УДК 004.89
https://doi.org/10.37661/1816-0301-2026-23-2-68-79

Поступила в редакцию | Received 10.03.2026
Подписана в печать | Accepted 06.04.2026
Опубликована | Published 30.06.2026

Разработка интеллектуальной системы планирования на основе персонального ассистента

А. В. Яскевич, В. А. Чуйко✉
✉E-mail: Vchuyko@bsu.by

Белорусский государственный университет,
пр. Независимости, 4, Минск, 220030, Беларусь

Аннотация

Цели. Целью исследования является разработка интеллектуальной системы планирования, функционирующей локально на оборудовании пользователя без передачи данных в Интернет.

Методы. Рассматривается проблема обеспечения конфиденциальности и автономности цифровых ассистентов, зависящих от облачной инфраструктуры. Предложена клиент-серверная архитектура, в которой серверная часть реализована с использованием фреймворка FastAPI и базы данных SQLite, а клиентский интерфейс разработан на языке JavaScript. Визуализация расписания и редактирование записей осуществляются через веб-интерфейс.

Результаты. Описан голосовой конвейер системы: для активации используется движок Porcupine, для транскрибации – модель Faster-Whisper с квантованием int8. Проведен сравнительный анализ технологического стека, обеспечивающего высокую точность распознавания речи. Разработан гибридный модуль понимания естественного языка. Реализована технология RAG, интегрирующая данные расписания в контекст генерации ответа. Для синтеза речи используется нейросеть Piper, выполнение которой через ONNX Runtime обеспечивает высокую скорость обработки. Разработан эвристический алгоритм жадного поиска для управления временными ресурсами.

Заключение. Сделан вывод о применимости разработанной системы в корпоративном секторе, где критически важны защита информации и функционирование в условиях закрытого сетевого контура.

Ключевые слова: интеллектуальная система, голосовой ассистент, нейронные сети, планирование задач, распознавание речи, естественный язык, графическая оболочка

Для цитирования. Яскевич, А. В. Разработка интеллектуальной системы планирования на основе персонального ассистента / А. В. Яскевич, В. А. Чуйко // Информатика. – 2026. − Т. 23, № 2. – С. 68–79. – https://doi.org/10.37661/1816-0301-2026-23-2-68-79.

Конфликт интересов. Авторы заявляют об отсутствии конфликта интересов.

Development of an intelligent scheduling system based on a personal assistant

Anton V. Yaskevich, Vladislav A. Chuyko✉
✉E-mail: Vchuyko@bsu.by

Belarusian State University,
av. Nezavisimosti, 4, Minsk, 220030, Belarus

Abstract

Objectives. The aim of the research is to develop an intelligent scheduling system operating locally on the user's equipment without data transmission via Internet.

Methods. This paper examines the problem of ensuring the privacy and autonomy of digital assistants dependent on cloud infrastructure. A client-server architecture is proposed, in which the server component is implemented using the FastAPI framework and an SQLite database, and the client interface is written in JavaScript. Schedule visualization and entry editing are performed through the web interface.

Results. The system's voice pipeline is described: the Porcupine engine is used for activation, and the Faster-Whisper model with int8 quantization is used for transcription. A comparative analysis of the technology stack, ensuring high speech recognition accuracy, is conducted. A hybrid natural language understanding module is developed. RAG technology is implemented, integrating schedule data into the response generation context. Speech synthesis is performed using the Piper neural network, whose execution through ONNX Runtime ensures high processing speed. A heuristic greedy search algorithm for managing time resources has been developed.

Conclusion. The developed system is considered applicable in the corporate sector, where information security and operation in closed network environments are critical.

Keywords: intelligent system, voice assistant, neural networks, task planning, speech recognition, natural language, graphical shell

For citation. Yaskevich A. V., Chuyko V. A. *Development of an intelligent scheduling system based on a personal assistant*. Informatika [*Informatics*], 2026, vol. 23, no. 2, pp. 68–79 (In Russ.). https://doi.org/10.37661/1816-0301-2026-23-2-68-79.

Conflict of interests. The authors declare of no conflict of interest.

Введение

Сегодня развитие информационных технологий направлено на автоматизацию рутинных процессов и повышение личной эффективности пользователя за счет предоставления инструментов, способных оптимизировать временны́е затраты и снизить когнитивную нагрузку. Одним из способов автоматизации рутинных процессов является использование интеллектуальных систем, представленных в виде персональных ассистентов, позволяющих решать задачи управления личным временем пользователя. Существующие коммерческие решения, такие как Google Assistant и Amazon Alexa, демонстрируют высокую эффективность в различных сферах деятельности человека: организации рабочего процесса, управлении умным домом, поиске и структурировании информации. Существенным ограничением подобных платформ остаются зависимость от облачной инфраструктуры и необходимость передачи пользовательских данных на удаленные серверы, что создает риски для конфиденциальности и безопасности персональной

информации. Возможным решением данной проблемы является разработка интеллектуальных систем, объединяющих голосовое управление и алгоритмическое планирование без обращения к облачным сервисам, что позволяет обеспечить приватность, автономность и безопасность цифровых персональных ассистентов.

Разработка интеллектуальной системы

Проектирование голосового интерфейса требует решения ряда задач, связанных с обработкой естественного языка и обеспечением высокой скорости отклика системы. Основным сценарием использования голосового ассистента является бесконтактное взаимодействие, позволяющее пользователю управлять расписанием встреч и мероприятий без отрыва от текущей деятельности.

Одним из ключевых требований при разработке голосового ассистента является обеспечение непрерывного фонового прослушивания для детекции ключевой фразы активации. Данный процесс должен осуществляться с минимальным энергопотреблением, исключать ложные срабатывания, а также обеспечивать функционирование системы на стандартном компьютерном оборудовании без необходимости использования специализированных серверных ускорителей.

Особенностью разрабатываемой системы является внедрение механизма автоматизации интеграции гибких задач в расписание. Планирование рабочего времени основывается на разделении задач на две категории. К первой категории относятся события (встречи, поездки), имеющие фиксированное время начала и окончания, ко второй категории – гибкие задачи, обладающие определенной длительностью и сроком выполнения, но не привязанные к конкретному моменту времени.

При разработке диалоговых систем выделяют три основных подхода к организации вычислений: облачный, локальный и гибридный. В рамках данной работы используется локальный подход.

Разработанная интеллектуальная система предназначена для планирования личного времени на базе персонального голосового ассистента. В основе системы лежит клиент-серверная архитектура, адаптированная для локального использования на оборудовании пользователя. Схема информационных потоков и взаимодействия компонентов изображена на рис. 1.

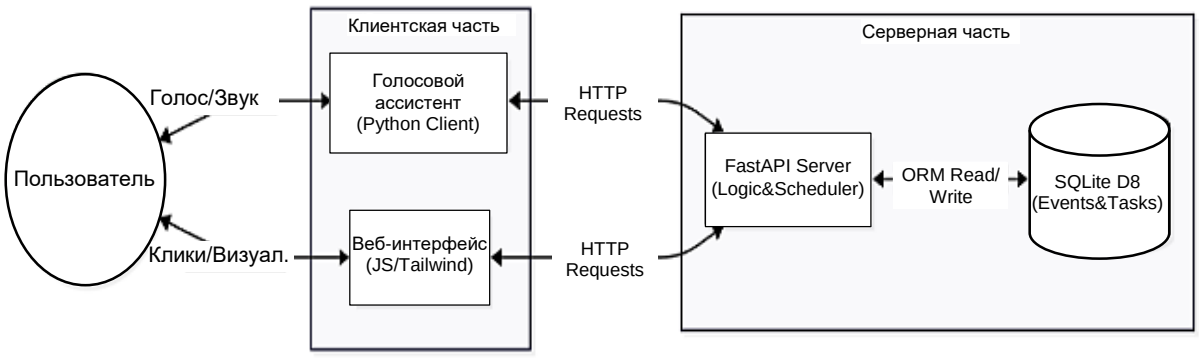

Рис. 1. Схема потоков данных между интерфейсами ввода и серверной частью
Fig. 1. Data flow diagram between input interfaces and the server part

Клиентская часть объединяет два программных модуля, отправляющих запросы на сервер. Первый модуль – голосовой ассистент, написанный на языке Python. Компонент отвечает за работу с микрофоном и динамиками, преобразуя речь в команды. Второй модуль – веб-интерфейс, созданный на базе языка JavaScript и фреймворка Tailwind CSS. Рассматриваемые инструменты служат для визуального отображения календаря и позволяют пользователю вручную редактировать записи, внесенные голосовым ассистентом. Несмотря на разные способы ввода, оба приложения работают по единому принципу: действия пользователя превращаются в стандартные HTTP-запросы и передаются на сервер.

Серверная часть отвечает за вычисления и хранение информации. Главным узлом обработки является фреймворк FastAPI. Компонент принимает входящие запросы от клиентов, проверяет корректность данных и запускает логику работы, включая алгоритмы планирования. Для хранения данных используется база данных SQLite. В ней содержатся таблицы для двух типов записей: фиксированных «Событий» и гибких «Задач». Связь между сервером и базой данных организована через механизм ORM. Сервер выполняет операции чтения и записи, работая с данными как с обычными программными объектами, без написания сложных SQL-команд.

Использование голосового интерфейса позволяет управлять ассистентом бесконтактно (голосом пользователя), не отрываясь от текущих задач. При этом локальное выполнение всей цепочки обработки – от детекции голоса до генерации ответа – решает задачу защиты персональной информации: аудиопоток и содержимое календаря не покидают устройство пользователя, что гарантирует приватность данных.

Голосовой конвейер на базе моделей Porcupine и Faster-Whisper

Преобразование акустического сигнала в текст является ресурсоемкой задачей, в связи с чем в разработанной системе реализован двухэтапный подход для минимизации нагрузки на центральный процессор и обеспечения высокой точности распознавания в условиях бытовых шумов [1–5]. Для реализации первого этапа – непрерывного мониторинга эфира и детекции ключевой фразы – используется движок Porcupine. Его выбор обоснован результатами сравнительного анализа с классическими решениями. В отличие от библиотек на базе скрытых марковских моделей, например CMU PocketSphinx, которые чувствительны к шумам и показывают частоту пропуска активации до 52 %, Porcupine базируется на глубоких нейронных сетях, обученных в режиме End-to-End. На рис. 2 продемонстрирована эффективность различных движков: сравнивается вероятность пропуска ключевой фразы (Wake Word Miss Rate) при фиксированном уровне чувствительности, допущение одного ложного срабатывания за 10 ч. Видно, что Porcupine демонстрирует наилучший результат с частотой пропуска всего 2,7 %. Для сравнения, классический PocketSphinx в тех же условиях пропускает более половины команд (52 %), а Snowboy – около трети (31,9 %). Это обосновывает выбор Porcupine как наиболее стабильного решения для зашумленной среды.

Помимо точности распознавания, рассмотренной выше, важным параметром для встраиваемых систем является вычислительная эффективность, которая напрямую влияет на энергопотребление устройства [6–9].

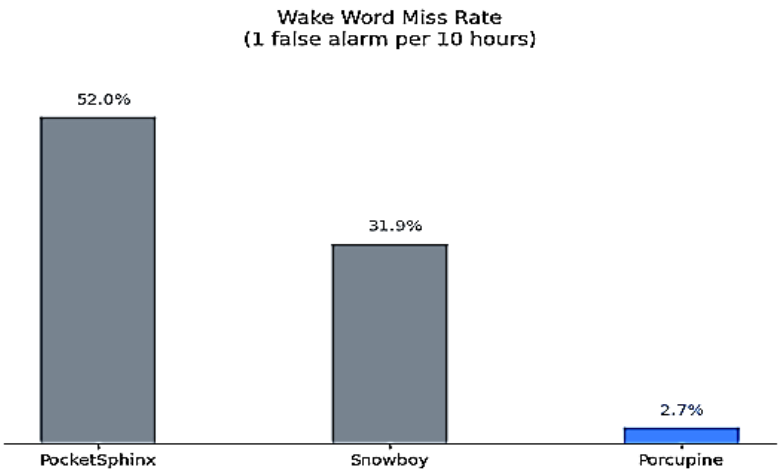

Рис. 2. Сравнительный график частоты пропуска активации при условии одного ложного срабатывания за 10 ч
Fig. 2. Comparative graph of the activation miss rate under the condition of one false alarm per 10 hours

Эксперименты по оцениванию производительности голосовых движков проводились на микрокомпьютере Rasberry Pi 5 под управлением 32-битной Rasberry Pi OS. Устройство оснащено процессором Broadcom BCM2712 (Quad-core ARM Cortex-A76, 2,4 ГГц).

На рис. 3 представлен сравнительный анализ нагрузки на центральный процессор при работе модулей детекции ключевой фразы в режиме непрерывного прослушивания. Тесты показали, что Porcupine является наиболее легковесным решением, потребляя менее 0,6 % ресурсов CPU. Это значительно эффективнее аналогов: Snowboy требует в шесть раз больше ресурсов (3,8 %), а PocketSphinx – более 12 %, что делает их менее пригодными для фонового мониторинга эфира в автономном режиме.

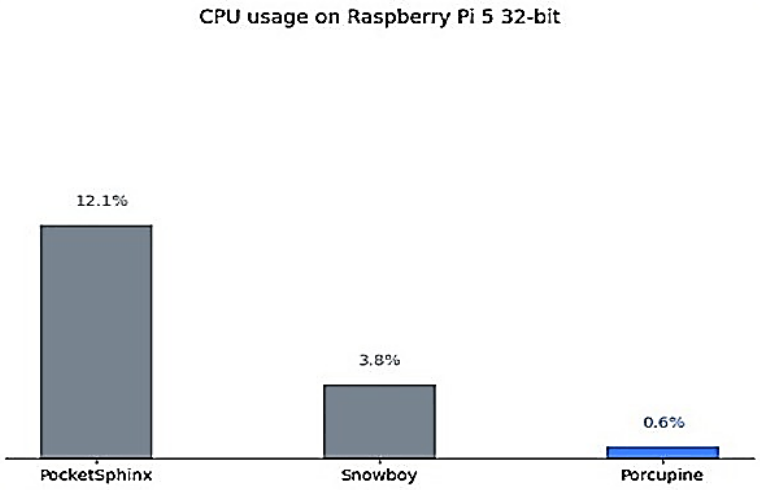

Рис. 3. График потребления ресурсов CPU на компьютере Raspberry Pi
Fig. 3. CPU resource consumption graph on a Raspberry Pi computer

Автоматическое распознавание речи

В качестве технологии выбрана модель Faster-Whisper с применением квантования весов до формата int8. Семейство моделей Vosk привязано к жесткому словарю и лишено механизма внимания, что приводит к высокому уровню ошибок (WER ~9,2–14,5 %). В то же время архитектура Transformer в Faster-Whisper позволяет эффективно использовать контекст фразы. Применение квантования CTranslate2 дает возможность сохранять точность транскрибации на уровне эталонных облачных сервисов (WER ~4,8 %), обеспечивая при этом высокую скорость обработки. Секунда речи обрабатывается за 0,5–0,6 с процессорного времени. Таким образом, сформированный стек технологий включает в себя:

Porcupine – для энергоэффективной активации по ключевому слову;

Faster-Whisper с квантованием int8 – для быстрого перевода речи в текст;

CTranslate2 – для оптимизации инференса модели на CPU.

Предложенная комбинация позволяет реализовывать полностью автономный и конфиденциальный голосовой интерфейс.

Обработка текстовых команд требует преобразования неструктурированной фразы в формализованный объект. Подход на базе регулярных выражений не использовался ввиду высокой вариативности естественного языка, а применение исключительно больших языковых моделей (Large Language Model, LLM) для каждого запроса требует высоких вычислительных затрат, что приводит к задержкам ответа системы. В разработанной системе реализован двухуровневый подход к анализу текста, позволяющий балансировать между скоростью и гибкостью.

На первом уровне в качестве основного компонента NLU-модуля используется модель BERT для первичной обработки запроса и классификации намерений. Ее применение обусловлено способностью преобразовывать запрос в векторное представление и соотносить с конкретными слотами (датой, временем, названием задачи) с высокой точностью [10–13]. Описанные возможности обеспечивают мгновенную реакцию системы на типовые команды управления календарем без привлечения тяжелых нейросетевых ресурсов.

Второй уровень активируется в тех случаях, когда запрос выходит за рамки стандартных сценариев или требует логического вывода. Для этого включается гибридный режим с привлечением LLM. По результатам сравнительного тестирования в качестве базовой модели выбрана Qwen 2.5-7B-Instruct. В отличие от моделей семейства Llama 3 и Mistral Qwen 2.5-7B-Instruct продемонстрировала высокую скорость генерации на центральном процессоре (8–10 токенов/с против трех-шести у аналогов) и более качественную поддержку русского языка.

Интеграция LLM реализована по методике RAG (Retrieval-Augmented Generation). Для обеспечения точности система в реальном времени формирует контекстное окно, выгружая из локальной базы данных SQLite актуальное расписание пользователя на ближайшие 48 ч. Данные внедряются в системный промпт, что позволяет модели выступать в роли интеллектуального консультанта: находить свободные окна, предлагать оптимальное время для переноса задач и отвечать на вопросы о занятости, сохраняя при этом полную конфиденциальность благодаря локальной реализации системы [14–16].

Эвристический алгоритм планирования

Для визуального управления расписанием разработан интерактивный веб-интерфейс в виде календаря (рис. 4), который служит средой для работы алгоритма автоматического планирования. Процесс автоматического распределения гибких задач описывается как последовательный рабочий цикл, объединяющий алгоритмические вычисления и пользовательский интерфейс. В основе этой функции лежит эвристический алгоритм жадного поиска.

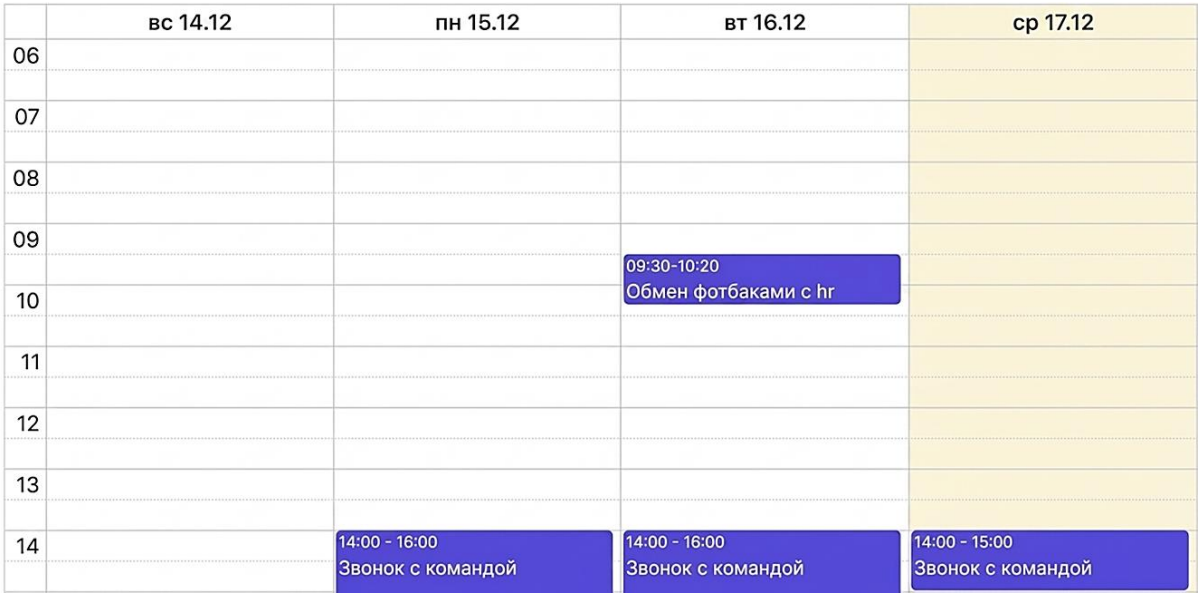

Рис. 4. Главный экран веб-интерфейса с отображением событий и задач
Fig. 4. The main screen of the web interface displaying events and tasks

Для определения оптимального временного интервала каждый потенциальный слот ранжируется с помощью весовой функции оценки:

$$Score = w_1 + P_{time} + w_2 + C_{compact} - w_3 L_{gap},$$

где P_{time} – метрика близости к желаемому времени начала задачи; $C_{compact}$ – критерий компактности, поощряющий размещение задач вплотную к уже существующим; L_{gap} – штраф за создание окон с интервалом менее 15 мин, которые невозможно эффективно использовать; w_1, w_2, w_3 – весовые коэффициенты, подобранные эмпирическим путем для баланса между пожеланиями пользователя и плотностью расписания.

Работа алгоритма начинается на этапе параметризации задачи, когда пользователь через графический интерфейс (рис. 5) задает название, длительность и уровень приоритета, при необходимости активируя опцию дробления задачи на части. После сохранения данные попадают в список ожидающих задач (рис. 6), где классифицируются как плавающие и ранжируются в очереди на основе вычислительного рейтинга приоритетности, учитывающего важность и близость дедлайна.

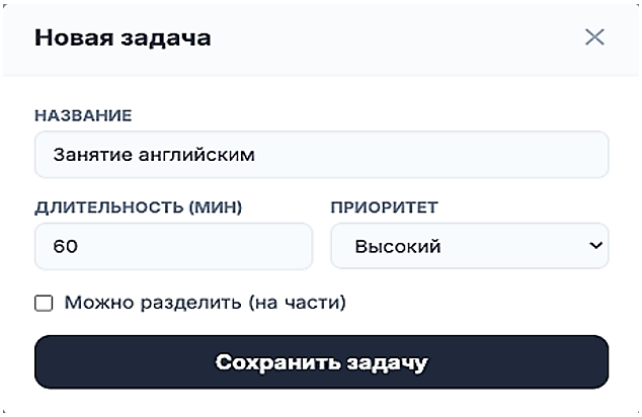

Рис. 5. Интерфейс создания гибкой задачи с параметрами для алгоритмического планирования

Fig. 5. Interface for creating a flexible task with parameters for algorithmic planning

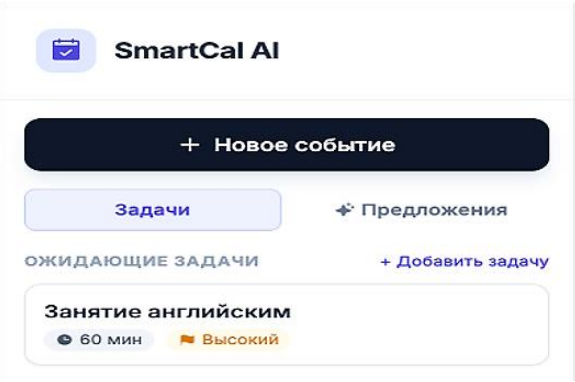

Рис. 6. Панель навигации и управления очередью нераспределенных задач

Fig. 6. Navigation and control panel for the queue of unassigned tasks

Процесс планирования инициируется командой «Составить расписание», после чего алгоритм выявляет свободные временные интервалы в сетке календаря, не занятые фиксированными событиями. Для каждого найденного слота вычисляется итоговый вес, который визуализируется в интерфейсе в виде конкретного значения рейтинга (рис. 7).

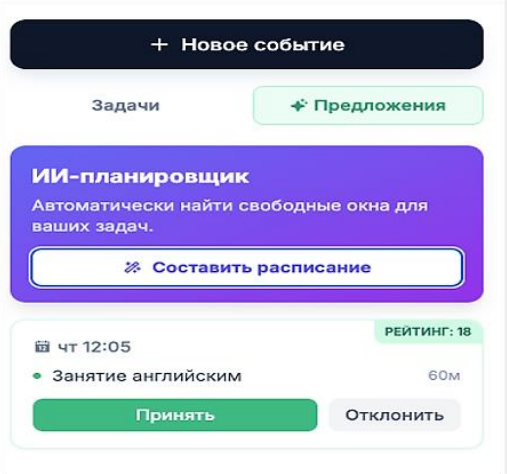

Рис. 7. Панель управления предложениями планировщика и механизм подтверждения задач

Fig. 7. Planner suggestion control panel and task confirmation mechanism

Сформированные результаты не вносятся в график немедленно, а направляются в специализированную буферную зону – вкладку «Предложения». Здесь пользователь имеет возможность провести предварительный просмотр предложенного времени (например, четверг 12:05) и верифицировать решение алгоритма, нажав кнопки «Принять» или «Отклонить». Описанный механизм подтверждения гарантирует полный контроль пользователя над расписанием и исключает риск нежелательных автоматических изменений в структуре личного времени.

Синтез речи

Завершающим этапом коммуникационного цикла является подсистема синтеза речи, при проектировании которой решалась задача выбора баланса между естественностью звучания и скоростью работы на центральном процессоре [16–20]. Сравнительный анализ в таблице показывает, что стандартные системные решения работают быстро, но не обеспечивают должного качества голоса, в то время как тяжелые генеративные модели (Bark, Tortoise) создают паузы в диалоге из-за высокого показателя RTF.

Сравнительная характеристика технологий синтеза речи
Comparative characteristics of speech synthesis technologies

Решение *Solution*	Нагрузка на CPU *CPU load*	Скорость (RTF)* *Speed (RTF)**
System (pyttsx3)	Минимальная	< 0,01
Bark / Tortoise	Экстремальная	> 2,5
Piper (VITS)	Средняя	0,05 – 0,1

В качестве компромисса выбран движок Piper на базе архитектуры VITS, объединяющей акустическую модель и вокодер в единую нейросеть. Использование среды исполнения ONNX Runtime и квантование моделей позволяют достичь показателя RTF на уровне 0,1, что означает генерацию 10-секундной фразы всего за 1 с. Такой подход минимизирует нагрузку на оперативную память и позволяет удерживать общую задержку системы в пределах комфортных для пользователя одной-двух секунд, обеспечивая высокую отзывчивость интерфейса в полностью автономном режиме.

Заключение

В результате выполнения работы спроектирована и программно реализована автономная система интеллектуального голосового управления. Тестирование подтвердило, что система работает корректно и успешно выполняет функции распознавания речи, планирование расписания, однако имеет некоторые недостатки:

1. *Производительность и ограничения.* Несмотря на оптимизацию стека (Porcupine + Faster-Whisper), анализ показал, что в моменты пиковой нагрузки (активация нейросетевых моделей) потребление ресурсов устройства существенно возрастает. При использовании на персональных компьютерах средней мощности фоновые процессы способны замедлять работу системы, создавая дискомфорт для пользователя и тем самым приводя к замедлению системы при режиме отзывчивости.

2. *Сравнение с облачными аналогами.* Для различных задач подходят разные сервисы: облачные сервисы остаются актуальным и уобным решением для частных пользователей, которые не предъявляют жестких требований к конфиденциальности и готовы доверять свои данные сторонним провайдерам ради экономии ресурсов ПК. Однако при работе с данными, запрещенными к распространению, лучшим решением являются локальные сервисы.

3. *Целевая сфера применения.* Разработанная система показала наибольшую эффективность как решение для корпоративного сектора. Архитектура проекта позволяет развернуть ассистента на выделенном мощном оборудовании внутри закрытого локального домена организации.

Разработанное решение обеспечивает интеллектуальное голосовое управление без передачи данных в Интернет, что гарантирует полную защиту корпоративной информации. Такой подход целесообразен для компаний, где политика безопасности запрещает использование облачных сервисов, а наличие серверных мощностей нивелирует проблему ресурсоемкости алгоритмов. В дальнейшем в работе планируется рассмотреть возможность замены жесткого алгоритма жадного поиска на более гибкие метаэвристические методы (например, генетические алгоритмы) или внедрить механизм адаптивной, а не эмпирической настройки весовых коэффициентов для функции оценки слота.

Вклад авторов. *В. А. Чуйко* разработал концепцию работы, провел ряд исследований, сделал критический анализ системы и текста статьи. *А. В. Яскевич* осуществил разработку системы, провел исследования и написал текст статьи.

Список использованных источников

1. Robust Speech Recognition via Large-Scale Weak Supervision / A. Radford, J. W. Kim, T. Xu [et al.]. – 2022. – URL: https://arxiv.org/pdf/2212.04356 (date of access: 13.01.2026).

2. Кузьменков, Л. П. Система транскрибации речи и перевода с русского языка на китайский / Л. П. Кузьменков, В. А. Чуйко, Е. И. Козлова // Информатика. – 2025. – Т. 22, № 3. – С. 25–34. – https://doi.org/10.37661/1816-0301-2025-22-3-25-34.

3. Рыбина, Г. В. Основы построения интеллектуальных систем : учеб. пособие / Г. В. Рыбина. – М. : Финансы и статистика, 2010. – 432 с.

4. Куратов, Ю. М. Адаптация глубоких двунаправленных трансформеров-энкодеров для задач русского языка / Ю. М. Куратов, М. Ю. Архипов // Компьютерная лингвистика и интеллектуальные технологии : по материалам ежегодной Междунар. конф. «Диалог». – 2019. – № 18. – С. 333–339.

5. Зегжда, Д. П. Основы безопасности информационных систем / Д. П. Зегжда, А. М. Ивашко. – М. : Горячая линия – Телеком, 2000. – 452 с.

6. Кипяткова, И. С. Аналитический обзор систем автоматического распознавания русской речи / И. С. Кипяткова, А. А. Карпов // Труды СПИИРАН. – 2013. – № 6 (29). – С. 5–20.

7. BERT: Pre-training of deep bidirectional transformers for language understanding / J. Devlin, M.-W. Chang, K. Lee, K. Toutanova // Proc. of the 2019 Conf. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Minneapolis, MN, USA, 2–7 June 2019. – Minneapolis, 2019. – Vol. 1. – P. 4171–4186.

8. Retrieval-augmented generation for knowledge-intensive NLP tasks / P. Lewis, E. Perez, A. Piktus [et al.] // NIPS'20 : 34th Intern. Conf. on Neural Information Processing Systems, Vancouver, BC, Canada, 6–12 Dec. 2020. – Vancouver, 2020. – P. 9459–9474.

9. Бурцев, М. С. Разговорный интеллект: от тестов Тьюринга к глубокому обучению / М. С. Бурцев // Труды Московского физико-технического института. – 2022. – Т. 14, № 1. – С. 4–12.

10. Kim, J. Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech / J. Kim, J. Kong, J. Son // Proc. of the 38th Intern. Conf. on Machine Learning, ICML 2021, Online, 18–24 July 2021. – 2021. – Vol. 139. – P. 5530–5540.

11. Qwen Technical Report / J. Bai, S. Bai, Y. Chu [et al.]. – 2023. – URL: https://arxiv.org/pdf/2309.16609 (date of access: 13.01.2026).

12. Efficient keyword spotting using dilated convolutions and gating / A. Coucke, M. Chlieh, T. Gisselbrecht [et al.] // IEEE Intern. Conf. on Acoustics, Speech and Signal Processing, ICASSP 2019, Brighton, United Kingdom, 12–17 May 2019. – Brighton, 2019. – P. 6351–6355.

13. Hoy, M. B. Alexa, Siri, Cortana, and more: An introduction to voice assistants / M. B. Hoy // Medical Reference Services Quarterly. – 2018. – Vol. 37, no. 1. – P. 81–88.

14. Privacy attitudes of smart speaker users / N. Malkin, J. Deatrick, A. Tong [et al.] // Proceedings on Privacy Enhancing Technologies. – 2019. – Vol. 2019, iss. 4. – P. 250–271.

15. Attention is all you need / A. Vaswani, N. Shazeer, N. Parmar [et al.] // NIPS'17 : Proc. of the 31st Intern. Conf. on Neural Information Processing Systems, Long Beach, California, USA, 4–9 Dec. 2017. – Long Beach, 2017. – P. 6000–6010.

16. Лазарев, А. А. Теория расписаний. Задачи и алгоритмы / А. А. Лазарев, Е. Г. Мусатова. – М. : МГУ, 2012. – 208 с.

17. Edge intelligence: Paving the last mile of artificial intelligence with edge computing / Z. Zhou, X. Chen, E. Li [et al.] // Proceedings of the IEEE. – 2019. – Vol. 107, no. 8. – P. 1738–1762.

18. LLaMA: Open and Efficient Foundation Language Models / H. Touvron, T. Lavril, G. Izacard [et al.]. – 2023. – URL: https://arxiv.org/pdf/2302.13971 (date of access: 13.01.2026).

19. Щеглов, А. Ю. Защита информации: основы теории / А. Ю. Щеглов. – М. : Юрайт, 2019. – 309 с.

20. A Survey of Quantization Methods for Efficient Neural Network Inference / A. Gholami, S. Kim, Z. Dong [et al.]. – 2021. – URL: https://arxiv.org/pdf/2103.13630 (date of access: 13.01.2026).

References

1. Radford A., Kim J. W., Xu T., Brockman G., McLeavey C., Sutskever I. *Robust Speech Recognition via Large-Scale Weak Supervision*, 2022. Available at: https://arxiv.org/pdf/2212.04356 (accessed 13.01.2026).

2. Kuzmenkov L. P., Chuyko V. A., Kazlova A. I. *Speech transcription and translation system from Russian to Chinese*. Informatika [*Informatics*], 2025, vol. 22, no. 3, pp. 25−34 (In Russ.). https://doi.org/10.37661/1816-0301-2025-22-3-25-34.

3. Rybina G. V. Osnovy postroenija intellektual'nyh sistem. *Fundamentals of Building Intelligent Systems*. Moscow, Finansy i statistika, 2010, 432 p. (In Russ.).

4. Kuratov Ju. M., Arhipov M. Ju. *Adaptation of deep bidirectional transformer-encoders for Russian language tasks*. Komp'juternaja lingvistika i intellektual'nye tehnologii: po materialam ezhegodnoj Mezhdunarodnoj konferencii «Dialog» [*Computational Linguistics and Intelligent Technologies: Based on the Materials of the Annual International Conference "Dialogue"*], 2019, no. 18, pp. 333–339 (In Russ.).

5. Zegzhda D. P., Ivashko A. M. Osnovy bezopasnosti informacionnyh sistem. *Fundamentals of Information Systems Security*. Moscow, Gorjachaja linija – Telekom, 2000, 452 p. (In Russ.).

6. Kipjatkova I. S., Karpov A. A. *Analytical review of automatic Russian speech recognition systems*. Trudy SPIIRAN [*SPIIRAS Proceedings*], 2013, no. 6 (29), pp. 5–20 (In Russ.).

7. Devlin J., Chang M.-W., Lee K., Toutanova K. BERT: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association*

for Computational Linguistics: Human Language Technologies, Minneapolis, MN, USA, 2–7 June 2019. Minneapolis, 2019, vol. 1, pp. 4171–4186.

8. Lewis P., Perez E., Piktus A., Petroni F., Karpukhin V., …, Kiela D. Retrieval-augmented generation for knowledge-intensive NLP tasks. *NIPS'20: 34th International Conference on Neural Information Processing Systems, Vancouver, BC, Canada, 6–12 December 2020*. Vancouver, 2020, pp. 9459–9474.

9. Burcev M. S. *Conversational intelligence: From turing tests to deep learning*. Trudy Moskovskogo fiziko-tehnicheskogo instituta [*Proceedings of Moscow Institute of Physics and Technology*], 2022, vol. 14, no. 1, pp. 4–12 (In Russ.).

10. Kim J., Kong J., Son J. Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech. *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, Online, 18–24 July 2021*. 2021, vol. 139, pp. 5530–5540.

11. Bai J., Bai S., Chu Y., Cui Z., Dang K., …, Zhu T. *Qwen Technical Report*, 2023. Available at: https://arxiv.org/pdf/2309.16609 (accessed 13.01.2026).

12. Coucke A., Chlieh M., Gisselbrecht T., Leroy D., Poumeyrol M., Lavril T. Efficient keyword spotting using dilated convolutions and gating. *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2019, Brighton, United Kingdom, 12–17 May 2019*. Brighton, 2019, pp. 6351–6355.

13. Hoy M. B. Alexa, Siri, Cortana, and more: An introduction to voice assistants. *Medical Reference Services Quarterly*, 2018, vol. 37, no. 1, pp. 81–88.

14. Malkin N., Deatrick J., Tong A., Wijesekera P., Egelman S., Wagner D. Privacy attitudes of smart speaker users. *Proceedings on Privacy Enhancing Technologies*, 2019, vol. 2019, iss. 4, pp. 250–271.

15. Vaswani A., Shazeer N., Parmar N., Uszkoreit J., Jones L., …, Polosukhin I. Attention is all you need. *NIPS'17: Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, California, USA, 4–9 December 2017*. Long Beach, 2017, pp. 6000–6010.

16. Lazarev A. A., Musatova E. G. Teorija raspisanij. Zadachi i algoritmy. *Scheduling Theory: Problems and Algorithms*. Moscow, Moskovskij gosudarstvennyj universitet imeni M. V. Lomonosova, 2012, 208 p. (In Russ.).

17. Zhou Z., Chen X., Li E., Zeng L., Luo K., Zhang J. Edge intelligence: Paving the last mile of artificial intelligence with edge computing. *Proceedings of the IEEE*, 2019, vol. 107, no. 8, pp. 1738–1762.

18. Touvron H., Lavril T., Izacard G., Martinet X., Lachaux M.-A., …, Lample G. *LLaMA: Open and Efficient Foundation Language Models*, 2023. Available at: https://arxiv.org/pdf/2302.13971 (accessed 13.01.2026).

19. Shheglov A. Ju. Zashhita informacii: osnovy teorii. *Information Security: Theoretical Basics*. Moscow, Jurajt, 2019, 309 p. (In Russ.).

20. Gholami A., Kim S., Dong Z., Yao Z., Mahoney M. W., Keutzer K. *A Survey of Quantization Methods for Efficient Neural Network Inference*, 2021. Available at: https://arxiv.org/pdf/2103.13630 (accessed 13.01.2026).

Информация об авторах

Яскевич Антон Викторович, студент, Белорусский государственный университет.
E-mail: Tosha.yaskevich@mail.ru

Чуйко Владислав Александрович, магистр физико-математических наук, старший преподаватель, Белорусский государственный университет.
E-mail: Vchuyko@bsu.by

Information about the authors

Anton V. Yaskevich, Student, Belarusian State University.
E-mail: Tosha.yaskevich@mail.ru

Vladislav A. Chuyko, M. Sci. (Phys.-Math.), Senior Lecturer, Belarusian State University.
E-mail: Vchuyko@bsu.by