

KLT 4.13 / 4.14 / 5.1 / 5.10. Итоговый корпус чек-листов

Иван Борисович Курпишев · Independent Researcher · Kaliningrad · 2026

KLT-PROGRAM-FINAL-CHECKLISTS-KLT413-KLT414-KLT51-KLT510-RU-v2.5

Источник и основание анализа

Ниже указана внутренняя источниковая рамка, использованная для чек-листов. Отдельно отмечено, что KLT 4.13 и KLT 5.10 в данном пакете трактуются как baseline/target layers, если самостоятельный архив соответствующей версии отсутствует в текущем наборе файлов.

ID	Источник	Роль
SRC-CORE-01	monograph5_0_ru.pdf	Монография 5.0: ядро C@C, Rep(R,I,U;D), lambda, CGI, KLT/RBD.
SRC-PILOT-01	PILOT01_PREPRINT_RU_FINAL.pdf	Дисциплина formula-chain audit: harmonic condition не дает truth-status без domain и sufficient foundation.
SRC-VOL6-01	KLT_DOCTRINE_VOL6_KLT_RBD_SOFTWARE_LEGAL_RU_v8_1.md	Том VI: программный слой, KLT 4.14, KLT 5.1, RBD/RPD, сайт, public/private split.
SRC-413-01	KLT_4_14_CHECKED_BUILD.zip / README_RU.md	Секция "Сохранено из 4.13": восстановление baseline KLT 4.13.
SRC-414-01	KLT_4_14_CHECKED_BUILD.zip	Исходная сборка KLT 4.14: CLI, importers, reports, auth, web, Flutter shell, selftest.
SRC-51-01	KLT5_1_FLUTTER_SDK_PACKAGE.zip	SDK KLT 5.1: Flutter UI, Python reference core, Project JSON, lambda engine, SVG/JSON/MD outputs.
SRC-510-01	klt_rbd_v2_2_flutter_web_program (1).zip	KLT-RBD v2.2 Flutter/Web program as source base for KLT 5.10 target checklist.
SRC-PUB-01	KLT_RBD_PREDICTOR_PUBLIC_DEMO_* v1.8-v2.4	Публичный read-only contour, site payload, API/UI, remote evidence and release logs.

Сводная карта решений

Программа	Статус	Назначение	Главная цепь
KLT 4.13	reconstructed legacy baseline	legacy baseline: учебно-аудиторская lambda-проверка текстовых работ	Work -> text normalization -> ideal-answer comparison -> lambda checklist -> 100-point score -> 5-point scale -> grade register.
KLT 4.14	checked source/build or SDK package	checked source/build package: универсальный импорт, web/mobile shell, demo/paid gate	Input files/archives -> import/OCR adapter -> lambda audit -> DOCX checklist/register -> web/mobile shells -> paid/demo export gate.
KLT 5.1	checked source/build or SDK package	Unified Flutter SDK: PIX@PEAKS, Project JSON, lambda engine, SVG/JSON/MD reports	Project JSON -> PIX@PEAKS graph -> Reper(U,I,R,D) -> lambda/delta/Auth -> D*/MSI/DI/CRI -> SVG/JSON/MD report.
KLT 5.10	target architecture checklist	target next-generation KLT-RBD: Flutter/Web + FastAPI + extraction/OCR + RBD + LambdaAudit + Rebuild	Intake(F) + OCR + Extract + RBD + LambdaAudit + LimitGates + Report + Rebuild.

KLT 4.13: итоговый развернутый чек-лист

legacy baseline: учебно-аудиторская lambda-проверка текстовых работ

Статус исходного анализа

В текущем корпусе отдельный архив KLT 4.13 не найден. Чек-лист реконструирован из раздела "Сохранено из 4.13" в KLT 4.14 и из программно-регистрационного Тома VI.

Главная формула слоя

Work -> text normalization -> ideal-answer comparison -> lambda checklist -> 100-point score -> 5-point scale -> grade register.

Анализ исходного текста

- KLT 4.13 является исходной учебно-аудиторской линией: индивидуальный DOCX-чек-лист, массовая проверка ZIP, сравнение с идеальным ответом, ЕГЭ-совместимая 100-балльная нормировка и 5-балльная шкала.
- Ключевая методическая функция: не заменить преподавателя, а формализовать структурную проверку работы через текстовые признаки, критерии, замечания и ведомость.
- Для итогового freeze следует отделять наследуемую функцию KLT 4.13 от расширений KLT 4.14: универсальный импорт, web-shell, OCR и demo/paid-gate не входят в чистый baseline 4.13.

Репер-паспорт программы

Компонент	Интерпретация
R_build	Фактический учебно-аудиторский механизм проверки текста и формирования оценочного результата.
I_lambda_check	Идея лямбда-чек-листа: работа проверяется не как голая оценка, а как структурная согласованность с идеальным ответом и критериями.
U_classroom	Поле применения: школа, класс, группа, преподаватель, массовая ведомость, сравнение работ.
D_verification	Основание: идеальный ответ, критерии, входной текст, файл ученика, журнал проверки, DOCX-чек-лист.

Итоговый checklist

Идентичность и граница версии

ID	OK	Проверяемый пункт	Критерий приёмки / Evidence-D	Комментарий
413-ID-01	[]	Название и версия фиксированы как KLT 4.13 legacy baseline.	Есть отдельное обозначение KLT 4.13; указано, что это не KLT 4.14.	
413-ID-02	[]	Зафиксирована граница: текстовая учебная проверка, без универсального импорта и мобильной коммерческой оболочки.	В паспорте версии указано: legacy / baseline / не signed application.	
413-ID-03	[]	Сохранена авторская атрибуция И.Б. Курпишева.	В документах и шаблонах указан автор/правообладатель проекта.	
413-ID-04	[]	Для каждого отчёта указаны дата, проверяющий, класс/группа, файл, идеальный ответ.	Отчёт воспроизводим без неявной памяти проекта.	

Функциональный аудит

ID	OK	Проверяемый пункт	Критерий приёмки / Evidence-D	Комментарий
413-FN-01	[]	Индивидуальный lambda-чек-лист DOCX формируется для одной работы.	DOCX содержит критерии, баллы, замечания и итог.	
413-FN-02	[]	Массовая проверка ZIP/папки не нарушает идентификацию учеников.	Каждая работа имеет отдельный результат и строку ведомости.	
413-FN-03	[]	Сравнение с идеальным ответом выполняется детерминированно.	При повторном запуске на тех же данных результат стабилен.	
413-FN-04	[]	Нормализация классов 7-б / 7б / 7 Б -> 7-Б сохраняет группировку.	Классы и группы не дробятся из-за орфографических вариантов.	
413-FN-05	[]	Отдельный лист подозрения на заимствование не смешивается с оценкой.	Плагиатный риск помечается как review-флаг, не как автоматический приговор.	

Оценочные шкалы

ID	OK	Проверяемый пункт	Критерий приёмки / Evidence-D	Комментарий
413-SC-01	[]	100-балльная шкала совместима с учебной ведомостью.	Итоговая строка содержит score 100.	
413-SC-02	[]	5-балльная шкала с плюсами/минусами выводится из 100-балльной.	Схема пересчёта документирована.	
413-SC-03	[]	Три строки замечаний проверяющего сохраняются в отчёте.	Отчёт допускает человеческую редакцию, но не скрывает автоматический результат.	
413-SC-04	[]	Разделение автоматической оценки и экспертного комментария явно показано.	Нет смешения машинного индикатора и окончательного педагогического решения.	

D-основания и риски

ID	OK	Проверяемый пункт	Критерий приёмки / Evidence-D	Комментарий
413-D-01	[]	Без идеального ответа результат получает статус needs-D.	Нет truth-status без sufficient foundation.	
413-D-02	[]	Без исходного текста работы отчёт не создаётся как полный.	Пустой или повреждённый файл переводится в gap-input.	
413-D-03	[]	Нечитаемая кодировка фиксируется как import-gap.	Не допускается молчаливое искажение текста.	
413-D-04	[]	Итоговый PDF/DOCX должен иметь контрольную сумму.	Артефакт можно повторно идентифицировать.	

Acceptance gate

`KLT413Ready(x) <=> InputText(x) and IdealAnswer(x) and Criteria(x) and ChecklistDOCX(x) and GradeRegister(x) and NoHiddenManualOverride(x)`.

Если отсутствует идеальный ответ, исходный текст или критерии, объект не отвергается, а получает gap-status: needs-ideal, needs-input или needs-criteria.

Риски и обязательные оговорки

- Нет самостоятельного пакета KLT 4.13 в текущей папке источников; версия восстанавливается как baseline из KLT 4.14.
- Нельзя переносить в KLT 4.13 платный режим, OCR и мобильную оболочку как уже существовавшие функции.
- Нужен отдельный архив freeze, если требуется юридико-техническая фиксация именно KLT 4.13.

Финальное решение

Статус: reconstructed-legacy-checklist. Для юридико-технического freeze требуется самостоятельный пакет KLT 4.13 или явное утверждение, что baseline фиксируется через KLT 4.14.

KLT 4.14: итоговый развернутый чек-лист

checked source/build package: универсальный импорт, web/mobile shell, demo/paid gate

Статус исходного анализа

Проверенная сборка KLT_4_14_CHECKED_BUILD.zip. В текущем запуске selftest завершился с кодом 0.

Главная формула слоя

Input files/archives -> import/OCR adapter -> lambda audit -> DOCX checklist/register -> web/mobile shells -> paid/demo export gate.

Анализ исходного текста

- KLT 4.14 является переходным расширением KLT 4.13: сохраняет учебный checklist/register, но добавляет universal import, archive adapters, OCR adapter, локальную авторизацию, demo/paid режим, web shell и Flutter mobile shell.
- Исходный текст показывает модульность: importers.py отвечает за чтение форматов, archives.py за архивы, auth.py за локальный профиль, engine.py за scoring/metadata/plagiarism, reports.py за DOCX-выходы, app.py за CLI/batch/compare.
- CHECK_RESULT фиксирует ограничения: это source/build package, не подписанный APK/IPA; production payments и OCR требуют внешней инфраструктуры.

Текущий контрольный запуск

current selftest: exit 0; output: KLT 4.14 selftest OK.

Репер-паспорт программы

Компонент	Интерпретация
R build	Python package, CLI, web shell, DOCX templates, examples, verification output.
I_lambda_check	Проверка через lambda checklist, ideal-answer comparison, grade register, plagiarism sheet.
U import export	ТХТ/DOC/DOCX/HTML/PDF/images/TEX/ODT/EPUB/archives and batch export.
D verification	selftest.log, CHECK_RESULT, render.status, templates, verification DOCX/PDF/PNG.

Итоговый checklist

Пакет и запуск

ID	OK	Проверяемый пункт	Критерий приёмки / Evidence-D	Комментарий
414-PKG-01	[]	Архив KLT_4_14_CHECKED_BUILD.zip распаковывается без ошибок.	В структуре есть klt414/, examples/, templates/, verification/, web/, mobile flutter shell/, tests/, docs/.	
414-PKG-02	[]	CLI запускается через python -m klt414.app.	Команды single, demo, batch archive, compare доступны.	
414-PKG-03	[]	tests/selftest.py завершается успешно.	Текущий запуск: KLT 4.14 selftest OK.	
414-PKG-04	[]	CHECK_RESULT.txt включён в verification.	Версия 4.14.0, selftest=0, DOCX render=0.	

Импорт и извлечение текста

ID	OK	Проверяемый пункт	Критерий приёмки / Evidence-D	Комментарий
414-IMP-01	[]	ТХТ/MD/TEX/HTML читаются как текстовые источники.	HTML очищается от тегов; TEX читается как исходный текст.	

414-IMP-02	[]	DOCX читается через python-docx.	Параграфы извлекаются без потери основного текста.	
414-IMP-03	[]	PDF читается через PDF extractor.	Сканированные PDF помечаются как требующие OCR.	
414-IMP-04	[]	DOC/ODT/EPUB имеют отдельные адаптеры или fallback.	Ошибки импорта переводятся в import-gap, а не скрываются.	
414-IMP-05	[]	JPG/PNG/TIFF/BMP/WEBP идут через OCR-адаптер.	В production требуется tesseract/pytesseract или внешний OCR.	
414-IMP-06	[]	ZIP/RAR/7Z/TAR/TGZ проходят archive adapter.	RAR/7Z требуют дополнительных инструментов, если используются фактически.	

Расчётное ядро и отчёты

ID	OK	Проверяемый пункт	Критерий приёмки / Evidence-D	Комментарий
414-ENG-01	[]	score_text вычисляет сопоставление с идеальным ответом.	Результат содержит score, underwork, surplus, scale_pack.	
414-ENG-02	[]	grade5_from_100 корректно переводит 100-балльную шкалу в 5-балльную.	Шкала документирована и воспроизводима.	
414-ENG-03	[]	extract_metadata извлекает ФИО/класс/группу из текста или имени файла.	Нормализация класса применяется единообразно.	
414-ENG-04	[]	plagiarism_suspicion отделён от основного score.	Сравнение двух текстов выводит отдельный plagiarism sheet.	
414-REP-01	[]	create_checklist_docx создаёт индивидуальный DOCX.	В отчёте есть title, sections, таблицы, remarks, metadata.	
414-REP-02	[]	create_grade_register_docx создаёт ведомость.	Группировка по классам/группам сохраняется.	
414-REP-03	[]	create_plagiarism_sheet_docx создаёт отдельный лист подозрения.	Не подменяет юридическую или педагогическую оценку.	

Авторизация, demo/paid, безопасность

ID	OK	Проверяемый пункт	Критерий приёмки / Evidence-D	Комментарий
414-AUTH-01	[]	Профиль проверяющего содержит login, salted SHA-256 password hash, ФИО, телефон, email.	Пароль не хранится открытым текстом.	
414-AUTH-02	[]	Demo-mode допускает 3 проверки без DOCX export.	JSON результат сохраняется, DOCX выгрузка блокируется.	
414-AUTH-03	[]	Paid-mode открывает checklist, grade register, ZIP export.	Нет обхода paid gate в CLI.	
414-SEC-01	[]	Production должен добавить серверную авторизацию, rate limits, персональные согласия, action log.	Локальный профиль не считается полноценной production security.	

Web/mobile contour

ID	OK	Проверяемый пункт	Критерий приёмки / Evidence-D	Комментарий
414-WEB-01	[]	web/index.html существует и работает как drag-and-drop shell.	Для SaaS требуется проверка server.py/FastAPI и CORS policy.	
414-MOB-01	[]	mobile_flutter_shell содержит pubspec.yaml, main.dart, screens, services, RU/EN/ZH strings.	Это source-shell, не подписанные APK/IPA.	
414-MOB-02	[]	File picker / camera picker / demo-limit / in-app purchase stub описаны.	Реальные платежи подключаются через store/server.	

Freeze и приёмка

ID	OK	Проверяемый пункт	Критерий приёмки / Evidence-D	Комментарий
414-FRZ-01	[]	В package присутствуют README, MANIFEST, requirements, docs.	Сборка воспроизводима.	
414-FRZ-02	[]	DOCX render verification сохранён в verification/render_*.	Нет black square character в исходниках и verification JSON/logs.	
414-FRZ-03	[]	KLT 4.14 не заявляет собственность на классическую математику.	Авторский объект: программная архитектура KLT/RBD audit.	

Acceptance gate

KLT414Ready(x) <=> SourcePackage(x) and UniversalImport(x) and ChecklistReports(x) and AuthGate(x) and DemoPaidGate(x) and VerificationLog(x) and RenderQA(x).

ProductionReady(x) дополнительно требует server auth, payment integration, OCR runtime, RAR/7Z runtime, privacy consent, signed mobile builds.

Риски и обязательные оговорки

- OCR не самодостаточен без tesseract/pytesseract или внешнего OCR.
- RAR/7Z зависят от внешних системных инструментов.
- Flutter shell не является подписанным мобильным релизом.
- Локальная авторизация не равна промышленной серверной авторизации.

Финальное решение

Статус: ready-for-controlled-use при сохранении указанных ограничений. Production-ready только после внешней проверки окружения, безопасности и юридико-технической границы.

KLT 5.1: итоговый развернутый чек-лист

Unified Flutter SDK: PIX@PEAKS, Project JSON, lambda engine, SVG/JSON/MD reports

Статус исходного анализа

Проверенная SDK-сборка KLT5_1_FLUTTER_SDK_PACKAGE.zip. Python tests: 3 passed; CLI smoke-run: PASS. Flutter runtime в контейнере отсутствует.

Главная формула слоя

Project JSON -> PIX@PEAKS graph -> Reper(U,I,R,D) -> lambda/delta/Auth -> D*/MSI/DI/CRI -> SVG/JSON/MD report.

Анализ исходного текста

- KLT 5.1 переносит KLT из текстовой учебной проверки в проектно-графовый цифровой продукт: works/resources/deliveries/costs/finance превращаются в PIX-узлы и Reper-состояния.
- Исходный Python reference-core является проверяемым расчетным ядром: dataclasses Reper4/LambdaResult/PixNode/GraphEdge/Defect, функции lambda_truth, d_star, MSI, DI, CRI и класс Klt51Engine.
- Документация фиксирует юридическую границу: результат является аналитическим индикатором и не заменяет официальную экспертизу, инженерное заключение, аудит, юридическую консультацию или инвестиционную рекомендацию.

Текущий контрольный запуск

current pytest: 3 passed; CLI smoke-run: PIX=11, edges=9, defects=12, lambda_count=6, CRI=0.4473.

Reper-паспорт программы

Компонент	Интерпретация
R code	Flutter app + Python reference core + examples/out outputs.
I_method	Projective construction graph engine: PIX -> PEAKS -> Reper -> lambda -> defect -> scheme/report.
U runtime	Flutter UI, Python CLI, JSON projects, future import/API/PDF/DWG/BIM adapters.
D verification	pytest, CLI smoke-run, CHECK_RESULT, rendered PDF docs and SVG previews.

Итоговый checklist

Пакет и среда

ID	OK	Проверяемый пункт	Критерий приёмки / Evidence-D	Комментарий
51-PKG-01	[]	Архив KLT5_1_FLUTTER_SDK_PACKAGE.zip распаковывается.	Есть app/, core_python/, examples/, docs/, verification/, legacy_sources/.	
51-PKG-02	[]	Python reference-core тестируется отдельно от Flutter.	Текущий pytest: 3 passed.	
51-PKG-03	[]	CLI smoke-run выполняется на examples/demo_project.json.	Текущий результат: PIX=11, edges=9, defects=12, lambda_count=6, CRI=0.4473.	
51-PKG-04	[]	Flutter SDK execution помечен как external requirement.	Нужны flutter pub get, flutter test, flutter run в среде с Flutter SDK.	

Ввод Project JSON

ID	OK	Проверяемый пункт	Критерий приёмки / Evidence-D	Комментарий
----	----	-------------------	-------------------------------	-------------

51-IN-01	[]	works содержат зоны, даты, ресурсы, predecessors, deliveries, Reper.	Каждая работа становится PIX work-node.	
51-IN-02	[]	resources становятся ресурсными узлами.	Связь uses_resource строится из works.resources.	
51-IN-03	[]	deliveries требуют document_basis.	Поставка без основания создаёт logistics defect.	
51-IN-04	[]	costs имеют amount, parent, coefficient, basis, Reper.	Коэффициент без основания создаёт cost defect.	
51-IN-05	[]	finance имеет planned/actual/document_ref.	Отклонение >30% создаёт finance deviation.	
51-IN-06	[]	c_min/c_offer используются для DI.	Offer ниже контрольной стоимости создаёт deficit zone.	

Расчётные формулы

ID	OK	Проверяемый пункт	Критерий приёмки / Evidence-D	Комментарий
51-MATH-01	[]	lambda_truth вычисляет $\lambda = ((U-R)(I-D))/((U-D)(I-R))$.	При нулевом знаменателе статус singular.	
51-MATH-02	[]	$\delta = \lambda + 1 $, Auth = $1/(1 + \delta)$.	Высокий delta создаёт lambda defect.	
51-MATH-03	[]	$D^* = (2UI - UR - IR)/(U + I - 2R)$ вычисляется безопасно.	Нулевой знаменатель возвращает None.	
51-MATH-04	[]	MSI = $ \log(Cb/Ca) $ выявляет сметные скачки.	MSI > 0.5 создаёт cost scale jump.	
51-MATH-05	[]	DI = $\max(0, (Cmin - Coffer)/Cmin)$ выявляет дефицитную зону.	DI > 0.05 создаёт deficit defect.	
51-MATH-06	[]	CRI агрегирует severities, deltas, MSI, DI.	Значение нормировано в [0,1].	

Граф и дефекты

ID	OK	Проверяемый пункт	Критерий приёмки / Evidence-D	Комментарий
51-GR-01	[]	precedes edges строятся между работами.	Missing predecessor и technology overlap выявляются.	
51-GR-02	[]	resource conflicts находятся по пересечению ресурсов и дат.	Пересекающиеся работы с общим ресурсом создают resource defect.	
51-GR-03	[]	cost_rollup проверяет родительские суммы и дочерние позиции.	Несовпадение масштаба фиксируется через MSI.	
51-GR-04	[]	defects имеют id, class, title, message, severity, pix ids, payload.	Каждый дефект трассируется к узлам.	

Визуализация и отчёты

ID	OK	Проверяемый пункт	Критерий приёмки / Evidence-D	Комментарий
51-OUT-01	[]	result.json генерируется после анализа.	Содержит counts, nodes, edges, defects, lambda.	
51-OUT-02	[]	report.md генерируется с legal note.	Отчет не подменяет экспертное заключение.	
51-OUT-03	[]	scheme.svg показывает PIX@PEAKS graph.	Узлы подсвечиваются по defect severity.	
51-OUT-04	[]	blueprint.svg показывает работы по зонам и срокам.	Помогает выявлять фронт работ.	
51-OUT-05	[]	graph.svg показывает CRI/delta/MSI/DI/DEF.	Сводный визуальный риск-граф.	

Flutter UI

ID	OK	Проверяемый пункт	Критерий приёмки / Evidence-D	Комментарий
51-UI-01	[]	app/pubspec.yaml, lib/main.dart, models/services/screens/widgets	Структурная проверка Flutter проекта проходит.	

		существуют.	
51-UI-02	[]	RU/EN/ZH строки интерфейса присутствуют.	Трёхязычная оболочка готова к Flutter SDK loading.
51-UI-03	[]	Вкладки Dashboard/Input/Diagrams/Report/Method реализованы как цель.	Требуется runtime-проверка в Flutter SDK.

Юридико-техническая граница

ID	OK	Проверяемый пункт	Критерий приёмки / Evidence-D	Комментарий
51-LAW-01	[]	KLT 5.1 не заменяет экспертизу, аудит, юриста, инженера или инвестиционный вывод.	Это аналитический индикатор с proof/gap статусами.	
51-LAW-02	[]	Каждый вывод связан с source, checksum, test, documentation.	Без D-основания результат переводится в gap/review.	
51-LAW-03	[]	Future adapters PDF/DOCX/XLSX/DWG/BIM должны иметь отдельные proof logs.	Нельзя заявлять production import без адаптера и теста.	

Acceptance gate

KLT51Ready(x) <=> ProjectJSON(x) and PythonCorePass(x) and LambdaEngine(x) and DefectTrace(x) and SVGJSONMMDOutput(x) and FlutterStructurePass(x).

KLT51RuntimeReady(x) дополнительно требует real Flutter SDK run, app tests, platform build, PDF export QA and API integration tests.

Риски и обязательные оговорки

- Flutter не запускался в текущем контейнере из-за отсутствия SDK.
- Импорт PDF/DOCX/XLSX/DWG/BIM находится в будущих адаптерах, а не в полном runtime 5.1.
- Экономические индикаторы CRI/MSI/DI требуют калибровки на реальных кейсах.

Финальное решение

Статус: ready-for-controlled-use при сохранении указанных ограничений. Production-ready только после внешней проверки окружения, безопасности и юридико-технической границы.

KLT 5.10: итоговый развернутый чек-лист

target next-generation KLT-RBD: Flutter/Web + FastAPI + extraction/OCR + RBD + LambdaAudit + Rebuild

Статус исходного анализа

В текущем корпусе отдельного архива с именем KLT 5.10 не найдено. Чек-лист оформляет KLT 5.10 как целевую следующую ступень на основе klt_rbd_v2_2_flutter_web_program, KLT 5.1 и публичного контура v1.8-v2.4.

Главная формула слоя

Intake(F) + OCR + Extract + RBD + LambdaAudit + LimitGates + Report + Rebuild.

Анализ исходного текста

- KLT 5.10 должен быть не просто расширенным KLT 5.1, а системой нового поколения: загрузка смешанных файлов, извлечение фрагментов с локаторами, ReperStore, доменные критерии, LambdaAudit, LimitGates, HTML/API/UI отчётность, очередь спорных мест и пересборка.
- Исходный klt_rbd_v2_2_flutter_web_program уже содержит Backend/FastAPI, extractor layer, analyzer, report_html, store, Flutter app и site_upload; это достаточная техническая база для target checklist KLT 5.10.
- С учётом публичного контура v1.8-v2.4 KLT 5.10 должен иметь два режима: controlled engine с серверной обработкой и public static demo/read-only route без раскрытия controlled Evidence-D.

Текущий контрольный запуск

current backend import/health check: status=ok, version=2.2.0, repers=6, domains=construction/finance/science_text.

Reper-паспорт программы

Компонент	Интерпретация
R_runtime	FastAPI backend, Flutter UI, extraction/OCR, RBD seed, reports, site upload.
I_rebuild	Метод KLT: RBD, lambda-сопоставление, LIM limits, PIX@PEAKS, спорные места, рекомендации, rebuild.
U_domains	construction, finance, science_text, general; future adapters for CAD/BIM/bank/exchange formats.
D_audit	SHA256SUMS, source files, health endpoint, supported formats, method notice, local/live check logs.

Итоговый checklist

Граница KLT 5.10

ID	ОК	Проверяемый пункт	Критерий приёмки / Evidence-D	Комментарий
510-ID-01	[]	KLT 5.10 фиксируется как target next-generation layer, а не как уже найденный отдельный source archive.	В паспорте указана основа: KLT-RBD v2.2 + KLT 5.1 + public demo v1.8-v2.4.	
510-ID-02	[]	Сохраняется различие controlled engine и public static demo.	Публичный слой не раскрывает controlled Evidence-D.	
510-ID-03	[]	Версия не должна подменять source of truth сайтом или API.	Источник истины: source package + RBD node + Evidence-D + lambda/CGI + checksum + QA + rollback.	

Backend / API

ID	OK	Проверяемый пункт	Критерий приёмки / Evidence-D	Комментарий
510-API-01	[]	FastAPI app имеет /health.	Текущий import/health check: status=ok, version=2.2.0, repers=6.	
510-API-02	[]	/repers возвращает поиск по ReperStore.	Проверяются query/domain/limit и список доменов.	
510-API-03	[]	/analyze/file принимает UploadFile, domain, language, enable ocr.	Файл сохраняется во временный путь и удаляется после анализа.	
510-API-04	[]	/analyze/text принимает JSON payload.	Создаёт text_input.txt fragment и возвращает AnalysisResult.	
510-API-05	[]	/report/html строит HTML по AnalysisResult.	HTML-отчёт воспроизводим из JSON.	
510-API-06	[]	CORS policy на production должен быть ограничен.	allow_origins=["*"] допустимо для dev, не для публичного production.	

Extraction / OCR / локаторы

ID	OK	Проверяемый пункт	Критерий приёмки / Evidence-D	Комментарий
510-EXT-01	[]	TEXT formats читаются с decode fallback: utf-8, utf-16, cp1251, latin-1.	Ошибки не должны приводить к silent data loss.	
510-EXT-02	[]	HTML/XML/FB2/XBRL/OFX/SVG очищаются от тегов.	Фрагменты сохраняют locator file/line.	
510-EXT-03	[]	CSV возвращает fragments с row locator.	Нужен контроль разделителей и кодировок.	
510-EXT-04	[]	DOCX возвращает paragraph fragments.	Нужно добавить таблицы/сноски/headers в будущих версиях.	
510-EXT-05	[]	XLSX/XLSM возвращает spreadsheet row с sheet/row/cell.	Нужна отдельная проверка формул и merged cells.	
510-EXT-06	[]	PDF возвращает page fragments.	Сканированные PDF требуют OCR path.	
510-EXT-07	[]	Images идут через pytesseract при enable ocr=True.	Runtime должен иметь tesseract и языковые пакеты.	
510-EXT-08	[]	ZIP/EPUB извлекает до 80 текстовых members.	Нужна защита от zip bombs и path traversal.	
510-EXT-09	[]	Зарегистрированы внешние форматы: DOC/XLS/DJVU/MOBI/AZW3/DWG/STEP/IFC/OBJ/GLTF/RVT и банковские форматы.	Каждый внешний adapter должен иметь отдельный D-log.	

Analyzer / критерии / спорные места

ID	OK	Проверяемый пункт	Критерий приёмки / Evidence-D	Комментарий
510-AN-01	[]	analyze_extracted связывает fragments с ReperStore.	Если нет доменных Reper, создаётся KLT-GEN-000.	
510-AN-02	[]	Criterion содержит score_5, score_100, score_100000, weight, explanation, evidence.	Каждый критерий должен иметь локаторы evidence, когда они найдены.	
510-AN-03	[]	Findings обнаруживают statements без основания, construction logistics/cost/resources, science sources, finance risks.	Спорное место содержит severity, criterion_id, message, locator, recommendation.	
510-AN-04	[]	lambda_value и defect_delta считаются из итогового score.	Формула должна быть явно документирована, чтобы не стать скрытым индексом.	
510-AN-05	[]	Graph nodes/edges строятся по домену.	Graph должен отображать путь от тезиса/документа к риску и исправлению.	

Домены и форматы

ID	OK	Проверяемый пункт	Критерий приёмки / Evidence-D	Комментарий
510-DOM-01	[]	construction domain проверяет графики, ресурсы, бригады, поставки, сметы, фронт работ.	Нужен mapping на ФЕР/ФССЦ/ГЭСН при строительных сметах.	
510-DOM-02	[]	science_text domain проверяет литературу, источники, эксперимент, DOI/ISBN, плагиатный риск.	Нужно разделять методический вывод и bibliographic fact.	
510-DOM-03	[]	finance domain проверяет показатель, основание, период, риск, сценарий.	Не является инвестиционной рекомендацией.	
510-DOM-04	[]	general domain должен оставаться fallback, не выдающий ложный truth-status.	При слабом D создаётся needs-D/review.	

Flutter/Web/UI

ID	OK	Проверяемый пункт	Критерий приёмки / Evidence-D	Комментарий
510-UI-01	[]	Flutter app имеет RU/EN/ZH interface.	Проверить pubspec, assets/i18n, main.dart, app.dart.	
510-UI-02	[]	UI должен показывать загрузку файлов, OCR switch, checklist, review, graph, спорные места.	Пользователь видит не только score, но и основания.	
510-UI-03	[]	site_upload содержит страницу для сайта рядом с KLT4.14 и KLT5.1.	Для текущего хостинга нужны короткие имена: file<=15, folder<=10, depth<=5, ASCII.	
510-UI-04	[]	Public demo route должен быть static read-only.	Нет write-back в controlled RBD.	

Security / privacy / production

ID	OK	Проверяемый пункт	Критерий приёмки / Evidence-D	Комментарий
510-SEC-01	[]	Временные файлы удаляются после analyze/file.	Проверяется finally unlink.	
510-SEC-02	[]	Upload size, extension whitelist и malware scanning должны быть добавлены до production.	Dev-сборка не равна public production.	
510-SEC-03	[]	No-private-leak gate обязателен для public route.	Controlled Evidence-D не публикуется.	
510-SEC-04	[]	Все release artifacts имеют SHA256 и rollback point.	Без checksum пакет не считается freeze-ready.	
510-SEC-05	[]	CORS, auth, rate limits, logging, consent, retention policy обязательны для server mode.	Иначе только local/demo.	

KLT 5.10 acceptance

ID	OK	Проверяемый пункт	Критерий приёмки / Evidence-D	Комментарий
510-ACC-01	[]	Backend health, supported formats, extraction, analyzer, report_html проверены.	Есть smoke logs.	
510-ACC-02	[]	Flutter Web build проверен в реальной Flutter SDK среде.	Нужен build log и browser QA.	
510-ACC-03	[]	Public static payload проходит hosting constraints.	file<=15, folder<=10, depth<=5, ASCII.	
510-ACC-04	[]	Controlled engine и public demo имеют разные manifests.	Нет смещения private/controlled и public.	
510-ACC-05	[]	RBD seed, ReperStore, lambda audit и recommendations воспроизводимы.	Есть test cases по construction/finance/science_text.	

Acceptance gate

KLT510Ready(x) <=> BackendHealth(x) and Extractors(x) and ReperStore(x) and LambdaAudit(x) and DomainFindings(x) and HTMLReport(x) and FlutterUI(x) and NoPrivateLeak(x) and ChecksumRollback(x).

Public510Ready(x) <=> StaticPayload(x) and ShortHostingPaths(x) and ReadOnly(x) and NoWriteBack(x) and NoControlledEvidence(x).

Риски и обязательные оговорки

- Нет отдельного исходного архива с именем KLT 5.10; чек-лист является целевой спецификацией на базе KLT-RBD v2.2 и публичного контура v2.4.
- FastAPI dev CORS открыт; перед production нужен security hardening.
- OCR, CAD/BIM, банковские и биржевые адаптеры требуют runtime-тестов и отдельных proof logs.
- Public demo не должен становиться источником истины или write-back engine.

Финальное решение

Статус: target-architecture-checklist. Для статуса готового KLT 5.10 требуется отдельная сборка, runtime QA, security hardening и public/private split.