data science: how ai-assisted conversational interfaces are revolutionizing the field. *Big data and cognitive computing*, *7*(2), 62.

Huang, Y., Chen, Y., Chen, X., Chen, J., Peng, R., Tang, Z., ... & Zheng, Z. (2024). Generative Software Engineering. *arXiv preprint arXiv:2403.02583*.

ISC2, 2022. 80 diversity and inclusion definitions. ISC2 Toolkit .

Kitchenham, B. A., Dyba, T., & Jorgensen, M. (2004). Evidence-based software engineering. In *Proceedings. 26th International Conference on Software Engineering* (pp. 273-281). IEEE.

Jurado de Los Santos, P., Moreno-Guerrero, A. J., Marín-Marín, J. A., & Soler Costa, R. (2020). The term equity in education: A literature review with scientific mapping in web of science. *International Journal of Environmental Research and Public Health*, *17*(10), 3526.

Jang, J., Moharana, S., Carrington, P., & Begel, A. (2024, May). "It's the only thing I can trust": Envisioning Large Language Model Use by Autistic Workers for Communication Assistance. In *Proceedings of the CHI Conference on Human Factors in Computing Systems* (pp. 1-18).

Kitchenham, B. A., Budgen, D., & Brereton, P. (2015). *Evidence-based software engineering and systematic reviews* (Vol. 4). CRC press.

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., ... & Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, *33*, 9459-9474.

Maathuis, C., & Chockalingam, S. (2024). Risk Assessment of Large Language Models Beyond Apocalyptic Visions. In *European Conference on Cyber Warfare and Security* (Vol. 23, No. 1, pp. 279-286).

Mather, N., White, J. & Youman, M. (2020). Dyslexia around the world: a snapshot

Minaee, S., Mikolov, T., Nikzad, N., Chenaghlu, M., Socher, R., Amatriain, X., & Gao, J. (2024). Large language models: A survey. *arXiv preprint arXiv:2402.06196*.

Qureshi, B. (2023). Exploring the use of chatgpt as a tool for learning and assessment in undergraduate computer science curriculum: Opportunities and challenges. *arXiv preprint arXiv:2304.11214*.

Rotisch, J. & Watson, S. M. (2019) An overview of dyslexia: definition, characteristics, assessment, identification, and intervention. *Science Journal of Education*, &(4).

Royce, W. W. (1970). Managing the development of large software systems.

Russen, M., & Dawson, M. (2023). Which should come first? Examining diversity, equity and inclusion. *International Journal of Contemporary Hospitality Management*, *36*(1), 25-40.

Savill, N. J., & Thierry, G. (2011). Reading for sound with dyslexia: Evidence for early orthographic and late phonological integration deficits. *Brain Research, 1385,* 192–205

Shaywitz S. E. (1996). Dyslexia. *Sci Am.,* 275(5):98-104.

Schwarz, S. K. (2023). Equity, Diversity, and Inclusion in the Journal and the United Nations Sustainable Development Goals. *Canadian Journal of Anesthesia/Journal canadien d'anesthésie*, *70*(6), 925-929.

Shah, C. (2024). From Prompt Engineering to Prompt Science With Human in the Loop. *arXiv preprint arXiv:2401.04122*.

UN (2015). What are the sustainable development goals. UN URL: https://www.undp.org/sustainable-development-goals.

UN (2019). United nations disability inclusion strategy. UN .

UN (2022a). European declaration on digital rights and principles for the digital decade. UN .

UN (2022b). Towards equity and inclusion in higher education in europe. UN .

UNESCO (2017). A guide for ensuring inclusion and equity in education. UNESCO.

Wang, S., Xu, T., Li, H., Zhang, C., Liang, J., Tang, J., ... & Wen, Q. (2024). Large language models for education: A survey and outlook. *arXiv preprint arXiv:2403.18105*.

Washington, A. N. (2020). When twice as good isn't enough: The case for cultural competence in computing. In *Proceedings of the 51st ACM technical symposium on computer science education* (pp. 213-219).