

A Design Science Approach to Developing Synchronized Tools for Mental Workload Assessment in Software Engineers^{*}

Sameera Gamage^{1,*†}, Pantea Keikhosrokiani^{2†}

¹University of Oulu, P.O. Box 123, Oulu, 90014, Northern Ostrobothnia, Finland

Abstract

Mental workload (MWL) critically affects productivity and well-being in high-cognitive fields like software engineering. Traditional MWL assessments that are based solely on subjective or physiological measures often lack reliability and relevance in the real world. This research presents a custom-built MWL data collection framework for software engineers, comprising a Task Simulation Desktop Application (TSDA) and a Self-Reporting Mobile Application (SRMA). TSDA simulates real-world software tasks, while SRMA collects NASA-TLX ratings synchronized with EEG signals.

Following the Design Science Research (DSR) methodology, the tools underwent iterative development and pilot testing, leading to improvements in usability, data accuracy, and system reliability. The dual source system allows for synchronized EEG and subjective workload evaluation, laying the foundation for future machine learning-based real-time workload prediction and adaptive cognitive load management. The developed tools offer scalable, structured, and domain-specific support to improving software engineer performance and mental well-being by supporting improved data collection.

Keywords

Mental Workload Assessment, Software Engineering, EEG-based Measurement, Design Science Research, Task Simulation Tools

1. Introduction

Mental workload (MWL) significantly influences task performance, cognitive efficiency, and well-being, particularly in high-demand fields such as software engineering [1]. Accurate MWL assessment is essential for enhancing productivity, reducing cognitive strain, and improving decision-making [2]. Traditional MWL assessment methods like NASA-TLX and physiological measures such as EEG have limitations due to subjective bias and sensitivity to external factors.

Software engineering tasks (e.g., debugging, problem solving, multitasking) impose substantial cognitive loads, yet current assessment tools fail to capture real-time workload variations specific to this domain. Existing methods like primary and secondary task performance are influenced by skill levels or introduce additional cognitive burdens, while physiological signals require careful contextual interpretation [3, 4, 5].

This research presents a custom-built MWL assessment framework integrating:

1. Task Simulation Desktop Application (TSDA) – replicates software engineering tasks in a controlled setting.
2. Self-Reporting Mobile Application (SRMA) – collects NASA-TLX assessments alongside EEG data.

TKTP 2025: Annual Doctoral Symposium of Computer Science, 2.–3.6.2025 Helsinki, Finland

*Corresponding author.

† Sameera Gamage led the research design, implementation, data analysis, and manuscript preparation. Pantea Keikhosrokiani provided supervision and feedback.

✉ sameera.gamage@oulu.fi (S. Gamage); pantea.keikhosrokiani@oulu.fi (P. Keikhosrokiani)

🌐 <https://www.linkedin.com/in/sameera-gamage/> (S. Gamage);

<https://www.linkedin.com/in/pantea-keikhosrokiani-55145a276/> (P. Keikhosrokiani)

🆔 0009-0009-7389-4934 (S. Gamage); 0000-0003-4705-2732 (P. Keikhosrokiani)



© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

Using synchronized EEG and subjective reports, this dual-source system enables real-time MWL validation tailored to software engineers. The research follows the Design Science Research (DSR) methodology with iterative development and usability testing.

The primary objectives of this research are the following.

1. Identify functional requirements for EEG-based MWL assessment tools in software engineering.
2. Analyze the relationship between EEG-derived metrics and NASA-TLX self-reports.
3. Develop secure and user-centered applications for synchronized MWL data collection.
4. Evaluate the effectiveness of tools through empirical testing and user feedback.

This research generates both procedural knowledge, in the form of a more practical framework for synchronized MWL data collection, and empirical knowledge through validation of the developed tools using real-world simulation data from software engineers. The development process also contributes to the knowledge of designing by applying and adapting the DSR methodology to the development of tools in the context of cognitive load measurement.

The knowledge produced by this research is expected to support future research and applications in real-time mental workload prediction, adaptive task scheduling, and cognitive-aware tool development and utilization in software development environments. Specifically, validated tools and collected datasets can serve as a foundation for machine learning-based cognitive load classifiers and interventions to improve developer productivity and well-being.

2. Background

While working on multitasks expecting to achieve critical deadlines, SE face higher levels of Cognitive loads. High cognitive load affects productivity, code quality, and mental well-being, leading to burnout and increased software defects. Tang et al. (2024) analyzed developer behaviors in validating AI-generated code using eye tracking and found high mental demand in IDE-based workflows [6]. Nakasai et al. (2024) measured the MWL of software developers using physiological indicators, providing objective information on cognitive strain [7]. Astuti et al. (2024) explored the impact of techno-stress on millennial work-life balance in digital work, focusing on its mediating role in employee well-being and job satisfaction [8]. Schott et al. (2024) examined usability, cognitive load, and presence in Virtual Reality (VR) environments, emphasizing how copresence and social interaction influence cognitive demands in mixed reality applications [9]. Dourado et al. (2024) evaluated MWL in Industry 5.0 wearable Augmented Reality (AR) systems, demonstrating high cognitive strain during software-related tasks [10]. These studies emphasize the need for workload optimization strategies to improve developer efficiency and well-being while reducing defects and cognitive overload in software engineering. It is important to identify tasks that increase the mental workload of SE.

2.1. Mental Workload (MWL): Concepts and Definitions

Mental workload (MWL) refers to the cognitive effort required to complete a task and is understood through resource-based and subjective perspectives [3]. The Multi-Resource Model (MRM) and Yerkes-Dodson law provide foundational models for understanding how MWL influences performance [4, 2]. MWL is critical to optimize in high-demand domains like software engineering, HCI, and aviation, where poor workload management can impact productivity and safety [5]. Understanding MWL supports designing effective tools for workload evaluation in dynamic environments.

2.2. Traditional Techniques for Mental Workload Measurement

MWL measurement techniques fall into three categories: primary task performance, secondary task performance, and subjective rating scales. Each has specific strengths and limitations. Primary Task Performance assesses accuracy, speed, or decision-making under load, but results may be influenced by individual differences or ceiling/floor effects. Combined use with physiological and subjective metrics

can improve the precision of assessment [2]. Secondary Task Performance introduces a parallel task to evaluate the reduction in performance under load. Although effective in detecting moderate workload changes, it can increase total cognitive load and interfere with task execution [6]. In addition, Subjective Rating Scales perform subjective assessments, especially NASA-TLX, are widely used because of their simplicity and applicability, despite susceptibility to bias. Table 1 compares subjective popular MWL scales.

Table 1
Comparison of Subjective Mental Workload Assessment Methods

Method	Pros	Cons	Vs. NASA-TLX
NASA-TLX [11]	Multi-dimensional, validated, task-sensitive	Subjective bias, recall errors	Baseline reference
SWAT [12]	Captures time, effort, stress	Needs pre-task ranking, complex	Structured but less flexible
Workload Profile [12]	Detailed cognitive breakdown	Complex scoring, time-consuming	Richer but impractical
RSME [13, 14]	Simple, real-time	Single-dimension	TLX offers more depth
DSSQ [13]	Includes emotional states	More for stress than MWL	TLX better for cognitive focus
CHRS [15]	Ideal for aviation	Task-specific, limited use	TLX more domain-flexible
ISA [13]	Real-time, quick feedback	Interrupts tasks, low detail	TLX more comprehensive

2.3. Design Science Research Method

Design Science Research Methodology (DSRM) enables structured artifact development through iterative cycles of relevance, rigor, and evaluation [16, 17]. Figure 1 outlines the six DSRM stages used in this research.

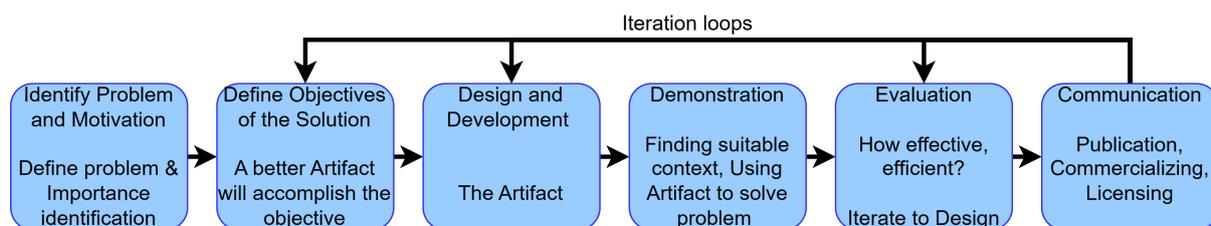


Figure 1: Design Science Research Steps

This research follows the combined approach of the DSR and Action Research (DSAR) approach [18], supporting the iterative development of TSDA and SRMA tools through the collection of mixed-method requirement gathering and empirical feedback. The method ensures that the research remains both scientifically rigorous and practically applicable in real-world software engineering settings.

3. Methodology

Given the gaps illustrated in Section 2, the development of TSDA and SRMA followed a DSR methodology to address the lack of synchronized domain-specific MWL assessment tools. Figure 2 presents the inputs and outputs of each cycle of the DSR process in the artifact development process.

3.1. Relevance Cycle

This cycle focused on identifying problems related to MWL, workforce efficiency, and well-being in software engineering. The literature highlighted the need for an optimized workload distribution and improved performance through the accurate detection of MWL. Traditional methods were found inadequate due to the dynamic nature of software engineering tasks. Consequently, this phase defined the core components such as task types, data collection approaches, involved physiological signals, and contributing cognitive organs.

3.2. Design Cycle

Key system requirements were defined, including the collection of performance metrics and controlled task simulations to ensure unbiased data. The company environments were unsuitable due to operational interference. Preliminary interviews with software engineers identified real-world task themes such as deadlines, multitasking, and R&D workflows. These insights informed the task simulation design and the system functionality.

3.3. Rigor Cycle

TSDA and SRMA were built on validated cognitive load models and usability principles. Existing research on EEG-based estimation and NASA-TLX was integrated for development guidance. Continuous EEG data required SEs to perform non-overlapping tasks to ensure traceability. Empirical testing and iterative refinement ensured scientific validity and practical utility. EEG fluctuations were planned to be mapped to individual tasks to support accurate MWL interpretation.

3.4. Evaluation Cycle

This phase involved systematic testing and refinement of TSDA and SRMA. Effectiveness was measured through correlations between EEG signals and NASA-TLX scores, as well as task performance metrics (e.g., accuracy, time). Usability testing improved interface design and task integration. Visual comparisons of data sets validated tool interoperability. Participant feedback informed refinements, enhancing system stability and data reliability. Table 2 summarizes the stages and their corresponding DSRM cycles.

Table 2
Design Science Research Methodology (DSRM) Stages and Cycles

Stage	Cycle Interaction	Approach Summary
Problem Identification	Cycle 1: Relevance	Review MWL in software engineering, identify assessment gaps, define tool requirements.
Define Objectives	Cycle 1: Relevance	Identify MWL factors, define task simulations, select assessment techniques.
Design & Development	Cycles 1–4: Relevance, Design, Rigor	Build TSDA and SRMA, integrate EEG, ensure cross-component data sync.
Demonstration	Cycle 4: Rigor	Pilot test with engineers, gather feedback, assess EEG-tool integration.
Evaluation	Cycles 4–5: Rigor, Evaluation	Compare NASA-TLX and EEG data, analyze performance, refine tools.
Communication	Cycle 6: Communication	Publish results, present at events, release tools on public platforms.

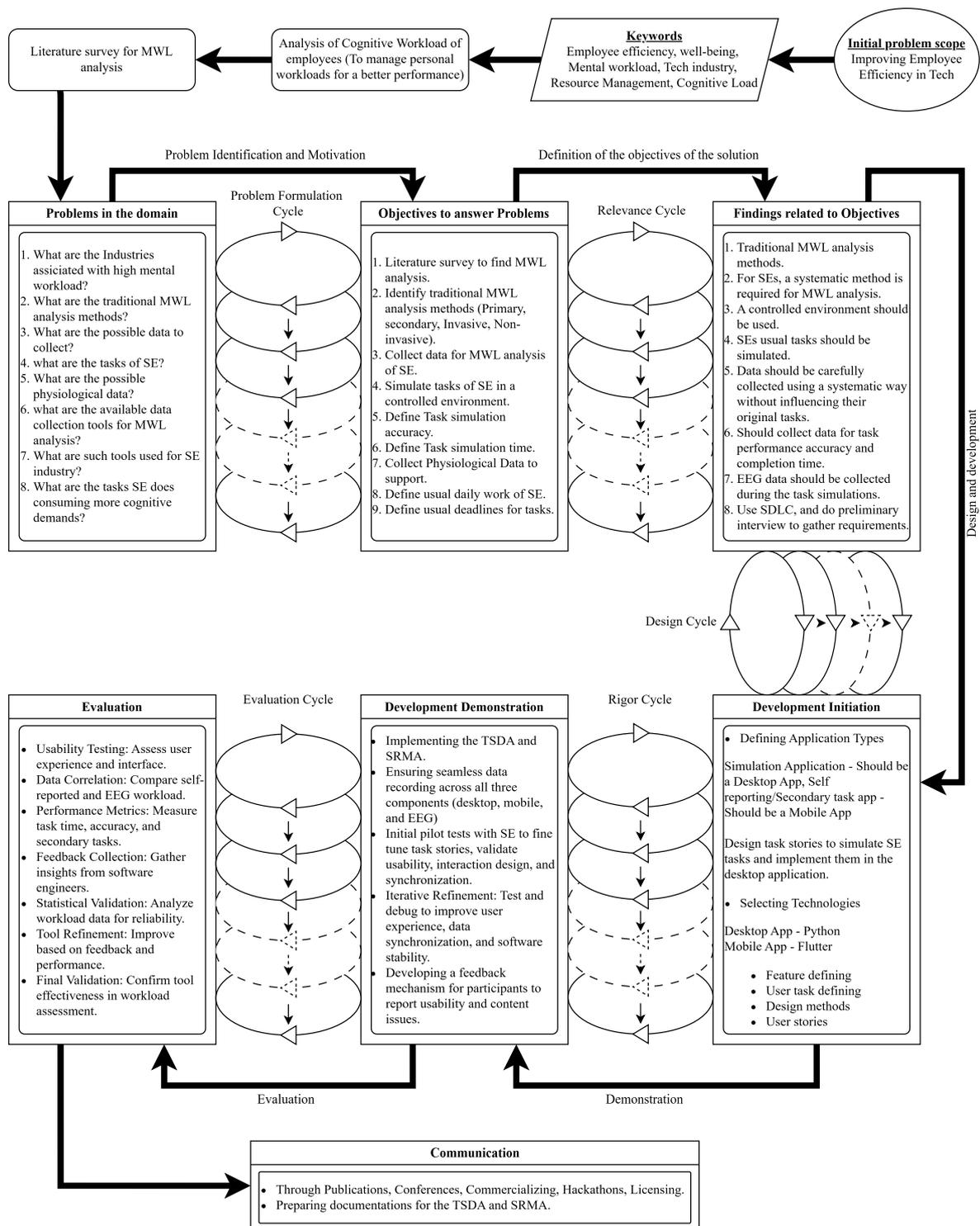


Figure 2: Relevant Design Science Research Steps

4. Research Implementation

4.1. Artifact Development phase

TSDA and SRMA were developed using Agile methodology in iterative sprints. The interviews guided task realism, UI design, and system requirements. The features of both applications are detailed in Table 3. TSDA replicates real-world SE tasks; SRMA captures NASA-TLX self-reports.

The non-functional requirements of the developed tools focused on performance, usability, security,

Table 3
Features of TSDA and SRMA

Task Simulation Desktop Application (TSDA)	Self Reporting Mobile Application (SRMA)
1. User Interface	
<ul style="list-style-type: none"> • Overview of experiment and tasks. • Task-specific instruction pages. • Final summary page. 	<ul style="list-style-type: none"> • Clear task instructions. • Six visual analog scale dimensions: Mental, Physical, Temporal Demand, Performance, Effort, Frustration. • Sliders for input. • Upload success/failure messages.
2. Task Simulation	
<ul style="list-style-type: none"> • Simulates real SE tasks. • Multiple difficulty levels. • Tracks completion time and accuracy. 	<ul style="list-style-type: none"> • Integrates NASA-TLX form. • Treated as secondary task capturing cognitive load.
3. Data Management	
<ul style="list-style-type: none"> • Uses Firebase for real-time storage. • Stores user task metadata and results. 	<ul style="list-style-type: none"> • Secure cloud upload. • Data anonymization for privacy.

and compatibility. The TSDA ensures fast response times, smooth task simulations, and an intuitive UI with clear instructions. The SRMA features a user-friendly interface, optimized for Android with quick interaction and low battery consumption. Both tools prioritize secure data handling through Firebase integration, encryption, and access control. Furthermore, the applications are designed to be compatible with all platforms, Windows, macOS, and various screen sizes, ensuring broad accessibility and user reliability. Table 4 outlines current limitations in real-world simulation and self-report accuracy.

Table 4
Limitations of the TSDA and SRMA

Task Simulation Desktop Application (TSDA)	Self Reporting Mobile Application (SRMA)
Relies on user honesty and accurate task performing during the test.	Relies on user honesty and accurate self-reporting, which can be susceptible to bias.
The simulated tasks may not perfectly replicate real-world experiences in chosen career paths.	Requires users to understand the meaning of each workload dimension.
Users may perform with lower stress in controlled environments, since they are not performing actual tasks.	May not capture the full user experience.
Student participants may not fully reflect real-world cognitive demands.	Data transmission requires secure handling.
	Limited mobile UI space.

4.2. Implementation of the artifact development

TSDA includes subtasks that simulate a complete sprint cycle, from planning to deployment and learning. SRMA includes the NASA-TLX form to capture the perceived workload. The following steps reflect a standard sprint workflow. It begins with Sprint Planning to define goals, review and prioritize user stories, and estimate effort. The user story refinement follows to clarify requirements, decompose stories into tasks, and set acceptance criteria. During development, features are coded using best practices, version control is maintained, and pull requests are submitted. Code review involves peer reviews, addressing feedback, and improving code quality. Testing includes unit, integration, and regression tests to ensure reliability. In the documentation, technical documentation, and usage guidelines are updated. The sprint review demonstrates features, gathers feedback, and identifies improvements. Finally, Continuous Learning promotes exploring new tools, practices, and sharing knowledge with the team.

A similar set of tasks simulated a typical high-workload scenario in software development, focusing on implementing a user log-in feature for a web site in TSDA.

4.3. Testing Validation and Refinements

Usability Testing, Qualitative Feedback, and Refinement: An iterative feedback process was used for refinement. Think-Aloud and post-test questionnaires helped identify usability issues. Adjustments included improving UI transitions, increasing task realism, and ensuring data consistency. The key feedback and actions are shown in Table 5.

Table 5
Summary of Qualitative User Feedback and Implemented Changes

Feedback Theme	Participant Insights	Implemented Changes
Usability & Interaction	Interface was intuitive but needed smoother transitions.	Improved UI design and optimized workflow transitions.
Task Simulation Realism	Coding tasks lacked complexity.	Increased task complexity and added IDE integration.
MWL Data Collection	Self-reported scores lacked consistency.	Implemented real-time validation between EEG and self-reported data.
Technical Issues	Occasional delays in Firebase data syncing.	Optimized data handling and added confirmation messages.

Pilot Testing and System Performance Evaluation is a three-phase pilot study that confirmed data synchronization, ease of workload entry, and log accuracy. This ensured that task timestamps were aligned with MWL ratings. The post-test refinements addressed stability, clarity, and response times.

Final Validation and Data Collection Workflow is there to ensure that TSDA and SRMA collect and synchronize MWL data effectively, a structured validation process was conducted before large-scale deployment. This phase focused on verifying the accuracy of the data synchronization and logging processes. The validation process ensured that the recorded workload values were correctly stored in Firebase and cross-checked with task execution logs to prevent inconsistencies. Further analysis was performed to validate data synchronization and system reliability. The system logs were reviewed to confirm that no workload submission attempts were lost, corrupted, or unsynchronized. By ensuring that the workload entries aligned precisely with task timestamps, the validation process reinforced data integrity and ensured that the developed tools provide a reliable foundation for future MWL analysis and machine learning-based workload predictions.

5. Results

In this research, software development processes were followed to develop specific and relevant applications for data collection for MWL analysis of employees in an organization. As a profession, software engineers were selected as there was a lack of research conducted on SE MWL analysis. Two applications were developed to collect data while performing their usual tasks. Primary and secondary tasks were considered to collect MWL data. Task performance data, such as,

1. Task performance accuracy and task completion time (collected from TSDA),
2. Self reported MWL as a secondary data collection (collected from SRMA),

were collected while recording the behavior of the cognitive load using the Emotive EEG signal sensing wireless device.

5.1. Task simulation desktop application

TSDA was designed to simulate real-world tasks faced by software engineers in a controlled way. It was developed during the Design and Development phase of the DSR methodology, based on the needs

identified in the Relevance Cycle and refined through feedback in the Evaluation Cycle.

Key Functionalities are as follows,

1. Landing and Consent Page: The application starts with explaining the research purpose and task details. Participants select the button "Software Engineering" and give their informed consent. This aligns with ethical considerations discussed in the Relevance and Evaluation cycles.
2. User Identification and Workload Expectation Input: Here, participants enter their user ID and rate how difficult they expect the task to be. This helps compare their initial expectation with their actual performance later, supporting the Rigor Cycle in the DSR process.
3. Task Scenarios and Simulations: The TSDA then presents a series of interactive tasks simulating a software sprint with realistic sub-tasks such as,
 - a) User Story Prioritization (drag-and-drop),
 - b) Priority Matrix Assignment (marking urgency and importance), is shown in Figure 3a.
 - c) Code Snippet Sequencing (arranging code segments for functionality),
 - d) Code Review (detecting errors),
 - e) Testing (Positive/Negative/Security),
 - f) Final Summary Dashboard displaying accuracy and completion time.

Each task is based on real software engineering activities identified through interviews. This ensures the simulation reflects real work, as defined in the Design Cycle of the DSR process.

4. Data Logging and Firebase Integration: During the session, user actions like time and accuracy are recorded and uploaded to Firebase. This allows real-time data collection and syncing with the EEG and SRMA app inputs, supporting the Rigor and Evaluation Cycles.

5.2. Self reporting mobile application

SRMA was developed to collect subjective MWL data using NASA-TLX ratings. It works alongside TSDA to capture self-reported feedback after each sub-task, allowing synchronized EEG analysis. The SRMA includes three key screens (Figure 3b), each designed iteratively and aligned with the DSR methodology. Functionality and DSR Alignment are described in the following.

1. Landing Page: Users enter their unique ID (same as TSDA), ensuring data linkage across both tools and click the button Software Engineer. Developed during the Design Cycle, and based on stakeholder interviews, emphasizing simple onboarding and real-world applicability.
2. Feedback Page: Participants rate six NASA-TLX dimensions using sliders. Once done, the data are uploaded to Firebase. Designed as a secondary task in the Rigor Cycle, it ensures low cognitive interruption and real-time feedback collection.
3. Success Page: After submission, users receive a confirmation message and are returned to the home page. Error handling is also included. Enhanced during the Evaluation Cycle, this page was refined to enhance reliability and system stability.

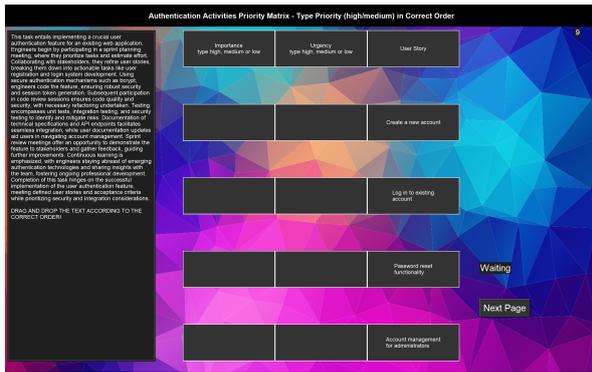
The complete source code and all developed data collection tools' interfaces are available on GitHub for access and further reference.¹

5.3. Comparative Analysis of MWL Tools

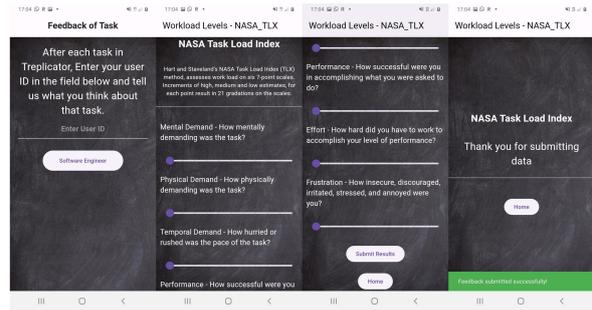
Based on the latest literature and research findings, Table 6 compares traditional and modern MWL assessment tools in terms of accuracy, usability, real-world applicability, and suitability for software engineering tasks. The final row shows the integrated system of this research (TSDA + SRMA), which uniquely combines subjective (NASA-TLX) and objective (EEG) data within realistic task simulations to address these gaps.

¹The complete implementation can be found at:

- TSDA Development: <https://github.com/Sameera-G/TreplicatorEEG>
- SRMA Development: https://github.com/Sameera-G/nasa_tlx_eeg_research



(a) Page for activities related to Refinement of user stories



(b) Pages of SRMA to mark and upload self reported MWL values

Figure 3: User interfaces of TSDA (a) and SRMA (b).

Table 6
Comparison of Mental Workload Assessment Tools for Software Engineers

Method	Measurement Type	Accuracy	Usability	Real-World Use	Fit for SW Engineers
NASA-TLX (Hart & Staveland, 1988)	Subjective (Self-report)	Moderate (Bias)	High (Easy)	Moderate	Limited (Subjective)
SWAT	Subjective	Moderate	Moderate	Moderate	Limited
Physiological (HR, GSR, Pupil)	Physiological	High	Low (Gear Needed)	High (Real-Time)	Partial (Not Load-Specific)
EEG-Based MWL	Neurophysiological	Very High	Moderate	High (Objective)	Strong (Cognitive-Specific)
This Tool (TSDA + SRMA)	Combined (Obj. + Subj.)	Very High	High	Very High	Strong (Built for SW Engineers)

6. Discussion

During the development process of the two applications, TSDA and SRMA, we followed a structured and iterative approach using the DSR method. Both tools were developed based on identified needs in MWL research in software engineering and refined using participant feedback. Figure 4 shows the tools developed in use during the experiment.



Figure 4: TSDA and SRMA in synchronized usage during EEG recording

The TSDA was developed to simulate real software engineering tasks such as prioritizing user stories, code review, and testing, reflecting real sprint workflows. It records task accuracy and completion time, which are automatically uploaded to Firebase, helping to evaluate mental load at each stage. The SRMA app supports TSDA by collecting NASA-TLX ratings after each task. Together, the tools sync EEG signals, performance data, and self-reported workload. We found that tasks with more errors and longer durations in TSDA often had higher ratings in SRMA, showing strong consistency between objective and subjective data.

The entire data pipeline, including TSDA, SRMA, and EEG, performed reliably. Firebase recorded all entries without data loss and the task logs matched the EEG and self-report timestamps, confirming the stability of the system.

6.1. Comparison with Previous Methods

Traditional MWL assessment tools such as NASA-TLX and SWAT rely only on self-reports. While they are easy to use, they often suffer from user bias and memory-related errors. Other physiological methods like heart rate variability and skin response are more objective but are not specific to mental effort and can be affected by unrelated stressors.

EEG is more specific to cognitive load, but past studies often used it in isolation. Our tools combine EEG with NASA-TLX in a synchronized way and link both to realistic task activities. This addresses a gap in earlier research. For example, recent work by Nakasai et al. (2024) used nasal skin temperature but did not connect it to software task structure. Our tools are specifically built for software engineers and provide task-by-task feedback using both physiological and subjective indicators. User feedback helped us improve both TSDA and SRMA. Participants found the interfaces easy to use, but they suggested adding more realistic tasks and smoother transitions. Based on this, we added visual dashboards, enhanced task complexity, and confirmed task data upload after each session. We also added error handling and success confirmation in SRMA to ensure a smooth user experience.

Real-time validation between EEG and NASA-TLX helped identify inconsistencies. For instance, if a participant reported low workload after a difficult task that showed high EEG activity, the system flagged it for further review. This adds reliability to the research and reduces the influence of random responses.

6.2. Real-World Application and Limitations

The tools have strong real-world potential for software teams to detect when developers are under high cognitive load and adjust task assignments accordingly. This can help reduce burnout and improve productivity. However, limitations still exist. TSDA simulates tasks in a controlled lab setting, which may not include real-world distractions like team communication, meetings, or interruptions. Also, some participants may still misunderstand NASA-TLX scales despite instructions. These limitations will be addressed in future iterations.

6.3. Future Research and Integration with Machine Learning

Although this phase focused on tool development, future work will use machine learning to analyze patterns in EEG and SRMA data. Deep learning models can be trained to recognize high workload states in real time and suggest break times or workload rebalancing automatically. This could lead to intelligent systems that adapt to individual cognitive states and improve mental well-being in the software development process.

7. Ethical Considerations & Limitations

The research was conducted in accordance with ethical principles for research involving professionals following the ethical guideline of the Ethics Committee of the Human Sciences Ethics Committee of

the University of Oulu. All participants were informed about the purpose of the research and gave their voluntary consent. The research addressed key ethical and methodological considerations. The privacy of the participant was protected by avoiding the collection of personal data and ensuring informed consent. Although the simulated tasks reflected real software engineering workflows, certain workplace factors such as distractions and collaboration were not fully replicated. All data handling was in accordance with GDPR and ethical research guidelines.

8. Conclusion

This research focused on developing and validating two custom-built tools, the TSDA and the SRMA, to help assess MWL. DSR methodology, the research aimed to provide a reliable method that captures both physiological data from EEG signals and subjective self-reported ratings using NASA-TLX. The overall goal was to create a synchronized system for real-time MWL measurement that captures the cognitive demands of software development tasks.

The main objectives were to define the technical requirements, study the link between EEG signals and self-reports, build synchronized applications, and evaluate them through user testing. These goals were achieved through iterative development and both qualitative and quantitative testing.

TSDA simulated real-world software tasks, tracking accuracy, and completion time, while SRMA collected NASA-TLX ratings after each task. The tools were connected through Firebase for synchronized data collection. The tests showed a strong consistency between the EEG and self-reported data, which confirms the reliability of the dual source system. User feedback led to improvements in usability, task realism, and data flow stability. This research provides a validated system for measuring MWL using physiological and subjective data in a real-world context. Future work will focus on using machine learning to automate workload classification and support smart workload management, aiming to boost developer well-being and performance.

Acknowledgments

The authors extend their gratitude to all responders, supervisors, and experts who participated in the usability tests and feedback sessions. The authors sincerely thank Anastasiia Voitenko for her contribution to the research, participation in the testing process, and kind permission to include her photograph in the publication.

Declaration on Generative AI

During the preparation of this work, the author(s) used CHATGPT-4 to check grammar and spelling. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the content of publication.

References

- [1] S. Munoz, O. Araque, J. F. Sánchez-Rada, C. A. Iglesias, An emotion aware task automation architecture based on semantic technologies for smart offices, *Sensors* 18 (2018) 1499. doi:10.3390/s18051499.
- [2] F. Kamari Ghanavati, A. Choobineh, S. Keshavarzi, A. A. Nasihatkon, A. S. Jafari Roodbandi, Assessment of mental workload and its association with work ability in control room operators, *La Medicina del Lavoro* 110 (2019) 389–397. doi:10.23749/mdl.v110i5.8115.
- [3] Y. Ding, Y. Cao, V. G. Duffy, Y. Wang, X. Zhang, Measurement and identification of mental workload during simulated computer tasks with multimodal methods and machine learning, *Ergonomics* 63 (2020) 896–908. doi:10.1080/00140139.2020.1759699.

- [4] I. Teoh Yi Zhe, P. Keikhosrokiani, Knowledge workers mental workload prediction using optimised elanfis, *Applied Intelligence* 51 (2021) 2406–2430. doi:10.1007/s10489-020-01928-5.
- [5] P. Keikhosrokiani, M. Isomursu, O. Korhonen, T. T. Sean, Intelligent mental workload mobile application in personalized digital care pathway for lifestyle chronic disease, in: M. Särestöniemi, et al. (Eds.), *Digital Health and Wireless Solutions. NCDHWS 2024*, volume 2083 of *Communications in Computer and Information Science*, Springer, Cham, 2024. URL: https://doi.org/10.1007/978-3-031-59080-1_24. doi:10.1007/978-3-031-59080-1_24.
- [6] N. Tang, M. Chen, Z. Ning, A. Bansal, Developer behaviors in validating and repairing llm-generated code using ide and eye tracking, in: *2024 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, IEEE, 2024, pp. 40–46. doi:10.1109/VL/HCC60511.2024.00015.
- [7] K. Nakasai, S. Komeda, M. Tsunoda, M. Kashima, Measuring mental workload of software developers based on nasal skin temperature, *IEICE TRANSACTIONS on Information and Systems* 107 (2024) 1444–1448.
- [8] S. Astuti, P. Nurtantio, K. K.S., Exploring the impact of technostress on millennial work-life balance in digital work: The mediating role of work well-being, *IEEE Transactions on Engineering Management* (2024). doi:10.1109/10762733.2024.
- [9] D. Schott, M. Kunz, F. Heinrich, J. Mandel, Stand alone or stay together: An in-situ experiment of mixed-reality applications in embryonic anatomy education, *ACM Transactions on Virtual Reality Software and Technology* (2024). doi:10.1145/3641825.3687706.
- [10] I. L. Dourado, L. C. Moreira, M. Kauffman, B. Horan, A wearable ar system for maintenance in industry 5.0: Assessing mental workload and usability, in: *2024 29th International Conference on Automation and Computing (ICAC)*, 2024, pp. 1–6. doi:10.1109/ICAC61394.2024.10718838.
- [11] S. G. Hart, L. E. Staveland, Development of nasa-tlx (task load index): Results of empirical and theoretical research, in: P. A. Hancock, N. Meshkati (Eds.), *Human Mental Workload*, volume 52 of *Advances in Psychology*, North-Holland, 1988, pp. 139–183. URL: <https://www.sciencedirect.com/science/article/pii/S0166411508623869>. doi:[https://doi.org/10.1016/S0166-4115\(08\)62386-9](https://doi.org/10.1016/S0166-4115(08)62386-9).
- [12] S. Rubio, E. Díaz, J. Martín, J. M. Puente, Evaluation of subjective mental workload: A comparison of swat, nasa-tlx, and workload profile methods, *Applied Psychology* 53 (2004) 61–86. doi:<https://doi.org/10.1111/j.1464-0597.2004.00161.x>.
- [13] T. Kosch, J. Karolus, J. Zagermann, H. Reiterer, A. Schmidt, P. W. Woźniak, A survey on measuring cognitive workload in human-computer interaction, *ACM Comput. Surv.* 55 (2023). URL: <https://doi.org/10.1145/3582272>. doi:10.1145/3582272.
- [14] R. Widiastuti, V. R. B. Kurniawan, Kusmendar, E. Nurhayati, R. A. Putra, Implementation of the cardiovascular load and rating scale mental effort to reduce the bakery worker’s workload, *AIP Conference Proceedings* 2590 (2023) 040004. URL: <https://doi.org/10.1063/5.0106674>. doi:10.1063/5.0106674.
- [15] H. Mansikka, K. Virtanen, D. H. and, Comparison of nasa-tlx scale, modified cooper–harper scale and mean inter-beat interval as measures of pilot mental workload during simulated flight tasks, *Ergonomics* 62 (2019) 246–254. URL: <https://doi.org/10.1080/00140139.2018.1471159>. doi:10.1080/00140139.2018.1471159, PMID: 29708054.
- [16] A. R. Hevner, S. T. March, J. Park, S. Ram, Design science in information systems research, *MIS Quarterly* 28 (2004) 75–105. URL: <https://doi.org/10.2307/25148625>. doi:10.2307/25148625.
- [17] K. Peffers, T. Tuunanen, M. A. Rothenberger, S. Chatterjee, A design science research methodology for information systems research, *Journal of Management Information Systems* 24 (2007) 45–77. URL: <https://doi.org/10.2753/MIS0742-1222240302>. doi:10.2753/MIS0742-1222240302.
- [18] V. de Castro, M. L. Martín-Peña, E. M. Martínez, M. Salgado, Combining action research with design science as a qualitative research methodology: An application to service (operations) management research, *International Journal of Qualitative Methods* 24 (2025). URL: <https://doi.org/10.1177/16094069241312018>. doi:10.1177/16094069241312018.