ARTICLE

QNEX: An Intuitive Platform for Visualizing and Analyzing Quantum Noise in Quantum Circuits

J. Maas[1,*]

[1] Fontys University of Applied Sciences, Eindhoven 5651 GW, Netherlands

Abstract

Quantum computing offers the potential to solve problems beyond the capabilities of classical systems. However, the presence of quantum noise stemming from environmental factors, hardware imperfections, and decoherence remains a significant challenge to the reliability and performance of quantum systems. Despite advancements like Microsoft's improvements in logical error rates, quantum noise continues to hinder progress. This paper introduces QNEX, an interactive quantum noise visualization dashboard designed to help students, educators, and researchers better understand quantum noise. QNEX enables users to configure noise parameters and observe their effects in real-time. Usability testing with participants of varying expertise demonstrated that QNEX improves the understanding of quantum noise, although it still requires foundational knowledge of quantum concepts for effective use.

Keywords: quantum noise, quantum algorithms, bell state, grover's algorithm, visualization techniques, qiskit, quantumsim, quantum circuits, noise simulation, quantum error analysis, quantum computing

Citation

Maas, J. (2024). Quantum Noise Explorer (QNEX). An Intuitive Platform for Visualizing and Analyzing Quantum Noise in Quantum Circuits, 1(1), 1–8.

1 Introduction

Quantum computers offer the potential to solve complex problems beyond the capabilities of classical computers [1]. However, the performance of today's quantum computers, which are well known as Noisy Intermediate-Scale Quantum (NISQ) computers, are hindered by quantum noise caused by environmental factors, hardware flaws, and decoherence, leading to errors in quantum states [2]. Despite promising research feats on speeding up quantum computers, the high noise levels in quantum computers remain one of the key challenges to achieve its full capabilities.

1.1 Recent Developments

Although quantum noise is still a major problem in quantum computing, recent developments in quantum noise are promising. Recently, Microsoft announced a critical breakthrough that advances the field of quantum computing by improving the logical error rate by 800x when compared to the error rate on corresponding physical qubits [3, 4, 5]. This allowed them to create four highly reliable logical qubits from only 30 physical qubits, where previously Google scientists speculated it would take 1000 physical qubits to create a single logical qubit [6].

1.2 Research Gaps

The field of quantum noise visualization is an emerging area of research, and thus existing research is relatively scarce. Most studies have primarily focussed on general aspects of quantum noise, such as its sources and effects on quantum systems. However, there is a notable gap in the literature regarding the visualization of quantum noise, specifically at the level of quantum algorithms. This deeper level of understanding is critical, as it directly influences the fidelity and performance of quantum computations.

Submitted: January 18, 2025

Vol. 1, **No.** 1, 2024.
***Corresponding author(s):**
✉J. Maas
jelle.maas@student.fontys.nl

1.3 Research Objectives

To address the identified research gap, we propose the Quantum Noise Explorer (QNEX). This tool aims to simplify the visualization of noise effects on quantum algorithms, making it easier for both students and researchers to understand the effects of noise. These insights can aid in developing more robust and error-tolerant quantum algorithms for the future. In this paper, it is investigated how the effects of quantum noise on the performance of quantum algorithms can be effectively visualized, aiming to facilitate better understanding and evaluation.

1.4 Research Contributions

The major contributions of this research paper can be summarized as follows:

- Requirements for an interactive and intuitive user interface to analyse the effects of quantum noise on quantum circuits were developed in collaboration with stakeholders.
- QNEX is introduced to address the identified research gap, simplifying the visualization of noise effects on quantum algorithms and making it accessible for both students and researchers.
- Usability tests were conducted with users of varying levels of expertise in quantum computing to assess the effectiveness and usability of QNEX.
- Two case studies were performed to demonstrate the practical application and effectiveness of QNEX in real-world scenarios.

2 Related Work

There are numerous visualization which aid in the understanding of quantum mechanics as well as quantum computing [7]. This section reviews significant contributions to the field of visualization in quantum computing, highlighting various tools and methods developed to enhance understanding and accessibility across the entire field.

2.1 Visualization of Qubit State

Most visualizations in quantum computing focus specifically on representing the state of a single qubit. This is essential for understanding quantum phenomena such as superposition and entanglement.

- The **phase disk**, which visually depicts the probability of a qubit being in state $|1\rangle$.
- The **Bloch Sphere** represents a single qubit's state as a point on the unit sphere, with its coordinates reflecting the qubit's amplitude and phase in superposition.
- The **Q-sphere** extends the Bloch Sphere visualization to multiple qubits, representing each basis state as a node on a sphere.

Each visualization provides unique insights into the behaviour and manipulation of qubits within quantum systems.

2.2 Visualization for Quantum Circuits

The increasing complexity of quantum circuits, combined with the inherently abstract nature of quantum mechanics, presents challenges in understanding their structure and behaviour. Various (interactive) visualization tools have been developed to aid in this process.

One such tool is **QuFlow** [8], which visualizes the flow of parameters in quantum circuits. **GraphStateVis** [9] provides a detailed visual analysis of qubit graph states and their stabilizers. **Quantivine** [10] introduces a novel approach to visualizing quantum circuits by employing semantic analysis to enhance comprehension and readability. Traditional quantum circuit diagrams often face challenges with scalability and clarity as circuit complexity grows. Quantivine addresses these issues by integrating semantic structures, such as Abstract Syntax Tree (AST), into the visual representation of quantum circuits, making them more intuitive and easier to analyse.

2.3 Visualization for Quantum Noise

Existing studies related to quantum noise predominantly focus on mitigating errors in quantum computers [11], while fewer efforts have been dedicated to visualizing this noise. Despite this, some advancements have been made:

1. **VACSEN:** Provides a novel method for visualizing quantum noise [12]. VACSEN enables users to understand and manage the noise characteristics of different quantum computers and their compiled circuits.
2. **QVis:** QVis provides insights into spatial and temporal patterns of noise, offering tools to analyse similarities and variances in key performance metrics of quantum devices [13]. Additionally, it supports a detailed analysis of error properties on individual qubits, enabling a more granular understanding of qubit-specific performance.

Their approach serves as inspiration for the design of QNEX, integrating similar techniques to provide a more effective means of visualizing and analysing quantum noise, ensuring deeper insight into its effects on circuit results.

3 Background

This section provides essential background on quantum computing for this study, covering quantum information, qubits, superposition, entanglement, and quantum noise.

3.1 Quantum Information

Quantum information investigates how information can be stored and processed using quantum computers.

3.1.1 Quantum Bits

Qubits, or quantum bits, are the basic units of information in quantum computing [14]. Unlike classical bits, which are either 0 or 1, qubits can exist in a superposition of both 0 and 1. This means that a qubit's state is represented by the distribution of the probability that it is measured as 0 or 1, rather than being in a definite state until measured.

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \tag{1}$$

Where $|\psi\rangle$ is a superposition of the basis states $|0\rangle$ and $|1\rangle$, with α, β as complex coefficients that determine the probability amplitudes for $|0\rangle$ and $|1\rangle$, respectively. These coefficients must be constrained according to the second axiom of probability theory, which states [15].

$$|\alpha|^2 + |\beta|^2 = 1 \tag{2}$$

A key distinction between qubits and classical bits is the ability of qubits to exhibit correlations between their states, a phenomenon known as quantum entanglement. Entanglement ensures that the state of one qubit is intrinsically connected to the state of another, regardless of the distance separating them.

3.1.2 Quantum Gates

Quantum gates, or quantum logical gates, are fundamental operations that manipulate qubits in quantum computing. Similar to classical logic gates, which operate on bits (0 or 1), quantum gates transform qubit states. However, quantum gates operate on qubits' probabilities and superposition states, making them more powerful and versatile. Some examples of quantum gates are:

- **Pauli-X (NOT) Gate:** Flips the state of a qubit, changing $|0\rangle$ to $|1\rangle$ and vice versa.
- **Pauli-Z Gate:** Introduces a π-radian phase shift, effectively reflecting it around the z-axis.
- **Hadamard Gate (H Gate):** Places a qubit in a superposition, giving it equal probabilities of being in state $|0\rangle$ or $|1\rangle$
- **Controlled-NOT (CX) Gate:** A two-qubit gate that flips state of the target qubit if the control qubit is in state $|1\rangle$.
- **Controlled-Z (CZ) Gate:** A two-qubit gate that applies a phase flip to the target qubit if the control qubit is in state $|1\rangle$.

3.1.3 Quantum Circuits

Quantum circuits are sequences of quantum gates applied to qubits, designed to perform specific quantum computations. Each gate modifies the state of one or more qubits, and together they execute quantum algorithms. The circuit starts with qubits in an initial state (usually $|0\rangle$) and applies a series of gates to transform the qubits, followed by a measurement to obtain the output.

OpenQASM A way to describe quantum circuits and operations at a low level is by using a language called Quantum Assembly Language (QASM) [16, 17]. It is often used in conjunction with quantum computers or simulators such as Qiskit [18], which supports QASM as a way to execute quantum circuits on real quantum processors.

3.2 Noise in Quantum Computing

Quantum noise is a major challenge in creating scalable, fault-tolerant quantum computers. Techniques such as quantum error correction and noise mitigation are being developed to address this issue [19]. The performance and accuracy of quantum circuits are primarily affected by two types of noise: quantum noise and classical noise.

Classical Noise Refers to disturbances that arise from environmental factors and imperfections in control systems. Unlike quantum noise, which is a result of inherent quantum effects, classical noise is typically caused by external sources [20]. Different types of classical noise include:

- **Thermal noise:** Caused by temperature fluctuations affecting qubits, particularly in superconductors. Insufficient isolation can lead to relaxation errors, causing premature returns to the ground state.
- **Electromagnetic interference:** External fields can disrupt qubit states in systems like trapped ions or superconductors, leading to unintended phase shifts or qubit flips.
- **Initialization error:** Occurs when a qubit intended to be set to $|0\rangle$ is mistakenly initialized to $|1\rangle$, affecting computation accuracy.

Fundamental Quantum Noise Intrinsic to the nature of quantum mechanics and refers to the random fluctuations and uncertainties that arise from phenomena like superposition and entanglement [19]. Different types of quantum noise include:

- **Decoherence:** The loss of quantum coherence, where qubits lose their ability to maintain superpositions or entanglement.
- **Bit-flip and phase-flip errors:** Specific types of noise where the state of a qubit changes from 0 to 1 or vice versa (bit-flip) or the phase of the qubit's superposition state shifts (phase-flip).
- **Measurement errors:** Errors that occur when measuring the final state of qubits.

While quantum noise is inherent to the system, classical noise can often be reduced with better hardware and environmental controls.

3.3 Quantum Simulators

Quantum simulators are powerful tools that enable researchers and developers to explore and experiment with quantum computing in a controlled, software-based environment [21]. They provide valuable insights into quantum algorithms by offering a glimpse into the interactions between quantum states and operations, which is otherwise difficult to observe with physical quantum hardware due to state collapse. This allows for a deeper understanding of how quantum computations function. Simulators relevant to this study include Qiskit's [18] simulator called Aer from IBM, Cirq [22] from Google, and QuantumSim [23] from Fontys.

Modelling Quantum Noise Quantum noise can be modelled using various noise functions [24]. While different simulation platforms implement these functions with slight variations and under different names, their application and usage remain consistent. A non-exhaustive list of common noise functions includes:

- **State Preparation Error**: Occurs at initialization, where the qubit's state is initialized as $|1\rangle$ instead of $|0\rangle$. Modelled by the **initialization error probabilities** p_0, representing the chances of an incorrect initialization.
- **Pauli Noise**: Models random applications of Pauli gates (X, Y, Z), causing bit-flips, phase-flips, or both. Configured using probabilities for each operator (e.g., p_X, p_Y, p_Z).

Table 1. Noise Model Compatibility Matrix for Qiskit, Cirq, and QuantumSim. Symbols: ✓(Fully supported), $\sim$ (Partially Supported), $\times$ (Not Supported)

Noise Function	Qiskit	Cirq	QuantumSim
State Preparation Error	$\sim$	$\sim$	✓
Pauli	✓	✓	✓
Amplitude Damping	✓	✓	✓
Phase Damping	✓	✓	✓
Depolarizing	✓	✓	✓
Reset	✓	$\sim$	$\times$
Thermal Relaxation	✓	✓	✓
Readout Error	✓	✓	✓

- **Amplitude Damping**: Represents energy loss where a qubit decays from $|1\rangle$ to $|0\rangle$ due to environmental interaction. Governed by the **T1 relaxation time**, indicating the rate of decay.
- **Phase Damping**: Causes a loss of phase coherence without affecting population. Characterized by the **T2 coherence time**, which determines how long phase coherence is maintained before dephasing occurs.
- **Depolarizing Noise**: Randomly applies a Pauli error (X, Y, or Z) with a probability, leading to a mixed state. Controlled by the **depolarization probability** p, which defines the likelihood of any error occurring.
- **Reset Noise**: Forcibly resets a qubit to the $|0\rangle$ state, often due to measurement or other operations. Configured by the **reset probability** p_{reset}.
- **Thermal Relaxation**: Models both population decay and dephasing in qubits due to thermal effects. Described by **T1** (relaxation time) and **T2** (coherence time).
- **Readout Error**: Occurs during measurement, where the qubit's state is read incorrectly. Modelled by the **readout error probabilities** p_0, representing the chances of an incorrect measurement outcome.

Table 1 provides a compatibility overview of these noise functions across relevant quantum simulators.

Where ✓means that the noise function is fully implemented in the respective framework. The $\sim$ means that while the noise function is not directly implemented, its behaviour can be replicated with additional steps. The $\times$ means that the noise function is not supported.

3.3.1 Quantifying Quantum Noise In Simulation

An effective method to quantify the amount of quantum noise in simulation is by comparing ideal and noisy simulation results. This can be done by measuring the closeness between ideal and noisy quantum states using the fidelity function given by [25, 26]:

$$F(\rho,\sigma) = \left(\mathrm{Tr}\sqrt{\sqrt{\rho}\sigma\sqrt{\rho}}\right)^2 \tag{3}$$

Where ρ and σ are density matrices and Tr is the trace. Fidelity ranges from 0 to 1, with 1 indicating identical states and 0 indicating orthogonal states.

4 Methodology

This chapter outlines the methodology used in the research and development process of QNEX.

4.1 Research Design

This research includes four main components: design phase, prototype development, usability testing, and two case studies. Each phase is outlined in a separate section of this paper.

Informing the Design The design of QNEX was informed by background research, stakeholder discussions, and insights from existing quantum computing tools. The goal was to create an intuitive interface with a left-to-right flow, building upon the design of existing quantum circuit designer tools. This approach ensured the platform was accessible to users with varying levels of experience in quantum computing, making it both a valuable learning tool for students and educators, and a practical tool for researchers.

QNEX Based on the initial design specification, a functional prototype of QNEX was developed. The prototype incorporated key features, including a QASM input, quantum circuit diagram visualization, a noise modelling editor, and various visualizations, all within a user-friendly interface. The prototype was designed to be adaptable to various types of simulators, as well as facilitating the easy incorporation of new noise modelling functions.

Usability Testing To assess the effectiveness, user experience, and usability of QNEX, a usability test was conducted with users of varying levels of experience in quantum computing using the functional prototype. Feedback was collected through Likert scale [27] statements, observations, and questionnaires. Based on the feedback from usability testing, iterative adjustments were made to improve usability, functionality, and overall user satisfaction.

Case Studies To assess the effectiveness and practical applicability of QNEX, two case studies were conducted. The first case study evaluated the impact of noise on a basic example using the Bell state circuit, while the second explored a more complex example with Grover's algorithm. The results from both case studies were compared with prior research to validate accuracy.

4.2 Ethical Considerations

The development and evaluation of QNEX incorporated several ethical considerations to ensure responsible research practices and equitable access:

Accessibility The accessibility of QNEX was aimed at accommodating users with varying degrees of expertise in quantum computing, promoting inclusion in quantum instruction and research. Special attention was given to creating an intuitive interface to reduce barriers for beginners while maintaining functionality for advanced users.

Open Source The source code for QNEX is publicly available on GitHub [28], which ensures transparency, reproducibility, and collaboration within the quantum computing research domain. This aligns with ethical principles in research to ensure that advancements in quantum computing are shared and accessible to a global community.

Usability Testing During usability testing, ethical principles were carefully followed. Participants received clear and detailed information about the study's purpose, the tasks involved, and how their feedback would be utilized. Informed consent was obtained from all participants, including specific permission for recording their screens and/or audio when necessary.

5 Informing The Design

This section outlines the key requirements and inspirations that guided the design of QNEX, alongside the user journey used.

5.1 Requirements

The initial set of requirements was drafted based on previously discussed background knowledge. These requirements were subsequently expanded, prioritized using the MoSCoW-method [29], and validated through discussions with relevant stakeholders.

R1 **Must: Reading and parsing OpenQASM files.** Provides real-time support for loading and visualizing quantum circuits from OpenQASM text.

R2 **Must: Support different types of simulators.** Allows users to choose from various simulators, like Qiskit [18] Aer and Cirq [22] Simulator, for running quantum circuits.

R3 **Must: Visualization of quantum noise across the circuit.** Shows how quantum noise affects different parts of the circuit, providing insight into noisy versus ideal results.

R4 **Must: Side-by-side visualizations with different noise conditions.** Enables users to compare different noise conditions side-by-side, such as ideal vs. noisy results, for better analysis.

R5 **Should: Interacting with the visualizations through the quantum circuit diagram.** Allows users to interact with the circuit visualization by selecting elements such as gates for configuring noise.

R6 **Should: Saving visualizations to different image formats.** Provides functionality to save circuit and noise visualizations as images in various formats for sharing or documentation.

R7 Could: Multi-level quantum circuits. Supports multi-level circuits with nested sub-circuits, allowing for easy navigation and exploration of complex designs.

Furthermore, a key technical requirement from stakeholders was that the system be primarily developed using Python.

R8 Must: Python programming language. The entirety of the software system is developed using the Python programming language as much as possible.

This is because professionals in the field are typically familiar with Python, making QNEX easier to maintain and understand if the code requires updates or clarification.

5.2 Inspiration

The main inspiration from QNEX was drawn from IBM Composer [30], which allows users to build, visualize, and simulate quantum circuits, featuring an intuitive drag-and-drop interface for making circuits and a simulator for observing circuit outputs.

5.3 User Journey

QNEX builds upon the user journey of quantum circuit designer tools like IBM Composer [30]. These tools generate QASM code [16, 17], which serves as the initial step in the user journey within QNEX:

A Inputting QASM Code Users start by entering or pasting OpenQASM code into the left panel, often exported from tools like IBM Quantum Composer.

B Configuring Simulation Parameters After loading the circuit, users adjust parameters in the right panel, including backend selection, execution count, and noise settings (bit-flip, phase-flip, thermal relaxation, etc.)

C Executing the Simulation Users run the quantum circuit with selected settings and noise parameters.

D Visualizing Simulation Results Results are shown in the central panel as interactive visualizations like probability distributions and Q-spheres, illustrating circuit behaviour under noise for further analysis.

This flow from A to D ensures a logical and user-friendly progression that supports both beginners and experienced users in exploring quantum circuit behaviour.

6 QNEX

To address the identified research gap, we propose the QNEX. QNEX, shown in Figure 1, is an interactive quantum noise visualization dashboard proposed to bridge the gap in providing effective visualizations to show the impact of quantum noise, improving the understanding of the effects of noise. The source code of QNEX can be accessed on GitHub via the URL: https://github.com/typiqally/qnex.

6.1 Panels

QNEX is organized into several panels: the QASM panel (A), visualization panel (B), and simulation panel (C).

A — QASM Panel The QASM panel, located on the left of QNEX, enables users to input or import QASM code. This code can be generated using editors such as IBM Composer [30] or Quantum Circuit Simulator from the University of Stuttgart [31]. The panel supports both versions 2.0 [16] and 3.0 [17] of the OpenQASM specification. Once the QASM code is entered, it is validated and then loaded in real-time, displaying the corresponding quantum circuit diagram in the visualization.

B — Visualization Panel The visualization panel, located in the middle of QNEX, allows users to explore several key data visualizations. It currently includes the following:

- **Quantum State Fidelity (B_1):** Visualizes the fidelity of the quantum state after simulation, allowing users to assess how closely the noisy output matches the ideal state.
- **Quantum Circuit Diagram (B_2):** Displays a graphical representation of the quantum circuit based on the loaded QASM code.
- **Measurement Counts (B_3):** Provides a histogram of measurement outcomes from the circuit's execution, helping users analyse the distribution of results.
- **Measurement Probabilities Per Shot (B_4):** Displays a shot-by-shot breakdown of measurement probabilities, offering detailed insights into how noise impacts each run of the circuit.

C — Simulation Panel The simulation panel, located on the right side of QNEX, allows users to simulate the loaded quantum circuit while configuring and applying various noise conditions. Users start by selecting a supported backend, such as Qiskit, labelled as C_1 in Fig. 1. They can then configure noise by choosing a predefined model (C_2), which represents profiles based on real quantum processors. Alternatively, users can select the "custom" option to create personalized models by configuring noise parameters at the gate level, accessible via C_3. Finally, users can initiate the simulation by clicking the run button, labelled as C_4.

Table 2. List of scenarios used during the usability test.

#	Scenario
S1	Input and Circuit Visualization
S2	Selecting a Noise Model Profile
S3	Visualization
S4	Adjusting Execution Iterations
S5	Visualizing Different State Vectors
S6	Visualizing Different Iterations
S7	Modifying the Noise Profile
S8	Saving the Noise Profile
S9	Deleting the Noise Profile

Table 3. List of questions and statements used in the feedback questionnaire.

#	Question
Q1	What aspects of QNEX were most helpful or enjoyable?
Q2	Did QNEX improve your understanding of quantum noise effects? If so, how?
Q3	Were there challenges or difficulties while completing the tasks?
Q4	Were the visualizations clear and informative? Was any information missing?
Q5	Were any interface aspects unintuitive or confusing?
Q6	Suggestions for improving usability, functionality, or clarity of visualizations?
Q7	Was the interaction smooth and responsive? If not, what issues arose?
Q8	Did QNEX meet expectations in modelling and visualizing quantum noise effects? Why or why not?
ST1	I would use QNEX in the future for analysing quantum noise effects.
ST2	I would recommend QNEX to colleagues or students working in quantum computing.

7 Usability Testing

To further evaluate the effectiveness and usability of QNEX, an in-depth usability study was conducted with participants representing the target audience.

7.1 Test Design

The usability test was designed using an explorative testing approach [32]. In this approach, participants are allowed to brainstorm, share opinions, and express emotional impressions about ideas and concepts.

Participants Seven participants with varying levels of expertise in quantum computing were selected. Specifically, **U1-2** have no prior experience with quantum computing, **U3-6** are familiar with the subject but lack extensive experience, and **U7** is active in quantum mechanics and slightly in quantum computing.

Environment and Set up The usability test was conducted in a controlled environment using a laptop with preloaded instructions and the functional prototype of QNEX discussed in the previous section. An external high-definition display was connected to provide a clear view of QNEX running in a web browser.

Scenarios Participants completed 9 scenarios shown in Table 2, each containing a set of tasks designed to evaluate various features of QNEX. Participants were encouraged to explore QNEX freely and, after each scenario, rated their agreement with a set of statements using a 5-point Likert scale. Additional feedback was collected through an open-ended section at the end of the questionnaire.

Questionnaire A post-test questionnaire with open-ended questions shown in Table 3 was administered to gather more detailed insights and suggestions for improvement.

Figure 1. The QNEX interface enables users to analyse the impact of noise on quantum circuits under specified conditions through three main panels (A-C). The **QASM Panel (A)** allows users to input or import QASM code, which automatically loads the quantum circuit into the system. The **Visualization Panel (B)** provides various visualizations, including quantum state fidelity, circuit diagrams, counts, and probabilities, enabling comparisons between ideal and noisy results. Finally, the **Simulation Panel (C)** lets users configure the simulation backend, design custom noise models or select from pre-existing ones, and run simulations for a specified number of iterations.

Table 4. Usability testing results for participants interacting with QNEX, rated on a Likert scale from 1 to 5, where 1 indicates strong disagreement and 5 indicates strong agreement.

S	Statement	Mean	Std
1	Integration between IBM Composer and QNEX	4.14	0.99
	Quantum circuit diagram loaded as expected	4.57	0.73
2	Selecting a profile was clear	3.86	0.99
	Noise parameters clearly describe	3.00	1.07
3	Selecting a state vector	3.86	1.12
	Visualizations were useful in understanding	4.29	1.03
	Visualizations behaved as expected	3.86	0.99
	Easy to distinguish the influence of noise	4.43	1.05
	Comparison between ideal and noisy	4.14	0.83
4	Adjusting the number of executions/shots	4.57	0.49
5	Switching between different state vectors	4.43	0.73
	Visualizations updated as expected	4.57	0.49
	Correspondence between each state vector	2.57	1.59
	Switching state vectors enhanced understanding	4.43	0.49
6	Switching between different iterations was clear	4.29	0.70
	The visualizations clearly reflected changes	4.57	0.49
7	Adjusting the noise parameters was intuitive	4.57	0.49
	Noise parameter changes met expectations	3.86	0.83
8	The process of saving a profile was clear	4.43	0.90
	Reloading a saved profile and applying was smooth	3.86	1.46
9	The process of deleting a profile was clear	4.14	1.12

7.2 Results

The usability test results, presented in Table 4, are summarized for each scenario based on participant feedback. As a guideline, a scenario is deemed sufficient if the mean score for all relevant statements is greater than or equal to **4.0** and if the standard deviation indicates consistent responses (less than or equal to **1.0**).

Scenario 1 — Input and Circuit Visualization Most participants found the integration between IBM Composer and QNEX intuitive ($rating_{\mathrm{mean}} = 4.14, rating_{\mathrm{sd}} = 0.99$). Suggestions included adding an option to upload/import QASM files and a hyperlink to IBM Composer for easier navigation. The circuit diagram was clear and accurate for all participants ($rating_{\mathrm{mean}} = 4.57, rating_{\mathrm{sd}} = 0.73$).

Scenario 2 — Selecting a Noise Model Profile The process of selecting a noise profile was clear for most participants, but selecting a specific gate within the noise profile was occasionally confusing ($rating_{\mathrm{mean}} = 3.86, rating_{\mathrm{sd}} = 0.99$). Several participants attempted to click directly on the circuit diagram to adjust the noise profile of a specific gate instead of using the dropdown, indicating that this interaction may be beneficial.

Some participants found the noise parameters unclear, especially those with less experience in quantum computing ($rating_{\mathrm{mean}} = 3.00, rating_{\mathrm{sd}} = 1.07$). Suggestions included adding tooltips and additional explanations for each individual noise parameter to improve clarity.

Scenario 3 — Visualization Running the simulation and selecting a state vector were confusing for some users ($rating_{\mathrm{mean}} = 3.86, rating_{\mathrm{sd}} = 1.12$) due to a lack of feedback on simulation completion and missing visualizations. Suggestions included providing run feedback and defaulting to the first state vector for visualization. Most participants found the visualizations useful ($rating_{\mathrm{mean}} = 4.29, rating_{\mathrm{sd}} = 1.03$) and easy to distinguish noise ($rating_{\mathrm{mean}} = 4.43, rating_{\mathrm{sd}} = 1.05$), though some requested more detailed descriptions. Distinguishing between ideal and noisy conditions was mostly easy ($rating_{\mathrm{mean}} = 4.14, rating_{\mathrm{sd}} = 0.83$), though a colour-blind participant suggested using patterns for better visual distinction.

Scenario 4 — Adjusting Execution Iterations The process of adjusting the different iterations was clear for all participants ($rating_{\mathrm{mean}} = 4.57, rating_{\mathrm{sd}} = 0.49$). However, one participant noted that when the iterations were set to zero, the visualizations would disappear, resulting in an error. This indicates the need for better input validation.

Scenario 5 — Visualizing Different State Vectors The process of switching between state vectors was clear for most participants ($rating_{\mathrm{mean}} = 4.43, rating_{\mathrm{sd}} = 0.73$). However, the correspondence between the state vector and different parts of the quantum circuit was less clear ($rating_{\mathrm{mean}} = 2.57, rating_{\mathrm{sd}} = 1.59$). Suggestions included adding a visual marker in the quantum circuit diagram to indicate which step of the quantum circuit is being visualized, and using the names of quantum operations for the state vectors to better relate them to the executed quantum circuit.

Scenario 6 — Visualizing Different Iterations Switching between different iterations was clear for most participants ($rating_{\mathrm{mean}} = 4.29, rating_{\mathrm{sd}} = 0.70$), and most found the visualizations to clearly reflect changes between different executions ($rating_{\mathrm{mean}} = 4.57, rating_{\mathrm{sd}} = 0.49$). However, two participants were confused due to a wording issue in the instructions. Suggestions included updating the title of the visualizations to clearly indicate which iteration is being viewed and how the visualization is influenced by the parameter.

Scenario 7 — Modifying the Noise Profile The process of adjusting the noise profile was intuitive and straightforward for most participants ($rating_{\text{mean}} = 4.57, rating_{\text{sd}} = 0.49$). One participant suggested adding an input box next to the sliders to enable more precise input of percentages. Additionally, **U7** proposed allowing the input of infinity, as it is common for physicists to explore how behaviour changes when influenced by an infinite amount. Lastly, all participants noted that they would need a deeper understanding of quantum noise to evaluate whether the impact of the noise on the quantum circuit met their expectations ($rating_{\text{mean}} = 3.86, rating_{\text{sd}} = 0.83$).

Scenario 8 — Saving the Noise Profile The process of saving a noise profile was clear for most participants ($rating_{\text{mean}} = 4.43, rating_{\text{sd}} = 0.90$). Reloading the saved profile worked for most ($rating_{\text{mean}} = 3.86, rating_{\text{sd}} = 1.46$), but some encountered exceptions due to a misconfigured noise parameter. Suggestions included clarifying where the noise profile is saved and preventing exceptions on reload.

Scenario 9 — Deleting the Noise Profile The process of deleting a noise profile was generally clear for most participants ($rating_{\text{mean}} = 4.14, rating_{\text{sd}} = 1.12$). However, currently, the name of the deleted profile remains in the profile name field used for saving, which confused some participants, leading them to believe the profile had not been deleted. Suggestions included improving deletion feedback, by clearing the profile name field, using a popup modal or alert for confirmation.

7.3 Adjustments

Based on user testing feedback, several adjustments were made to the functional prototype to enhance usability and accessibility:

- **IBM Composer Link**: A direct link to IBM Composer was added at the QASM input field for easier navigation.
- **Selected State Vector**: A visual indicator was integrated into the quantum circuit diagram to strengthen the connection with the selected state vector, using descriptive names for state vectors.
- **Tooltips and Descriptions**: The interface was made more beginner-friendly with tooltips providing clear descriptions for each noise parameter and other components such as the QASM input field.
- **Saving and Deleting Profiles**: The handling of profiles shifted to client-side storage, making their management more intuitive and facilitating easier deployment.
- **Colour-blind Friendly Graphs**: Bar charts were enhanced with patterns for differentiation, and a colour-blind friendly colour scheme was applied to the fidelity graph.
- **Randomization Seed**: An option for defining a custom randomization seed was added for easier result reproduction.
- **Explicit Noise Parameter Input**: A number input field was introduced alongside noise parameter sliders for precise value entry.
- **Run Feedback**: Enhanced feedback with an animated loading icon, and the first state vector is now selected by default for immediate visualizations post-simulation.
- **Error Handling**: Improved mechanisms provide clearer and more actionable error feedback.
- **Font Size and Panel Titles**: Font sizes were increased for readability, and panel titles were added to help users navigate sections efficiently.

8 Case Studies

Two case studies were conducted on popular types of quantum circuits, namely the Bell state circuit and Grover's algorithm. These case studies explore the impact of quantum noise on these circuits, and compare the results to existing research to validate the accuracy and effectiveness of QNEX. The Qiskit backend was used to load the quantum circuits and simulate them using Qiskit Aer.

8.1 Case Study I — Bell State Circuit

In this case study, the impact of noise is analysed on Bell states using QNEX. In quantum information science, Bell's states are specific quantum states of two qubits that represent the simplest example of quantum entanglement [26]. While there are many ways to create entangled Bell states through quantum circuits, the simplest takes a computational basis as input and contains a Hadamard and CNOT gate, as depicted in Figure 2.

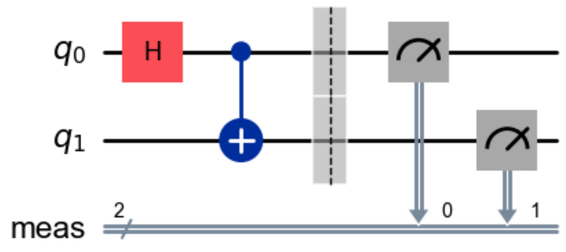

Figure 2. Quantum circuit diagram for the Bell State Circuit used in Case Study I

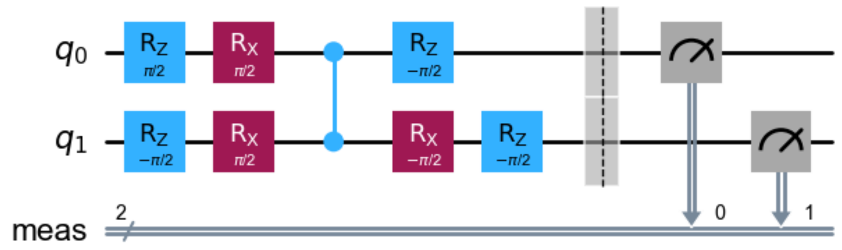

Figure 3. Quantum circuit diagram for the decomposed version of the Bell State Circuit for Rigetti hardware used in Case Study I

To compare results with existing findings, this case study replicates the approach outlined in an AWS Quantum Technology blog on noise in quantum computing [19]. The blog begins by decomposing the Bell state into hardware-native gates compatible with Rigetti [33], a quantum hardware provider, which results in the quantum circuit shown in Figure 3.

A noise model similar to that in the blog post is used, where depolarizing noise is applied to the **RX** and **RZ** gates equally ($P_{q_0} = 0.595...\%, P_{q_1} = 0.047...\%$), and the CZ ($p = 7.92...\%$) gate, along with a readout error ($P_{q_0} = 2.1...\%, P_{q_1} = 5.0...\%$). These noise probabilities are derived from the noise profile of the Rigetti Aspen M2 quantum processor. This presents one notable design limitation of QNEX, which is its inability to apply noise to specific qubits; instead, noise for each gate is uniformly applied across all qubits. To temporarily bypass this limitation, we chose the average of the probabilities of depolarizing noise on RX and RZ ($P = 0.321...\%$), and readout error ($P = 3.55...\%$) for both qubits.

The circuit was executed with *1000* shots, like in the blog post. The measurement counts from circuit simulation result, as shown in Figure 4, align with those shown in the blog post, confirming that the applied noise model effectively replicates the expected outcomes.

8.2 Case Study II — Grover's Algorithm

Originally proposed by Lov Grover in 1996 [34], **Grover's algorithm**, also known as the quantum search algorithm, is designed for unstructured searches. It efficiently identifies the unique input to a black-box function, or **oracle**, that yields a specified output with high probability. The detailed implementation of Grover's algorithm is beyond the scope of this case study.

This case study analyses the impact of quantum noise on the reliability of Grover's algorithm. Results from Kumar et al. [35] are replicated covering a noise analysis of Grover's algorithm. In this research, Kumar et al. analysed the effects of noise on the 2 to 5-qubit versions of Grover's algorithm.

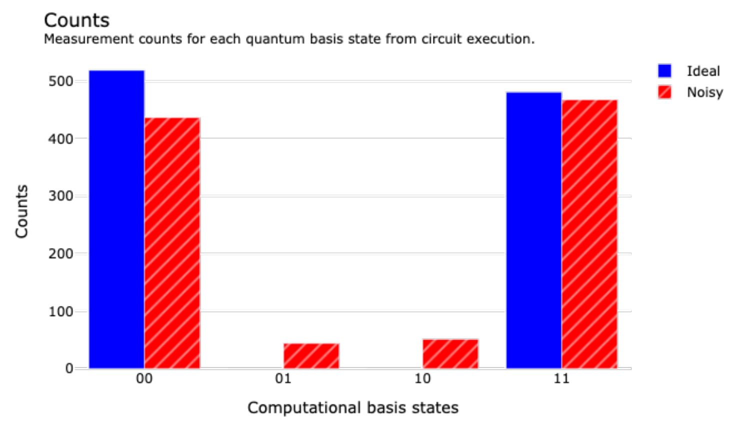

Figure 4. Measurement counts for each quantum basis state from Bell state circuit execution using the noise parameters in Case Study I

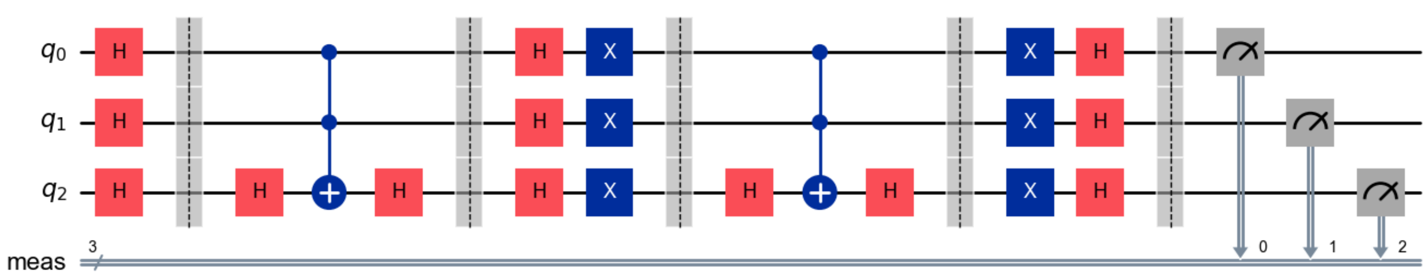

Figure 5. Quantum circuit diagram for the 3-qubit version of Grover's Algorithm with $|111\rangle$ as the marked state used in Case Study II

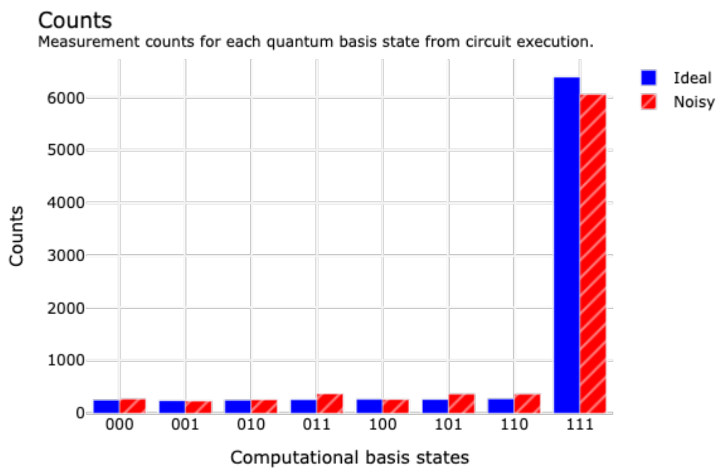

Figure 6. Measurement counts for each quantum basis state from 3-qubit Grover's algorithm execution using noise parameters from IBM-Q Santiago in Case Study II

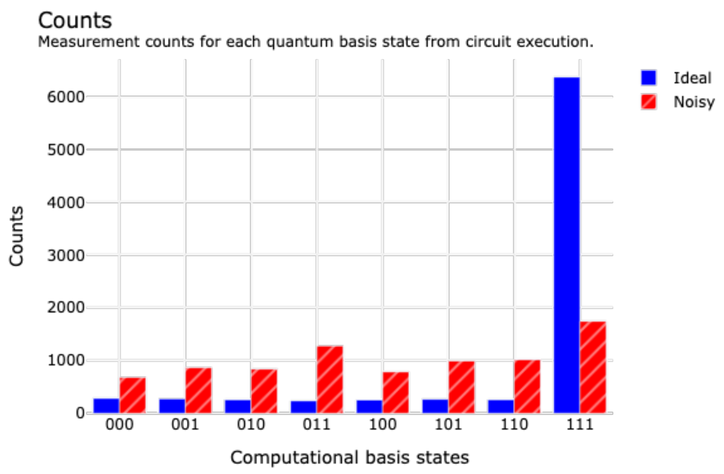

Figure 7. Measurement counts for each quantum basis state from 3-qubit Grover's algorithm execution using custom noise parameters in Case Study II

In this case study, the 3-qubit version of Grover's algorithm is replicated based on the noise analysis and is shown in Figure 5. This circuit has $|111\rangle$ set as the marked state, and applies 2 iterations, which is consistent with the approach in the noise analysis paper. The noise parameters used in the paper, including Pauli errors (rotation errors), dephasing (decay), amplitude damping (decoherence), depolarization, and thermal relaxation, are configured using calibration data from **IBM-Q Santiago**.

In the research paper, the circuit was executed with *8192* shots. However, the measurement counts from the circuit simulation results, as shown in Figure 6, do not align with those presented in the research paper. The tests in the research paper appear to be significantly more affected by noise compared to the results from our tests.

Increasing the probability of bit flip ($p = 15\%$) and depolarization ($p = 5\%$) and enabling thermal relaxation on the Hadamard gate, leads to results that align more closely with those found in the research paper. In this scenario, Grover's algorithm becomes highly unreliable, with all results being heavily influenced by noise.

It is possible that a different, older version of IBM-Q Santiago's calibration data is being used, which experienced greater noise levels. But this remains uncertain at the moment.

9 Discussion

This section summarizes key insights gained during the development of QNEX, followed by a discussion of its limitations and potential areas for future work and research.

9.1 Lessons

Throughout the design and implementation of QNEX, several important lessons were learned, which are highlighted below.

Quantum computing remains a complex topic to grasp One of the primary objectives of QNEX was to facilitate the understanding of quantum computing, particularly for beginners such as students, by providing a clear visualization of quantum noise. The usability test indicated that while QNEX enhances the comprehension of quantum noise for individuals with limited experience, it continues to be challenging for those unfamiliar with quantum concepts, requiring a foundational understanding to effectively utilize the tool.

Balancing complexity and accessibility Findings from usability testing highlight the importance of balancing comprehensive features with an intuitive design to serve both beginner and advanced users. These insights are consistent with those from VACSEN [12], which emphasizes the significance of intuitive visual designs in quantum computing for quantum computing users.

9.2 Limitations and Future Research

The usability test demonstrated that QNEX is an effective and intuitive tool for visualizing quantum algorithms under varying noise conditions. However, certain limitations remain.

Scalability Issues QNEX performs well with small quantum circuits of up to 5 qubits and limited depth. However, as circuit size increases, the visualizations face scalability challenges, and performance gradually decreases. Future work involves adopting a multi-level approach, limiting the number of qubits displayed simultaneously while preserving meaningful insights. Furthermore, developing novel visualization techniques and integrating methods from **Quantivine** [10] could enhance the visualization of large quantum circuit diagrams, improving usability and scalability. While performance challenges are inherent to current simulation capabilities, perceived performance could be improved by integrating features such as loading icons and progress bars.

Qubit-specific Noise The first case study on the Bell state circuit highlighted an opportunity for improvement in QNEX's functionality. Currently, noise is applied uniformly across all qubits, which may not fully capture the diverse nature of noise in real-world scenarios. Addressing this could enhance the tool's ability to simulate quantum circuits more realistically, offering even more accurate insights. Future work will focus on implementing the ability to apply noise to specific qubits, while ensuring a clear and intuitive user interface is maintained.

Interactive Quantum Circuit Diagram Although the functional prototype did not incorporate an interactive quantum circuit diagram, findings from the usability test indicated that this feature could provide substantial value. Future efforts will focus on developing a custom quantum circuit diagram renderer to enable this interaction.

10 Conclusions

In conclusion, we have introduced QNEX, an interactive quantum noise visualization dashboard designed to enhance the understanding of quantum noise. The design was guided by established visual and technical requirements. Usability testing has shown that QNEX effectively supports users with varying levels of expertise in understanding quantum noise. However, it still presents challenges for those without a foundational understanding of quantum concepts. These challenges have been addressed with various improvements aimed at enhancing the system's usability and accessibility.

Looking ahead, future work will focus on improving the scalability of QNEX through novel visualizations and methods inspired by **Quantivine** [10], enabling more effective noise visualization in circuits with deeper layers and larger qubit counts. Additionally, the interactive quantum circuit diagram, identified as a valuable feature in usability tests, will be further developed and tested to improve overall user experience.

Acknowledgement

This project was conducted in collaboration with Fontys University of Applied Sciences, ICT. The participants in our user interviews and anonymous reviewers are thanked for their feedback.

References

[1] A. Valavanidis, "Quantum computing. a revolutionary computing capability to sift through huge numbers of possibilities and extract potential solutions to complex problems," vol. 1, pp. 1–25, 01 2024.

[2] Z. Pan, Y. Feng, Z. Li, Y. Liu, and Y. Li, "Understanding the impact of quantum noise on quantum programs," pp. 426–437, 2023.

[3] A. Boyle, "Microsoft and quantinuum report a way to turn down the noise in quantum computing," April 2024.

[4] M. Swayne, "Microsoft-led team achieves record for reliable logical qubits in quantum computing," September 2024.

[5] D. Tom and K. Svore, "How microsoft and quantinuum achieved reliable quantum computing," August 2024.

[6] P. Smith-Goodson and P. Moorhead, "Microsoft and quantinuum improve quantum error rates by 800x," May 2024.

[7] E. W. Bethel, M. G. Amankwah, J. Balewski, R. Van Beeumen, D. Camps, D. Huang, and T. Perciano, "Quantum computing and visualization: A disruptive technological change ahead," *IEEE Computer Graphics and Applications*, vol. 43, no. 6, pp. 101–111, 2023.

[8] S. Lin, J. Hao, and L. Sun, "Quflow: Visualizing parameter flow in quantum circuits for understanding quantum computation," in *2018 IEEE Scientific Visualization Conference (SciVis)*, pp. 37–41, 2018.

[9] M. Miller and D. Miller, "GraphStateVis: Interactive Visual Analysis of Qubit Graph States and their Stabilizer Groups," in *IEEE International Conference on Quantum Computing and Engineering (QCE)*, pp. 378–384, IEEE, 2021.

[10] Z. Wen, Y. Liu, S. Tan, J. Chen, M. Zhu, D. Han, J. Yin, M. Xu, and W. Chen, "Quantivine: A visualization approach for large-scale quantum circuit representation and analysis," *IEEE Transactions on Visualization and Computer Graphics*, p. 1–11, 2023.

[11] Z. Cai, R. Babbush, S. C. Benjamin, S. Endo, W. J. Huggins, Y. Li, J. R. McClean, and T. E. O'Brien, "Quantum error mitigation," *Reviews of Modern Physics*, vol. 95, Dec. 2023.

[12] S. Ruan, Y. Wang, W. Jiang, Y. Mao, and Q. Guan, "Vacsen: A visualization approach for noise awareness in quantum computing," July 2022.

[13] J. Chae, C. A. Steed, and T. S. Humble, "Visual analytics of performance of quantum computing systems and circuit optimization," in *2024 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, (Los Alamitos, CA, USA), pp. 613–618, IEEE Computer Society, jul 2024.

[14] B. Schumacher, "Quantum coding," *Phys. Rev. A*, vol. 51, pp. 2738–2747, Apr 1995.

[15] C. P. Williams, *Explorations in quantum computing*. Springer Science Business Media, 12 2010.

[16] A. W. Cross, L. S. Bishop, J. A. Smolin, and J. M. Gambetta, "Open quantum assembly language," 2017.

[17] A. Cross, A. Javadi-Abhari, T. Alexander, N. De Beaudrap, L. S. Bishop, S. Heidel, C. A. Ryan, P. Sivarajah, J. Smolin, J. M. Gambetta, and B. R. Johnson, "Openqasm 3: A broader and deeper quantum assembly language," *ACM Transactions on Quantum Computing*, vol. 3, p. 1–50, Sept. 2022.

[18] A. Javadi-Abhari, M. Treinish, K. Krsulich, C. J. Wood, J. Lishman, J. Gacon, S. Martiel, P. D. Nation, L. S. Bishop, A. W. Cross, B. R. Johnson, and J. M. Gambetta, "Quantum computing with Qiskit," 2024.

[19] J. Moreno, G. Salton, and T. Chen, "Noise in quantum computing," September 2022. Accessed: 2024-10-13.

[20] K. Georgopoulos, C. Emary, and P. Zuliani, "Modeling and simulating the noisy behavior of near-term quantum computers," *Physical Review A*, vol. 104, Dec. 2021.

[21] T. H. Johnson, S. R. Clark, and D. Jaksch, "What is a quantum simulator?," *EPJ Quantum Technology*, vol. 1, 7 2014.

[22] C. Developers, "Cirq," May 2024.

[23] N. Kuijpers, "GitHub - QuantumSim," 12 2024.

[24] E. Onorati, T. Kohler, and T. S. Cubitt, "Fitting quantum noise models to tomography data," *Quantum*, vol. 7, p. 1197, Dec. 2023.

[25] R. Jozsa, "Fidelity for Mixed Quantum States," *Journal of Modern Optics*, vol. 41, pp. 2315–2323, 12 1994.

[26] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*. Cambridge University Press, 10 2000.

[27] K. Finstad, "The usability metric for user experience," *Interacting with Computers*, vol. 22, pp. 323–327, 09 2010.

[28] J. Maas, "GitHub - Quantum Noise Explorer," 1 2025.

[29] D. Clegg and R. Barker, *CASE Method fast-track*. Addison Wesley Longman, 1 1994.

[30] IBM, "IBM Quantum Composer," 2023.

[31] U. of Stuttgart, "Quantum Circuit Simulator," 3 2023.

[32] U. D. of Homeland Security, "Approaches to Usability Testing," 6 2024.

[33] R. Computing, "Quantum computing," 12 2024.

[34] L. K. Grover, "A fast quantum mechanical algorithm for database search," *STOC*, pp. 212–219, 1 1996.

[35] T. Kumar, D. Kumar, and G. Singh, "Noise Analysis of Grover's Quantum Search Algorithm," *Indian Journal of Pure Applied Physics*, 1 2023.

Maas, J.(Masters Student, Fontys) received the BASc degree in HBO-ICT from Fontys University of Applied Sciences, Eindhoven 5651 GW, Netherlands, in 2024. (Email: jelle.maas@student.fontys.nl)