

## Речь для защиты дипломной работы

Добрый день, уважаемые члены аттестационной комиссии!

Меня зовут Курнаев Данила Владимирович, и я представляю вашему вниманию дипломную работу на тему: «Разработка программного обеспечения для эргономической оценки кабины самолета с использованием методик PVT и NASA-TLX».

### **Слайд 2: Актуальность**

В условиях роста требований к безопасности полетов и эффективности работы пилотов особую актуальность приобретает задача объективной оценки эргономических характеристик кабины самолета. Эргономическая оценка с использованием современных методик, таких как PVT и NASA-TLX, позволяет выявить слабые места во взаимодействии человека и машины, минимизировать влияние человеческого фактора и снизить вероятность ошибок. Разработка системы, обеспечивающей автоматизацию такого анализа, актуальна для повышения надежности авиационных систем и улучшения условий работы экипажа в условиях высокой когнитивной нагрузки.

### **Слайд 3: Цель работы**

Целью дипломной работы является разработка программного обеспечения для проведения эргономической оценки кабины самолета. ПО реализует тесты PVT для анализа скорости реакции и психомоторной бдительности пилотов, а также методику NASA-TLX для оценки субъективной рабочей нагрузки. Система обеспечивает автоматизацию сбора данных, их обработку и визуализацию результатов, предоставляя удобный инструмент для анализа когнитивной и физической нагрузки на пилотов.

## Слайд 4: Постановка задач

Для достижения поставленной цели я выделил следующие задачи:

- Сформировать требования к системе.
- Выбор стека технологий для реализации ПО.
- Разработать ключевые модули системы.
- Создать графический пользовательский интерфейс.
- Обеспечить сохранение и управление данными.
- Провести интеграцию разработанной системы со стендом.
- Провести тестирование разработанного приложения, подтверждающее корректность его работы.

## Слайд 5: Используемые методы оценки эргономики

В своей работе я использовал две основные методики оценки эргономики:

- **PVT (Psychomotor Vigilance Task):** метод измеряет скорость реакции оператора на визуальные стимулы, что позволяет оценить его психомоторную бдительность и усталость.
- **NASA-TLX (Task Load Index):** методика субъективной оценки рабочей нагрузки, которая включает шкалы умственной и физической нагрузки, временного давления и других факторов.

Про каждый из методов и детали их реализации в системе будет рассказано позже (на таких-то слайдах).

## Слайд 6: Архитектура системы

Приложение построено на основе архитектурного паттерна MVC (Model-View-Controller), который обеспечивает чёткое разделение обязанностей между слоями системы. Это упрощает разработку, тестирование и поддержку, а также делает систему гибкой для дальнейшего развития.

1. **Слой данных (Model):** отвечает за управление данными, включая их обработку, сохранение и валидацию. Использует локальную базу данных SQLite и ORM-библиотеку SQLAlchemy для работы с данными. Это обеспечивает удобное взаимодействие с хранилищем данных без прямой зависимости от базового SQL.
2. **Сервисный слой (Controller):** реализует основную бизнес-логику приложения. Этот слой обрабатывает входные данные, запускает тесты (PVT и NASA-TLX) и выполняет расчеты. Контроллер связывает данные с пользовательским интерфейсом, обеспечивая их корректную обработку и передачу.
3. **Слой представления (View):** предоставляет графический интерфейс, созданный с использованием библиотеки Tkinter. Интерфейс включает элементы для взаимодействия с пользователем: выбор тестов, отображение текущего состояния системы, визуализация результатов.

Основной принцип заключается в четком разделении ответственности между слоями, что повышает гибкость и устойчивость приложения.

## **Слайд 7: Разработка системы**

В ходе работы я использовал язык Python версии 3.10. Основной стек технологий включал:

- **Python 3.10:** Выбран за его универсальность, наличие обширной экосистемы библиотек и удобство для прототипирования.
- **SQLite:** Простая, надёжная реляционная база данных для хранения пользовательских данных и результатов тестов.
- **SQLAlchemy:** ORM для работы с базой данных, позволяющая писать код на Python без необходимости написания сложных SQL-запросов.
- **Tkinter:** Инструмент для создания графического интерфейса пользователя, который обеспечивает простоту разработки и кросс-платформенность.

- **Pydantic:** Используется для валидации данных, чтобы гарантировать корректность информации на входе и выходе приложения.
- **PyInstaller:** Сборщик приложения в единый исполняемый файл, позволяющий упаковать все зависимости и сделать программу готовой к запуску без необходимости установки Python.

### Архитектурные подходы:

- **MVC (Model-View-Controller):**
  - Реализовано чёткое разделение приложения на модель (база данных), представление (графический интерфейс) и контроллер (бизнес-логика).
  - Такое разделение улучшает поддержку и тестируемость системы.
- **Фасад:**
  - Применён в слое логики приложения для инкапсуляции сложных операций. Например, управление базой данных и выполнение валидации с помощью Pydantic организованы через унифицированные сервисы.

### Слайд 8: Интерфейс приложения

Интерфейс приложения включает несколько основных экранов:

- Главное меню для выбора тестов.
- Экран проведения тестов, где отображаются стимулы и фиксируются результаты.
- Экран с итогами тестирования, который предоставляет графики и таблицы для анализа данных.

Программа интуитивно понятна и удобна в использовании.

### Слайд 9: Реализация теста PVT

Тест PVT (Psychomotor Vigilance Test) используется для оценки скорости реакции пользователя на визуальные стимулы. Этот тест разработан для анализа когнитивных характеристик, таких как внимание и быстрота реакции, в различных условиях. В рамках моего приложения тест проводится дважды: до выполнения упражнения и после, что позволяет оценить влияние нагрузки на когнитивные способности пользователя.

### **Алгоритм работы теста PVT:**

#### **1. Подготовка к тесту:**

- Пользователь выбирает упражнение и задачу, к которой относится тест.
- Программа настраивает параметры тестирования, включая количество раундов и временные интервалы.

#### **2. Генерация стимулов:**

- Тест начинается с того, что на экране отображается сообщение "Ожидайте сигнал".
- Через случайный промежуток времени (2–5 секунд) появляется сигнал, слово "ЖМИ!".
- Этот подход исключает возможность предугадывания момента появления сигнала.

#### **3. Реакция пользователя:**

- Пользователь должен как можно быстрее нажать на клавишу при появлении сигнала.
- Программа фиксирует время реакции с точностью до миллисекунд.

#### **4. Фиксация и сохранение данных:**

- Результаты каждого раунда записываются в базу данных.
- Для каждого теста сохраняются следующие параметры:
  - Время реакции пользователя.
  - Упражнение и задача, к которой относится тест.

- Тип теста (до или после нагрузки).
- Дата и время выполнения.

#### **5. Анализ данных:**

- После завершения всех раундов программа рассчитывает:
  - Среднее время реакции.
  - Минимальное и максимальное время реакции.
- Эти показатели позволяют оценить стабильность и точность реакций.

#### **6. Сравнение результатов:**

- После выполнения упражнений пользователь повторяет тест.
- Сравнение результатов до и после нагрузки помогает определить влияние физических или когнитивных нагрузок на время реакции.

#### **Пример работы теста:**

- Если среднее время реакции до нагрузки составляет 250 мс, а после — 310 мс, это указывает на снижение концентрации и внимания из-за усталости.
- Минимальное время реакции может показать пиковую скорость реакции, а максимальное — моменты срывов внимания.

#### **Преимущества реализации:**

- Генерация случайных стимулов делает тестирование объективным и исключает адаптацию к сигналам.
- Сохранение результатов в базе данных позволяет анализировать их в будущем и строить графики для более глубокого анализа.
- Разделение тестов на этапы "до" и "после" обеспечивает возможность количественной оценки воздействия упражнений.

Тест PVT в данном приложении — это не только инструмент измерения времени реакции, но и средство для изучения влияния внешних факторов на когнитивные способности пользователя.

### **Слайд 10: Реализация NASA-TLX**

NASA-TLX включает субъективный ввод данных пользователем по шести шкалам. После завершения теста программа автоматически рассчитывает общий индекс нагрузки и визуализирует результаты в виде диаграммы.

### **Слайд 11: Тестирование системы**

Тест NASA-TLX (Task Load Index) разработан для субъективной оценки рабочей нагрузки, испытываемой пользователем при выполнении задачи. В моём приложении этот тест используется для анализа восприятия нагрузки после выполнения упражнений, что позволяет оценить изменения в умственной и физической нагрузке, а также уровень стресса.

### **Алгоритм работы теста NASA-TLX:**

#### **1. Сбор оценок по шкалам:**

- Пользователь вводит субъективные оценки по шести шкалам:
  - Умственная нагрузка.
  - Физическая нагрузка.
  - Временная нагрузка.
  - Уровень усилий.
  - Уровень стресса (фрустрация).
  - Восприятие эффективности выполнения задачи.
- Каждая шкала оценивается от 0 (минимальная нагрузка) до 20 (максимальная нагрузка).

## 2. Парные сравнения для определения весов:

После ввода оценок программа предлагает пользователю сравнить важность каждой пары шкал.

Например, пользователь выбирает, что для него более значимо: умственная нагрузка или временные ограничения.

На основе ответов программа рассчитывает веса каждой шкалы, что позволяет учитывать индивидуальные приоритеты пользователя.

$$Weighted TLX = \frac{\sum_{i=1}^n S_i W_i}{\sum_{i=1}^n W_i}$$

$S_i$  — оценка пользователя по  $i$ -й шкале (например, умственная нагрузка, физическая нагрузка и т.д.);

$W_i$  — вес  $i$ -й шкалы, полученный в результате парных сравнений;

$n$  — количество шкал (в данном случае  $n=6$ );

$\sum_{i=1}^n W_i$  — сумма всех весов для нормализации итогового показателя.

## 3. Сравнение результатов:

- Сравнение итоговых индексов нагрузки позволяет выявить, как нагрузка на пользователя изменилась под влиянием физического или умственного напряжения.

## 4. Пример работы теста:

- Если пользователь оценивает умственную нагрузку в 18 баллов и считает её самой важной, то итоговый взвешенный индекс нагрузки будет сильно зависеть от этой шкалы.
- В случае парного сравнения, где пользователь выбирает, что временная нагрузка важнее физической, временная нагрузка получит больший вес.



## **Слайд 14: Выводы по дипломной работе**

- Сформулированы требования к системе эргономической оценки кабины самолета;
- Проведен анализ существующих решений и обоснован выбор стека технологий для реализации тестов PVT и NASA-TLX;
- Разработаны ключевые модули системы, включая проведение тестов психомоторной бдительности (PVT) и субъективной оценки рабочей нагрузки (NASA-TLX);
- Создан графический пользовательский интерфейс (GUI), обеспечивающий интуитивно понятное взаимодействие с системой и визуализацию результатов;
- Проведено тестирование разработанной системы, подтвердившее корректность ее работы и соответствие поставленным требованиям.

## **Слайд 15: Заключение**

Таким образом, все поставленные задачи были успешно выполнены. Реализация данной системы в будущем позволит эффективно анализировать эргономические характеристики кабины самолета, снизить число ошибок, связанных с человеческим фактором, и повысить безопасность полетов, обеспечив оптимальные условия работы для пилотов в условиях высокой нагрузки.

Спасибо за внимание! Готов ответить на ваши вопросы.