

Витрина сервисного входа ВКУС

Методическая рамка увязки клиентского пути, capabilities и архитектуры

Короткий рабочий документ для команды. Цель - связать изменения клиентского пути с требуемыми бизнес-возможностями, техническими возможностями, архитектурными компонентами, roadmap и метриками эффекта.

Главная установка	
Не так Проектируем отдельный интерфейс для вопросов пользователей.	А вот так Проектируем витрину сервисного входа ВКУС, которая помогает пользователю решить проблему или корректно создать обращение.

- Документ не фиксирует конкретный интерфейсный паттерн как единственное решение.
- Все функции должны быть обоснованы через изменение клиентского пути.
- Roadmap начинается с Data Discovery и baseline, а не с разработки компонентов.
- Целевой результат - решение рабочей проблемы пользователя, а не сам факт создания обращения.

1. Назначение и логика трассировки

Материал задает единый способ перехода от клиентского пути к архитектуре. Он нужен, чтобы команда проектировала не набор функций, а измеримые изменения в опыте пользователя.

Базовая цепочка:

Этап клиентского пути боль / разрыв целевое изменение business capability data / process capability technical capability architecture component metric MVP / release

Если функция не связана с болью пользователя и изменением клиентского пути, она не должна попадать в roadmap.

Ключевые изменения мышления

Объект проектирования	
Не так Отдельный интерфейс для вопросов.	А вот так Витрина сервисного входа ВКУС.
Путь пользователя	
Не так Найти услугу и заполнить анкету.	А вот так Описать задачу и получить лучший следующий шаг.
Capabilities	
Не так Список функций системы.	А вот так Способности компании обеспечивать целевой клиентский путь.
Архитектура	
Не так Набор компонентов.	А вот так Реализация технических возможностей, выведенных из CJM.
Данные	
Не так Материал для отчетности.	А вот так Фактическая основа выбора MVP и улучшения сервиса.

2. Главный рабочий артефакт: матрица трассировки

Матрица связывает UX, бизнес-анализ и архитектуру. Она показывает, зачем нужна каждая capability, какие gaps есть в HLD и какие изменения должны попасть в MVP.

Этап CJM	Боль / разрыв	Целевое изменение	Business capability	Data / process capability	Technical capability	Архитектурные компоненты	Метрика	MVP
Пользователь ищет услугу	Не знает внутреннюю классификацию ИТ-услуг	Описывает задачу своими словами, витрина предлагает сценарий	Понимание пользовательского намерения	Корпус обращений, поисковые запросы, словарь синонимов, история реклассов	Классификация намерения, семантический поиск	ВКУС, сервис классификации, каталог услуг, API Gateway, ITSM	Reclass rate, time to submit	MVP 3
Пользователь заполняет карточку	Много обязательных полей, часть данных уже известна	Карточка предзаполнена, пользователь вводит только недостающее	Минимизация ручного ввода	Данные пользователя, подразделение, роль, система, история обращений	Предзаполнение формы, интеграция с мастер-данными	ВКУС, prefill service, AD/IAM, HR, CMDB, ITSM	Ручные поля, time to submit	MVP 1
Пользователь ищет инструкцию	Инструкции разрознены, поиск не помогает	Витрина предлагает короткий ответ или инструкцию до обращения	Помощь до создания обращения	База знаний, метаданные инструкций, история типовых ответов	Поиск по знаниям, ранжирование, ответ с источником	ВКУС, база знаний, RAG/search-контур, API Gateway	Deflection, search success	MVP 2
Пользователь ожидает решение	Нет понятного статуса и следующего действия	Пользователь видит этап, срок и что требуется от него	Управление статусом и ожиданиями	Статусы ITSM, SLA, комментарии, события процесса	Нормализация статусов, уведомления, next best action	ITSM, ВКУС, status/event service, notifications	CSAT, обращения «где статус»	MVP 4
Пользователь оценивает результат	Обратная связь не превращается в улучшения	Оценка запускает улучшение карточек, знаний и маршрутизации	Управление качеством клиентского пути	Оценки, комментарии, переоткрытия, повторные обращения	Аналитика обратной связи, кластеризация проблем	ВКУС, ITSM, Data Mart, BI	Reopen, FCR, CSAT	MVP 5

3. Capability map целевой витрины

Capability - это не кнопка, компонент или экран. Это способность компании обеспечить целевой фрагмент клиентского пути.

Слой capabilities	Состав
Клиентские	Единый вход во ВКУС; описание проблемы на языке пользователя; быстрый ответ до обращения; понятная инструкция или следующий шаг; предзаполненное обращение; прозрачный статус; подтверждение результата.
Business capabilities поддержки	Управление каталогом услуг; управление знаниями; управление обращениями; управление маршрутизацией; управление SLA и ожиданиями; управление обратной связью; управление качеством клиентского пути.
Data / analytics	Тотальный анализ текстов обращений; анализ ответов техподдержки; анализ комментариев и уточнений; анализ логов действий пользователей и поддержки; выявление намерений, сценариев, проблемных карточек и knowledge gaps.
Technical capabilities	Семантический поиск; поиск по базе знаний; классификация намерений; intent service / knowledge / ticket mapping; предзаполнение; интеграции с мастер-системами и ITSM; статусы; уведомления; аналитика качества.

Мини-пример различения уровней

Не capability	Capability
Кнопка на форме	Минимизация ручного ввода
Поиск	Помощь до создания обращения
API-интеграция	Управляемая оркестрация сервисного процесса
Экран статуса	Управление ожиданиями пользователя
BI-отчет	Аналитика клиентского пути

4. Целевая архитектурная рамка

Архитектурный сдвиг	
Не так Пользователь вопросно-ответный интерфейс поиск ответа ответ / заявка	А вот так Пользователь ВКУС-витрина сценарии и намерения знания и каталог услуг карточки и предзаполнение ITSM статусы аналитика и улучшение

Слой	Назначение	Примеры компонентов
Frontstage / ВКУС	То, что видит пользователь	Задачная главная, умный поиск, карточка, мастер решения, статус, оценка
Scenario & Intent Layer	Понимание задачи пользователя	Intent service, scenario engine, clarification logic
Knowledge & Service Catalog Layer	Помощь до обращения и связь с услугами	База знаний, каталог услуг, RAG/search-контур, service mapping
Form & Prefill Layer	Минимизация ручного ввода	Prefill service, form engine, validation rules, context package
Integration & Orchestration Layer	Управляемые интеграции	API Gateway, connectors, auth, audit, policies
ITSM Execution Layer	Создание и исполнение обращений	ITSM / Creatio, routing, SLA, comments sync
Status & Notification Layer	Понятное ожидание	Status/event service, notification service, next best action
Analytics & Improvement Layer	Улучшение сервиса на данных	Event tracking, Data Mart, text analytics, BI, feedback loop

5. Gap-analysis текущего HLD

Текущий HLD содержит полезное технологическое ядро: ВКУС, API Gateway, LiteLLM, RAG/search-контур, интеграцию с ITSM / Creatio и логику передачи контекста. Его нужно расширить от отдельного интерфейсного решения до архитектуры витрины сервисного входа.

Capability	Покрытие в HLD	Что требуется добавить
Поиск по инструкциям	Покрыто / частично	Требования к качеству источников, актуальности, владельцам знаний
Понимание намерения пользователя	Частично	Выделить intent detection как отдельную capability и компонент
Сопоставление намерения с услугой	Не покрыто	Добавить mapping: intent service / knowledge / ticket
Предзаполнение обращения	Не покрыто	Добавить prefill service и интеграции с AD/IAM, HR, CMDB, ITSM
Динамическая карточка	Не покрыто	Добавить form/scenario engine
Прозрачный статус	Не покрыто	Добавить status/event service и UX-словарь статусов
Аналитика клиентского пути	Не покрыто	Добавить event tracking, Data Mart, BI, text analytics
Feedback loop	Частично	Связать оценки с backlog улучшений карточек, знаний и маршрутизации

HLD должен быть переработан в документ «High-Level Design: Витрина сервисного входа ВКУС», где каждый компонент связан с technical capability, каждая technical capability - с business capability, а каждая business capability - с изменением клиентского пути.

6. Roadmap / MVP-логика

Принцип roadmap	
Не так MVP 1: поиск; MVP 2: LLM; MVP 3: интеграция с ITSM; MVP 4: аналитика.	А вот так MVP 0: фактическая карта проблем на данных; MVP 1: меньше ручного заполнения; MVP 2: помощь до обращения; MVP 3: намерения; MVP 4: прозрачное ожидание; MVP 5: контур улучшения.

MVP	Фокус	Изменение в клиентском пути	Ключевые компоненты	Метрики
MVP 0	Data Discovery	Понять фактические разрывы текущего пути	Выгрузки ВКУС/ITSM, анализ текстов, логов, Data Mart light	Baseline, reclass, clarification, reopen
MVP 1	Упрощение карточки	Быстрее и проще подать обращение	ВКУС, prefill, мастер-данные, ITSM	Time to submit, ручные поля, clarification rate
MVP 2	Помощь до обращения	Решать часть вопросов без заявки	Knowledge layer, search/RAG, API Gateway	Deflection, search success, helpfulness
MVP 3	Намерения	Не выбирать услугу вручную	Intent service, catalog, routing, ITSM	Reclass, reassignment, time to submit
MVP 4	Статус	Понятно ждать и отвечать на уточнения	Status/event service, notifications	CSAT, обращения по статусу
MVP 5	Улучшение	Сервис сам выявляет, что улучшать	CX Data Mart, BI, text analytics	CSAT, FCR, reopen

7. MVP 0: Data Discovery как обязательный старт

Логика старта	
Не так Берем одну услугу и экспертно решаем, что улучшить.	А вот так Сначала анализируем весь массив обращений, ответов и логов, строим фактическую карту проблематики, затем выбираем пилотную услугу на основании данных.

Источник	Ценность
Тексты обращений	Пользовательские намерения, реальные формулировки, частотные темы
Ответы поддержки	Типовые решения, повторяющиеся ответы, качество языка
Комментарии	Уточнения, потеря контекста, недостающие данные
Маршрутизация	Реклассы, переназначения, ошибки выбора услуги
Статусы и сроки	Ожидание, задержки, зависания
Переоткрытие	Где результат не решает проблему с первого раза
Оценки	Качество глазами пользователя
Поиск во ВКУС	Пользовательский язык поиска и пробелы каталога
Логи действий во ВКУС	Открытые карточки, возвраты, незавершенные обращения
Действия поддержки	Типовые ручные операции, запросы данных, смены групп

Выходы MVP 0

- карта пользовательских намерений;
- топ проблемных услуг / карточек;
- список типовых недостающих данных;
- список повторяющихся ответов поддержки;
- список knowledge gaps;
- карта ошибок маршрутизации;
- baseline метрик;
- обоснованный выбор пилотной услуги.

8. Метрики и критерии успеха

Как оценивать эффект	
Не так Реализовано предзаполнение карточки.	А вот так Пользователь меньше вводит вручную, быстрее отправляет обращение, а поддержка реже запрашивает недостающие данные.

Блок	Метрика
Усилия пользователя	Time to submit
Усилия пользователя	Количество ручных полей
Поиск / помощь	Search success rate
Поиск / помощь	Deflection rate
Качество входа	Reclass rate
Качество входа	Clarification rate
Исполнение	MTTR
Результат	Reopen rate
Результат	FCR
Опыт	CSAT

Область	Как было	Как стало
Вход в поддержку	Пользователь выбирает канал и услугу	Пользователь описывает задачу, витрина предлагает сценарий
Инструкции	Пользователь сам ищет и читает длинные материалы	Витрина предлагает короткий ответ, источник и следующий шаг
Карточка	Пользователь заполняет анкету для поддержки	Пользователь проверяет предзаполненный контекст
Маршрутизация	Ошибка выбора услуги ведет к переназначениям	Намерение сопоставляется с услугой и группой поддержки
Ожидание	Пользователь видит формальный статус	Пользователь понимает этап, срок и next best action
Результат	Обращение закрыто в ITSM	Пользователь подтвердил решение рабочей проблемы
Улучшение	Обратная связь собирается отдельно	Обратная связь попадает в backlog улучшений

9. Решения, задачи и правила для команды

Ключевые решения

- Объект проектирования - витрина сервисного входа ВКУС.
- Roadmap строится от клиентского пути, а не от компонентов.
- Capabilities - это способности компании, а не функции интерфейса.
- HLD должен быть расширен до архитектуры витрины.
- MVP 0 - обязательный Data Discovery.
- Пилот выбирается на основании данных.
- Каждая функция должна иметь метрику эффекта.

Ближайшие задачи

#	Задача
1	Утвердить принцип: проектируем витрину сервисного входа.
2	Утвердить формат матрицы трассировки.
3	Запустить тотальный анализ обращений, ответов поддержки и логов действий.
4	На основе анализа выбрать пилотную услугу.
5	Собрать baseline по текущей карточке пилотной услуги.
6	Разобрать пилотную услугу детально: поля, уточнения, ошибки, маршрутизация.
7	Спроектировать короткую предзаполненную карточку.
8	Описать mini-HLD пилота.
9	Запустить пилот на ограниченной группе пользователей.
10	Измерить эффект и принять решение о масштабировании.

Рабочие правила

- Не начинать с выбора интерфейсного решения.
- Любая функция должна быть связана с болью клиента.
- Технический компонент не является capability.
- Интерфейсный паттерн не является бизнес-возможностью.
- Каждая capability должна иметь владельца и метрику.
- Каждое изменение HLD должно быть связано с изменением в клиентском пути.
- Roadmap должен начинаться с анализа данных и baseline.
- Витрина должна снижать усилия пользователя, а не переносить анкетирование в новый интерфейс.
- Целевой результат - не создание обращения, а решение пользовательской проблемы.

10. Итоговая формулировка

Команда проектирует не новый канал общения с поддержкой, а витрину сервисного входа во ВКУС.

Витрина должна менять клиентский путь: снижать ручной ввод, помогать до создания обращения, понимать намерение пользователя, создавать обращения с контекстом, показывать понятный статус и возвращать данные в контур улучшения.

Каждая функция витрины должна быть связана с этапом клиентского пути, business capability, data / process capability, technical capability, архитектурным компонентом и метрикой эффекта.

Первый практический шаг - MVP 0: тотальный анализ обращений, ответов поддержки и логов действий. Только после этого выбирается пилотная услуга и запускается проверка на ограниченном сценарии.

Финальная проверка: если команда не может объяснить функцию через связку «изменение клиентского пути capability архитектурный компонент метрика», функция не готова для включения в roadmap.