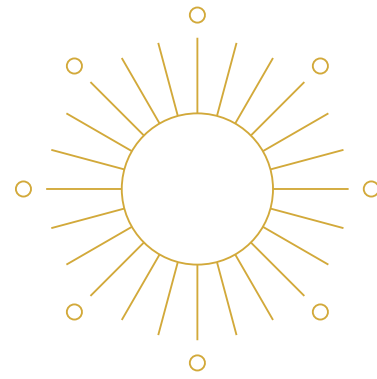


Money Map

Пакет спецификации проекта (10-15 страниц)

Версия: 2026-01-27



Коротко

Money Map - офлайн-first инструмент, который подбирает подходящие способы заработка под конкретную страну и профиль человека.

Он объединяет: формулу (что продаем / кому / как измеряется ценность), матрицу 2x2x2, таксономию 14 механизмов дохода, мосты и маршруты.

Реальный мир: Feasibility Engine (выполнимость), Economics Snapshot (сроки/диапазоны дохода), Legal & Compliance Kit (чеклисты и гейты) + контроль устаревания данных.

Статус: спецификация для реализации. Реализуется как Python-пакет с CLI (Typer) и Web-UI (Streamlit).

Содержание

Содержание 2

1. One-pager 3

2. Use cases и пользовательские сценарии 4

3. MVP-границы (MoSCoW) 5

4. Функции и требования 6

5. UX/UI: экраны, состояния, навигация 8

6. Данные: сущности, связи, хранение состояния 9

7. NFR: производительность, платформы, ограничения 11

8. Архитектура: модули и взаимодействие 12

9. План релизов: этапы и критерии готовности 13

10. Definition of Done и тест-подход 14

Приложение А. Patch Pack v1 (кратко) 15

1. One-pager

Цель: помочь человеку выбрать и запустить наиболее подходящий способ заработка в выбранной стране/контексте, учитывая реальные ограничения (законы, барьеры входа, ресурсы, сроки и деньги).

Для кого (MVP): люди, живущие в Германии (DE), которые хотят увеличить доход за 3-6 месяцев и готовы действовать по пошаговому плану; уровень языка и ресурсы могут быть ограничены.

Ценность: вместо абстрактных советов пользователь получает ранжированный список вариантов + объяснения + реалистичный план на 4 недели + юридико-комплаенс чеклист (без юридических вердиктов).

Результаты на выходе (Deliverables)

Топ-N вариантов с оценкой выполнимости, сроком до первых денег и диапазоном дохода (в виде диапазонов, не обещаний).

План запуска выбранного варианта: маршрут (мосты/пути), 4-недельный календарный план, артефакты и чекпойнты.

Legal gate + compliance kit: список проверок/документов/вопросов (не юридическая консультация).

Экспорты: plan.md, result.json, profile.yaml.

Аналоги/пересечения (опционально): заметки/шаблоны, карьерные тесты, консультанты, базы идей. Отличие Money Map: единая формальная модель + объяснимые решения + офлайн-first.

Не является: юридическим сервисом, биржей вакансий, системой прогнозирования дохода. Это инструмент структурирования и выбора с проверками и планом.

2. Use cases и пользовательские сценарии

Основной сценарий (end-to-end):

- 1) Пользователь задает страну/контекст и быстрый профиль (время, капитал, язык, активы, ограничения, цель).
- 2) Система показывает top-10 вариантов с реализмом: feasibility, legal gate, time-to-first-money, net range, costs.
- 3) Пользователь выбирает вариант и получает план: шаги, артефакты, 4-недельный план, чеклисты.
- 4) Пользователь экспортирует план и действует; при изменении профиля (время/капитал) пересчитывает.

Вторичные сценарии:

- А) Исследование карты: поиск по механизму дохода/ячейке матрицы/тегам; просмотр мостов и маршрутов.
- В) Классификация идеи: пользователь вводит текст идеи - система предлагает 3 кандидата taxonomy + cell + объяснения.
- С) Реальность-проверка (Reality Check): система показывает 3 главных блокера и быстрые способы расширить множество вариантов.

Правило реального мира

Каждый результат должен содержать: (1) что нужно сделать, (2) что может помешать, (3) что проверить по закону/комплаенсу, (4) сколько времени до первых денег и диапазон дохода.

3. MVP-границы (MoSCoW)

Категория	Содержимое
MUST	Офлайн-first; 1 страна (DE); dataset YAML/JSON; validate; recommend с objective presets; feasibility + economics + legal/compliance; plan (route + 4 weeks); экспорт (md/json/yaml); Streamlit UI.
SHOULD	Wizard (guided flow); compare 2-3 вариантов; пресеты весов; HTML-граф таксономии; больше вариантов (100-200).
COULD	Доп. страны; простые интеграции (без API) как ссылки; более богатая классификация; импорт пользовательских вариантов.
WON'T (MVP)	Онлайн-парсинг законов; гарантии дохода; автоматическое заполнение вакансий; аккаунты/облако; юридические заключения.

MVP фиксируется, чтобы не утонуть: DE + качественный набор 30-50 вариантов + ruleraсk с конкретными чеклистами.

4. Функции и требования

Ниже - MUST-фичи и контракты (входы/выходы/правила/ошибки).

MU ST	Фича	Коротко
1	Data loading + Validation	safe_load, schema_version, refs, семантика, отчет проблем (FATAL/WARN).
2	Recommendation Engine	кандидаты -> фильтры -> скоринг по objective -> top-N + explanations + diagnostics.
3	Route Plan + Plan.md	путь из bridges/paths + шаги с артефактами + агрегированный чеклист + 4 недели.
4	Streamlit UI	Profile -> Recommendations -> Plan -> Export + Data status + Reality Check.
5	Feasibility Engine	feasible/with_prep/not_feasible + blockers + prep_steps + estimate weeks.
6	Economics Snapshot	time_to_first_money + net range + volatility + costs + confidence.
7	Legal Gate + Compliance Kit	ok/require_check/registration/license/blocked + конкретные checklist и kits + staleness.

4.1 MUST-2: Recommendation Engine (контракт)

Входы

UserProfile (location, time/capital, language, assets, skills, constraints, objective preset).

AppData (variants, taxonomy, cells, bridges/paths, keywords) + RulePack (country).

Filters (top_n, max_risk, max_legal_friction, startability_window, include/exclude tags).

Выходы

RecommendationResult: ranked_variants[], explanations (3 причины 'за', 1-2 'против'), diagnostics (сколько отфильтровано и почему).

Для каждого варианта: FeasibilityResult + EconomicsSnapshot + LegalEvaluation + confidence/staleness.

Правила: детерминизм; запрещены обещания дохода; только диапазоны; tie-breaker фиксированный (например по variant_id). Если legal/rulepack устарел - повышаем осторожность (force require_check) для регулируемых доменов.

4.2 MUST-5/6/7: Feasibility, Economics, Legal (реальный мир)

Эти три слоя обязательны, иначе система будет давать абстрактные советы.

Feasibility Engine

Статус: feasible / feasible_with_prep / not_feasible.

Blockers: максимум 3 в summary (например: язык ниже floor, нет необходимого актива, недостаточно времени).

Prep steps: конкретные действия (что сделать за 2 часа/1 неделю).

Estimated prep: диапазон недель.

Economics Snapshot

time_to_first_money (days, range), typical_net_month (eur, range).

volatility/seasonality, variable_costs, scaling_ceiling, confidence.

Если economics отсутствует - penalize в ранжировании и показать warning.

Legal gate + Compliance kit

legal_gate: ok / require_check / registration / license / blocked (без 'разрешено').

Чеклист должен быть конкретным: вопросы/документы/контакты для проверки.

Compliance kits: tax_basics, invoicing, insurance_basics, etc. с outputs.

Staleness: если reviewed_at старый - предупреждение и повышенная осторожность.

5. UX/UI: экраны, состояния, навигация

UI (Streamlit) реализует быстрый сценарий и исследовательский режим. Навигация - sidebar + вкладки.

Экран	Цель	Ключевые элементы	Состояния
Data status	Проверить данные	dataset_version, reviewed_at, warnings, validate report	valid / invalid / stale
Profile	Ввод профиля	Quick (5-7 полей) + Advanced	draft / ready
Recommendations	Top-N + выбор	objective preset, filters, cards, Reality Check	loading / empty / results
Plan	План запуска	steps, 4-week view, outputs, checkpoints, compliance kit	no_selection / ready
Export	Скачивание	plan.md, result.json, profile.yaml	available / not_ready
Explore	(SHOULD) карта	matrix, taxonomy graph, bridges/paths	browse

Reality Check (обязательный блок на экране Recommendations)
Top blockers (3) по текущему профилю: что ограничивает варианты.
Quick fixes: кнопки 'снизить legal friction', 'цель fastest money', 'только startable in 2 weeks'.
Цель: за 1-2 минуты сделать решение понятным.

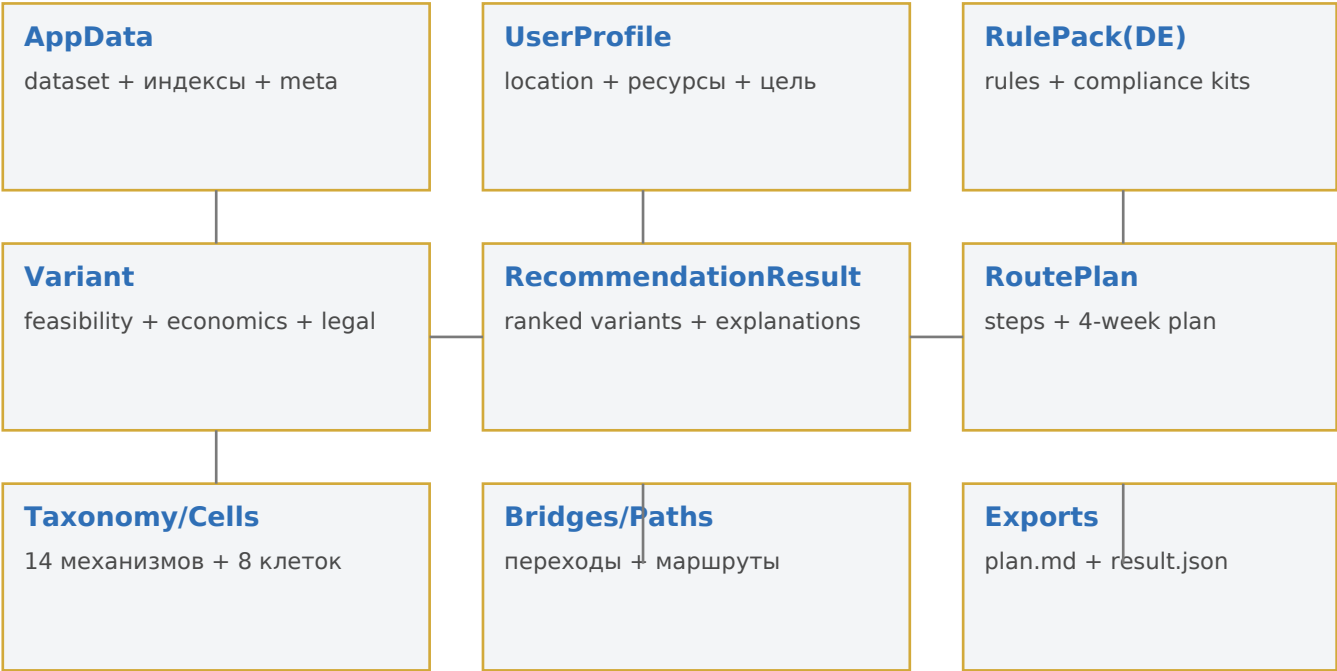
6. Данные: сущности, связи, хранение состояния

Все данные по умолчанию хранятся локально (YAML/JSON). Состояние UI - в session_state и кэше Streamlit.

Ключевые сущности:

Сущность	Назначение	Ключевые поля
Variant	Конкретный способ/вариант заработка	feasibility, economics, legal, taxonomy_id, cells, tags, review_date
UserProfile	Профиль пользователя	location, time, capital, language, assets, skills, constraints, objective
RulePack	Правила страны + комплаенс	rules, compliance_kits, reviewed_at, staleness_policy
Bridge/Path	Переходы и маршруты	from_cell, to_cell, steps(outputs), preconditions
RecommendationResult	Результат ранжирования	ranked_variants, explanations, diagnostics, profile_hash
RoutePlan	План запуска	route_steps, 4-week_plan, aggregated_checklist, artifacts

Схема связей (высокий уровень):



Где хранится состояние
Данные: data/*.yaml (и data/rulepacks/DE.yaml).
Кэш: Streamlit cache_resource/cache_data (AppData, результаты recommend).

Session state: текущий профиль, objective, фильтры, выбранный вариант, план.
Экспорты: папка exports/ (plan.md, result.json, profile.yaml).

7. NFR: производительность, платформы, ограничения

Категория	Требование (MVP)	Метрика/проверка
Производительность	validate <= 2с; recommend <= 1.5с (1000 variants)	pytest perf smoke / ручной прогон
Офлайн	все работает без интернета	нет сетевых вызовов
Безопасность	safe_load; без eval/exec; UI без raw HTML из данных	unit tests + code review
Аудитируемость	в экспорт включать rule_ids и причины	result.json содержит applied_rules
Staleness	предупреждать при устаревании rulepack/variants	review_date + staleness_policy
Платформы	Windows + Linux/Mac опционально	pip install -e .; ui extra

Ограничение MVP: не делаем юридических заключений и гарантий дохода; только диапазоны и чеклисты.

8. Архитектура: модули и взаимодействие

Архитектура слоистая: data -> core -> render -> interfaces (CLI/UI). Core не зависит от Streamlit/Typер.

Слой/модуль	Ответственность	Не делает
core/load.py	загрузка dataset + meta + rulepack	скоринг/рендер
core/validate.py	целостность refs/семантика	исправления данных
core/feasibility.py	feasible/with_prep/not_feasible	UI/экспорт
core/economics.py	снимок экономики + confidence	обещания дохода
core/rules.py	legal gate + compliance kits + staleness	юридический вердикт
core/recommend.py	ранжирование по objective	визуализация
core/graph.py	маршруты/мосты	контент шагов (это данные)
render/*	md/json/ascii/dot	бизнес-логика
app/cli.py	Typер CLI	алгоритмы
ui/*	Streamlit UI	алгоритмы

- Расширяемость (extension points)
- Новая страна: добавить data/rulepacks/XX.yaml + tests на staleness/coverage.
- Новые варианты: добавить variants с economics/feasibility/legal и review_date.
- Новые objectives: добавить preset весов без изменения моделей.
- Более умная классификация: заменить engine внутри core/classify, сохранив контракт результата.

9. План релизов: этапы и критерии готовности

План строится по принципу: сначала доказать полезность (golden dataset + engines), потом UI/полировка.

Этап	Содержание	Демо/критерий готовности
M0	Скелет репо, CLI/UI заглушки, CI	pip install -e .; money-map --help; money-map ui открывается
M1	Golden dataset v0 + meta + DE rulepack, load+validate	money-map validate OK; taxonomy/cells выводятся
M2	Query/search + индексы	search дает релевантные результаты
M3	Legal gate + compliance builder + staleness	check выводит gate+чеклист; stale -> warning
M4	Feasibility + Economics + recommend objectives	top-10 осмысленный; причины; диапазоны
M5	RoutePlan + 4-week plan + exports	plan.md содержит шаги, артефакты, compliance секцию
M6	Streamlit UI MVP (E2E flow)	Profile -> Recommendations -> Plan -> Export без падений
M7	Стабилизация: тесты, CI, контент	pytest green; demo профили стабильны

10. Definition of Done и тест-подход

Проект 'готов (MVP)', если проходит E2E сценарий, данные валидны, результаты детерминированы и покрыты тестами.

Категория	Критерий
E2E UI	Profile -> top-10 -> Plan -> Export; без stacktrace; offline
FeasibleTop10Rate	>= 70% на demo профилях (не blocked и без hard conflicts)
PlanActionability	>= 10 действий и >= 3 артефакта в plan.md
LegalChecklistCoverage	для regulated domains чеклист не пустой + compliance kits
StalenessBehavior	если reviewed_at старый: warning + force require_check для regulated domains
Determinism	одинаковые входы -> одинаковый top-10
CI	pytest + money-map validate в GitHub Actions

Тесты (минимум):

- test_validate_data.py: refs, enums, семантика
- test_feasibility_engine.py: статус и блокиры
- test_economics_snapshot.py: ranges, penalties, confidence
- test_rules_legal_gate.py: gate + checklist
- test_rulepack_staleness.py: предупреждения и safety behavior
- test_recommend_objectives.py: пресеты меняют ранжирование
- test_plan_includes_compliance.py: plan.md содержит compliance секцию
- test_ui_import.py: smoke import UI

Чеклист ручного демо-прогона (10 минут)

1) money-map validate

2) money-map recommend --profile profiles/demo_fast_start.yaml --top 10

3) выбрать top-1 -> money-map plan ...

4) money-map export all

5) UI: пройти flow и скачать файлы

6) проверить: дисклеймер, dataset_version, warnings (если stale).

Приложение А. Patch Pack v1 (кратко)

Этот пакет добавляет реализм: feasibility, economics, legal/compliance и staleness. Служит как обязательное расширение к базовому шаблону.

Компонент	Что добавляет	Где лежит
meta.yaml	dataset_version + staleness_policy	data/meta.yaml
DE rulepack	legal gates + compliance kits + reviewed_at	data/rulepacks/DE.yaml
Variant расширения	feasibility/economics/legal/evidence	data/variants.yaml + core/model.py
Engines	feasibility.py, economics.py, rules.py	src/money_map/core/
UI Reality Check	blockers + quick fixes + objective selector	src/money_map/ui/views/recommendations.py

Конец документа.