Proposed Artifacts

Complementary material for our paper entitled "*How are People with Disabilities Embraced in Software Development Teams? A Systematic Literature Review*".

Code	Artifact Name	Artifacts Description
[A1]	Auditory Code Overview and Navigation Tool (ACONT)	A tool designed to assist visually impaired programmers, in quickly scrolling through a class file to understand its structure. It utilizes a "*timeline*" navigation technique, allowing users to navigate line-by-line at their chosen speed. It also maps different sounds to programming constructs, such as a door opening for an "*if*" statement or a piano chord for a "*public*" function, to convey information non-visually [5].
[A2]	AudioHighlight	A tool that can be integrated into the Eclipse development environment or used as a web service. It allows blind programmers to quickly step through structural elements of code, such as class and function declarations and control flow statements, using HTML header tags to reflect the structure of the code [1].
[A3]	CodeTalk	A plugin for the Visual Studio IDE that aims to improve accessibility for visually impaired developers. It is designed to minimize developers' effort to search for information by extracting and presenting relevant data in an accessible way and introducing a secondary audio channel to complement the screen reader [10].
[A4]	CodeWalk	A set of features added to Microsoft's Live Share VS Code extension, designed to facilitate shared awareness in mixed-ability collaborative software development. It conveys sighted colleagues' navigation and edit actions through sound effects and speech, allowing BVI developers to participate more equally in coding tasks such as code review and refactoring [8].
[A5]	VSCode Improvements	Set of features integrated into Visual Studio Code (VSCode). An Accessible Terminal Buffer, which allows blind users to navigate using arrow keys, significantly improving the accessibility and usability of the integrated VSCode terminal. Audio Signals for Git Diff, Verbosity Settings, and Help Menus have also been implemented to enhance the non-visual programming experience in VSCode [12].
[A6]	Grid Editor	Grid Editor is an accessible programming tool designed to help blind and low-vision programmers navigate, understand, and edit source code in text-based programming languages. It presents code in a 2D grid structure, where each row represents a line of code, and each column represents an indentation level. It allows programmers to navigate code using arrow keys and edit code cells by pressing ENTER, making it easier to understand the context of the code and detect and correct syntax errors more efficiently. [4].
[A7]	No-Name	A code editor prototype specially designed for developers with physical impairments that integrates different code navigation approaches optimized for multimodal voice interaction [7].
[A8]	No-Name	Three Wizard of Oz prototypes to enhance communication between a deaf-blind employee and their colleagues. Consists of a Pager Alert System that includes a pager and a buzzer to notify the user of emergency situations quickly or to indicate that someone wants to initiate a personal interaction; an Add-on Braille Terminal Keys that allows the user to express emotions in emails, making communication more personal and less laborious; and an Accessible Messaging App with features that enable synchronous communication and tactile notifications for new messages, improving interaction in the workplace for deaf-blind people [13].
[A9]	Physical computing Streaming Sensor data Toolkit (PSST)	It is a toolkit designed to enable blind and visually impaired (BVI) developers to create accessible data displays, particularly for understanding sensor-generated data. It provides the developers with tools to author sonification of streaming sensor data, making physical computing platforms more accessible [9].
[A10]	StructJumper	Is a prototype of an Eclipse plugin designed to help blind programmers navigate and understand code structure in integrated development environments (IDEs). It allows blind programmers to quickly navigate through code, using a hierarchical tree that represents the nesting structure of a Java class [2].
[A11]	Accessible Web Modeler (AWMo)	It is a prototype web tool designed to allow software developers, including those with visual impairments, to work with graphical software models and diagrams, offering two distinct views for working with UML (Unified Modeling Language) class diagrams: one graphic and one textual vision of the same elements [3].
[A12]	CodeMirror-Blocks (CMB)	It is a browser-based programming environment designed to be fully accessible for visually impaired programmers. It provides spoken descriptions of the program structure and allows navigation using keyboard shortcuts [11].
[A13]	Accessible Blockly	It is a prototype of an accessible block-based programming library designed to enable users with visual impairments to create and navigate block-based code using a screen reader and keyboard [6].

REFERENCES

[1] Ameer Armaly, Paige Rodeghero, and Collin McMillan. 2018. AudioHighlight: Code Skimming for Blind Programmers. In *2018 IEEE International Conference on Software Maintenance and Evolution (ICSME)*. 206–216. https://doi.org/10.1109/ICSME.2018.00030

[2] Catherine M. Baker, Lauren R. Milne, and Richard E. Ladner. 2015. StructJumper: A Tool to Help Blind Programmers Navigate and Understand the Structure of Code. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (Seoul, Republic of Korea) *(CHI '15)*. Association for Computing Machinery, New York, NY, USA, 3043–3052. https://doi.org/10.1145/2702123.2702589

[3] Filipe Del Nero Grillo and Renata Pontin de Mattos Fortes. 2014. Tests with Blind Programmers Using AWMo: An Accessible Web Modeling Tool. In *Universal Access in Human-Computer Interaction. Design and Development Methods for Universal Access*, Constantine Stephanidis and Margherita Antona (Eds.). Springer International Publishing, Cham, 104–113.

[4] Md Ehtesham-Ul-Haque, Syed Mostofa Monsur, and Syed Masum Billah. 2022. Grid-Coding: An Accessible, Efficient, and Structured Coding Paradigm for Blind and Low-Vision Programmers. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology* (Bend, OR, USA) *(UIST '22)*. Association for Computing Machinery, New York, NY, USA, Article 44, 21 pages. https://doi.org/10.1145/3526113.3545620

[5] Joe Hutchinson and Oussama Metatla. 2018. An Initial Investigation into Non-visual Code Structure Overview Through Speech, Non-speech and Spearcons. In *Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems* (<conf-loc>, <city>Montreal QC</city>, <country>Canada</country>, </conf-loc>) *(CHI EA '18)*. Association for Computing Machinery, New York, NY, USA, 1–6. https://doi.org/10.1145/3170427.3188696

[6] Aboubakar Mountapmbeme, Obianuju Okafor, and Stephanie Ludi. 2022. Accessible Blockly: An Accessible Block-Based Programming Library for People with Visual Impairments. In *Proceedings of the 24th International ACM SIGACCESS Conference on Computers and Accessibility* (Athens, Greece) *(ASSETS '22)*. Association for Computing Machinery, New York, NY, USA, Article 19, 15 pages. https://doi.org/10.1145/3517428.3544806

[7] Bharat Paudyal, Chris Creed, Ian Williams, and Maite Frutos-Pascual. 2022. Inclusive Multimodal Voice Interaction for Code Navigation. In *Proceedings of the 2022 International Conference on Multimodal Interaction* (Bengaluru, India) *(ICMI '22)*. Association for Computing Machinery, New York, NY, USA, 509–519. https://doi.org/10.1145/3536221.3556600

[8] Venkatesh Potluri, Maulishree Pandey, Andrew Begel, Michael Barnett, and Scott Reitherman. 2022. CodeWalk: Facilitating Shared Awareness in Mixed-Ability Collaborative Software Development. In *Proceedings of the 24th International ACM SIGACCESS Conference on Computers and Accessibility* (Athens, Greece) *(ASSETS '22)*. Association for Computing Machinery, New York, NY, USA, Article 20, 16 pages. https://doi.org/10.1145/3517428.3544812

[9] Venkatesh Potluri, John Thompson, James Devine, Bongshin Lee, Nora Morsi, Peli De Halleux, Steve Hodges, and Jennifer Mankoff. 2022. PSST: Enabling Blind or Visually Impaired Developers to Author Sonifications of Streaming Sensor Data. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology* (Bend, OR, USA) *(UIST '22)*. Association for Computing Machinery, New York, NY, USA, Article 46, 13 pages. https://doi.org/10.1145/3526113.3545700

[10] Venkatesh Potluri, Priyan Vaithilingam, Suresh Iyengar, Y. Vidya, Manohar Swaminathan, and Gopal Srinivasa. 2018. CodeTalk: Improving Programming Environment Accessibility for Visually Impaired Developers. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (<conf-loc>, <city>Montreal QC</city>, <country>Canada</country>, </conf-loc>) *(CHI '18)*. Association for Computing Machinery, New York, NY, USA, 1–11. https://doi.org/10.1145/3173574.3174192

[11] Emmanuel Schanzer, Sina Bahram, and Shriram Krishnamurthi. 2019. Accessible AST-Based Programming for Visually-Impaired Programmers. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education* (Minneapolis, MN, USA) *(SIGCSE '19)*. Association for Computing Machinery, New York, NY, USA, 773–779. https://doi.org/10.1145/3287324.3287499

[12] JooYoung Seo and Megan Rogge. 2023. Coding Non-Visually in Visual Studio Code: Collaboration Towards Accessible Development Environment for Blind Programmers. In *Proceedings of the 25th International ACM SIGACCESS Conference on Computers and Accessibility* (<conf-loc>, <city>New York</city>, <state>NY</state>, <country>USA</country>, </conf-loc>) *(ASSETS '23)*. Association for Computing Machinery, New York, NY, USA, Article 102, 9 pages. https://doi.org/10.1145/3597638.3614550

[13] Abigail Tan, Michael Robin, Craig Williams, and LouAnne Boyd. 2020. Prototyping Accessible Work Systems with a Deaf-Blind Employee. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems* (<conf-loc>, <city>Honolulu</city>, <state>HI</state>, <country>USA</country>, </conf-loc>) *(CHI EA '20)*. Association for Computing Machinery, New York, NY, USA, 1–6. https://doi.org/10.1145/3334480.3382859