

# KLT 5.1. Итоговый чек-лист

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.

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

### Юридикто-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

$KLT51Ready(x) \Leftrightarrow ProjectJSON(x) \text{ and } PythonCorePass(x) \text{ and } LambdaEngine(x) \text{ and } DefectTrace(x) \text{ and } SVGJSONMDOuput(x) \text{ and } FlutterStructurePass(x).$

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.